

**UNIVERSITY COLLEGE OF ENGINEERING  
KANCHEEPURAM**

**(A *Constituent college of Anna University*  
*Chennai*)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



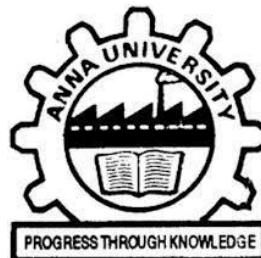
**CS 8711 – CLOUD COMPUTING LABORATORY**

**Name:** \_\_\_\_\_

**Register no:** \_\_\_\_\_

**Year/Semester:** \_\_\_\_\_ **Branch:** \_\_\_\_\_

**UNIVERSITY COLLEGE OF ENGINEERING KANCHEEPURAM**  
**KANCHEEPURAM-631 552**



**BONAFIDE CERTIFICATE**

**Registration No.**

<input type="text"/>											
----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------

Certified that this is the bonafide record of work done by  
Mr/Ms..... of ..... semester B.E. Computer Science and  
Engineering Branch / Batch during the academic year 20....to 20.... in the  
.....

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

**Staff In-Charge**

**Head of the Department**

**Internal Examiner**

**External Examiner**

# INDEX

S.NO	Date	Title	Page Number	signature
1		<b>Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows 7 or 8.</b>	4	
2		<b>Install a C compiler in the virtual machine created using virtual box and execute Simple Programs</b>	17	
3		<b>Install Google App Engine. Create <i>hello world</i> app and other simple web applications using python/java.</b>	35	
4		<b>Use GAE launcher to launch the web applications</b>	35	
5		<b>Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.</b>	48	
6		<b>Find a procedure to transfer the files from one virtual machine to another virtual machine</b>	56	
7		<b>Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)</b>	59	
8		<b>Install Hadoop single node cluster and run simple applications like word count</b>	68	

## **Ex No. 1 | Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows 7 or 8.**

### **Introduction**

#### **❖ Virtualization:**

- Virtualization is the creation of virtual servers, infrastructures, devices and computing resources.
- Virtualization changes the hardware-software relations and is one of the foundational elements of cloud computing technology that helps utilize the capabilities of cloud computing to the full.
- Virtualization techniques allow companies to turn virtual their networks, storage, servers, data, desktops and applications.

#### **❖ Hypervisor or Virtual Machine Monitor (VMM)**

A **hypervisor** or **virtual machine monitor (VMM)** is a piece of computer software, firmware or hardware that creates and runs virtual machines. A computer on which a hypervisor is running one or more virtual machines is defined as a *host machine*. Each virtual machine is called a *guest machine*. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources.

#### **❖ Types of Virtualization**

- **Operating-system-level virtualization** - is a server-virtualization method where the kernel of an operating system allows for multiple isolated user-space instances, instead of just one. Such instances (sometimes called **containers**, **software containers**,<sup>[1]</sup> virtualization engines (VE), virtual private servers (VPS), or jails) may look and feel like a real server from the point of view of its owners and users
- **Platform / Hardware virtualization** -Hardware virtualization or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine.
- **In hardware virtualization**, the host machine is the actual machine on which the virtualization takes place, and the guest machine is the virtual machine. The words *host* and *guest* are used to distinguish the software that runs on the physical machine from the software that runs on the virtual machine. Different types of hardware virtualization include:

- o **Full virtualization:** Almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified.
- o **Partial virtualization:** Some but not all of the target environment is simulated. Some guest programs, therefore, may need modifications to run in this virtual environment.
- o **Para virtualization:** A hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate system.
- **Application virtualization** is software technology that encapsulates computer programs from the underlying operating system on which it is executed. A fully virtualized application is not installed in the traditional sense, although it is still executed as if it were.

#### ❖ Oracle Virtualbox

- o VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS. VMware Workstation is developed and sold by VMware, Inc., a division of EMC Corporation

#### ❖ Ubuntu

- o Ubuntu is an operating system like any other and it is free & open source. It means that we can download it freely and install on as many computers as we like. By the term open source it means that we can actually see its code. To provide a more secure environment, the -SUDO! tool is used to assign temporary privileges for performing administrative tasks. Ubuntu comes installed with a wide range of Software that includes Libre Office ,Firefox, Thunderbird

### Steps in Installing Oracle Virtualbox with Ubuntu

#### Step 1:

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

#### Step 2:

- Download Ubuntu 7 OVA(Open Virtual Appliance) from
  - o Link : <https://linuxvmimages.com/images/Ubuntu-7>

**Step 3:** The files are downloaded in your Local machine and Click the Oracle VM VirtualBox 6.0.8 Setup Wizard

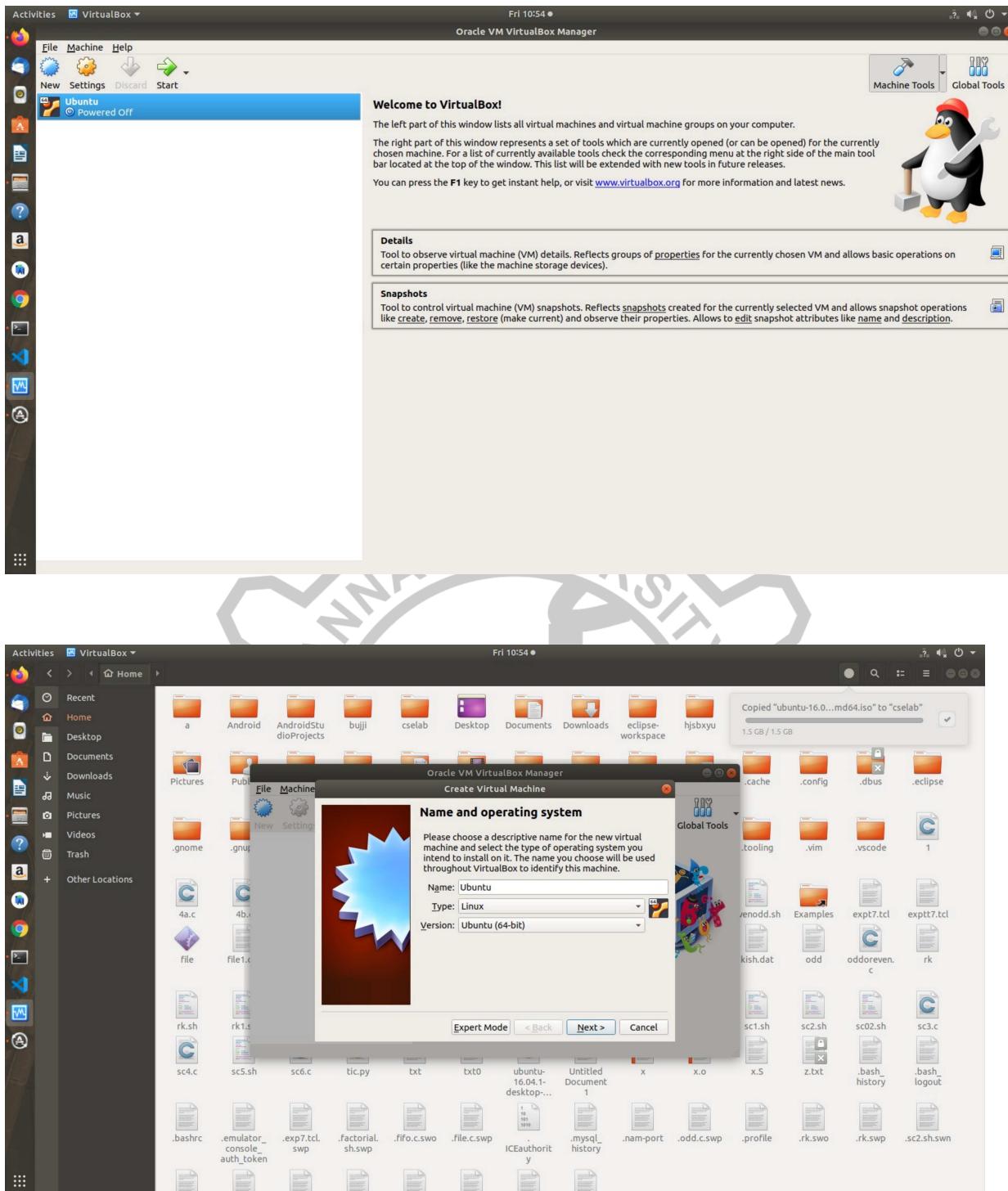
```
Fri 10:33 • cselab@cselab-Veriton-M2640G: ~
Activities Terminal
File Edit View Search Terminal Help
The following NEW packages will be installed:
dkms libgsoap-2.8.60 libqt5opengl5 libvncserver1 virtualbox virtualbox-dkms virtualbox-qt
0 upgraded, 7 newly installed, 0 to remove and 194 not upgraded.
Need to get 27.1 MB of additional disk space will be used.
After this operation, 117 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 dkms all 2.3-3ubuntu9.7 [68.1 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 libgsoap-2.8.60 amd64 2.8.60-2build1 [222 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libqt5opengl5 amd64 5.9.5+dfsg-0ubuntu2.6 [132 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 virtualbox-dkms all 5.2.42-dfsg-0-ubuntu1.18.04.1 [664 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 virtualbox amd64 5.2.42-dfsg-0-ubuntu1.18.04.1 [17.3 MB]
Get:6 http://in.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 virtualbox-qt amd64 5.2.42-dfsg-0-ubuntu1.18.04.1 [8,603 kB]
Fetched 27.1 MB in 3s (9,182 kB/s)
? Selecting previously unselected package dkms.
(Reading database ... 316140 files and directories currently installed.)
Preparing to unpack .../0-dkms_2.3-3ubuntu9.7_all.deb ...
Unpacking dkms (2.3-3ubuntu9.7) ...
? Selecting previously unselected package libgsoap-2.8.60;amd64.
Unpacking libgsoap-2.8.60;amd64 (2.8.60-2build1_amd64.deb) ...
? Selecting previously unselected package libqt5opengl5;amd64.
Preparing to unpack .../1-libqt5opengl5_5.9.5+dfsg-0ubuntu2.6_amd64.deb ...
Unpacking libqt5opengl5;amd64 (5.9.5+dfsg-0ubuntu2.6) ...
? Selecting previously unselected package libvncserver1;amd64.
Preparing to unpack .../2-libvncserver1_0.9.11+dfsg-1ubuntu1.4_amd64.deb ...
Unpacking libvncserver1;amd64 (0.9.11+dfsg-1ubuntu1.4) ...
? Selecting previously unselected package virtualbox-dkms.
Preparing to unpack .../3-libvncserver1_0.9.11+dfsg-1ubuntu1.4_amd64.deb ...
Unpacking libvncserver1;amd64 (0.9.11+dfsg-1ubuntu1.4) ...
? Selecting previously unselected package virtualbox-dkms.
Preparing to unpack .../4-virtualbox-dkms_5.2.42-dfsg-0-ubuntu1.18.04.1_all.deb ...
Unpacking virtualbox-dkms (5.2.42-dfsg-0-ubuntu1.18.04.1) ...
? Selecting previously unselected package virtualbox.
Preparing to unpack .../5-virtualbox_5.2.42-dfsg-0-ubuntu1.18.04.1_amd64.deb ...
Unpacking virtualbox (5.2.42-dfsg-0-ubuntu1.18.04.1) ...
? Selecting previously unselected package virtualbox-qt.
Preparing to unpack .../6-virtualbox-qt_5.2.42-dfsg-0-ubuntu1.18.04.1_amd64.deb ...
Unpacking virtualbox-qt (5.2.42-dfsg-0-ubuntu1.18.04.1) ...
Setting up libvncserver1;amd64 (0.9.11+dfsg-1ubuntu1.4) ...
Setting up libqt5opengl5;amd64 (5.9.5+dfsg-0ubuntu2.6) ...
Setting up libgsoap-2.8.60;amd64 (2.8.60-2build1) ...
Setting up dkms (2.3-3ubuntu9.7) ...
Setting up virtualbox-dkms (5.2.42-dfsg-0-ubuntu1.18.04.1) ...
Loading new virtualbox-5.2.42 DKMS files...
Building for 5.4.0-91-generic
Building initial module for 5.4.0-91-generic
[Progress: [====] 83% [#####
Fri 10:34 • cselab@cselab-Veriton-M2640G: ~
```

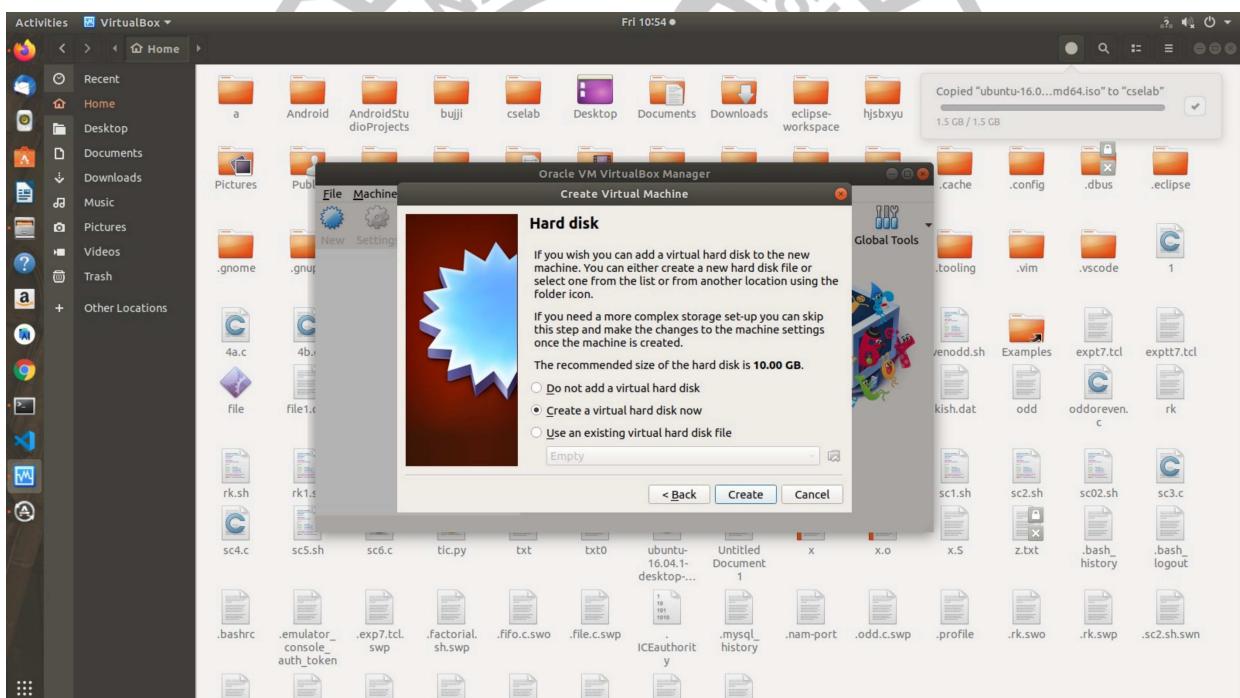
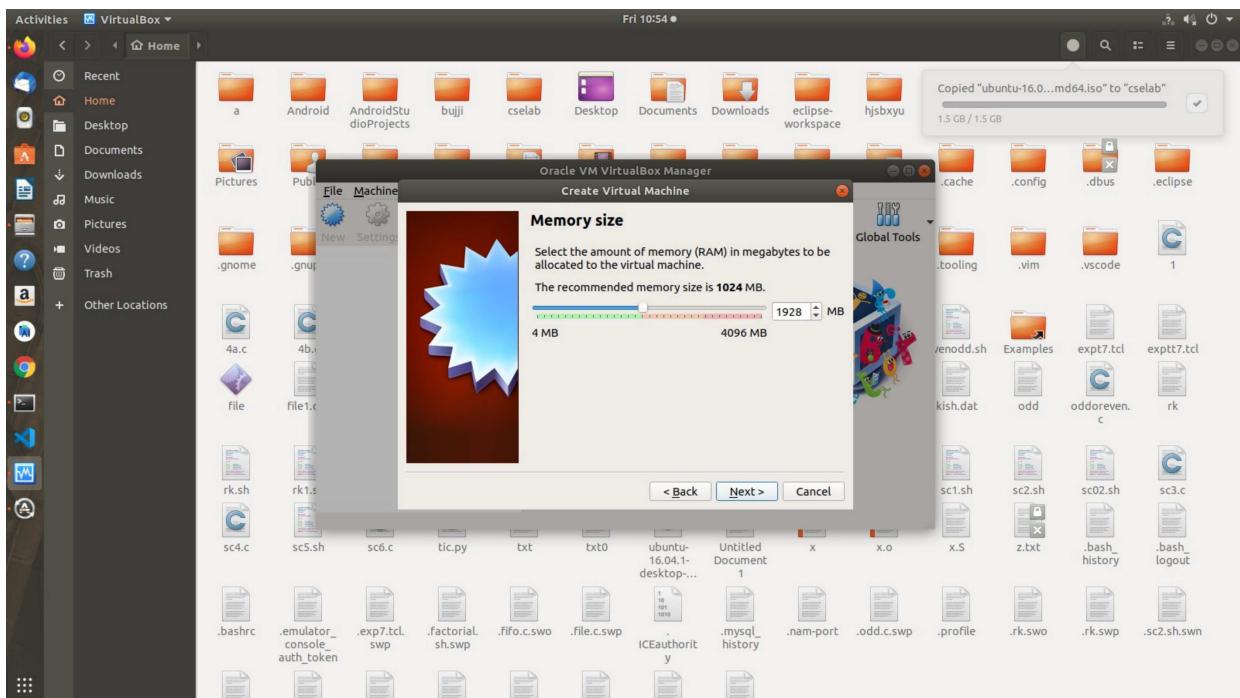
Oracle VM VirtualBox 6.0.8 Setup Wizard is open and follows the steps to install

```
Activities VirtualBox *
File Edit View Search Terminal Help
Running module version sanity check.
- Original module
  - No original module exists within this kernel
  - Installation
    - Installing to /lib/modules/5.4.0-91-generic/updates/dkms/
vboxnetadp.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
  - Installation
    - Installing to /lib/modules/5.4.0-91-generic/updates/dkms/
vboxnetflt.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
  - Installation
    - Installing to /lib/modules/5.4.0-91-generic/updates/dkms/
vboxpci.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
  - Installation
    - Installing to /lib/modules/5.4.0-91-generic/updates/dkms/
depmod.....
DKMS: install completed.
Setting up virtualbox (5.2.42-dfsg-0-ubuntu1.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for desktop-file-utils (0.23-3ubuntu3.18.04.2) ...
Processing triggers for libgbn (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu18.50) ...
Processing triggers for man-db (2.7.5-1ubuntu1.1) ...
Processing triggers for shared-mime-info (1.9-2) ...
Processing triggers for gnome-menus (3.13.3-1ubuntu1.1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
cselab@cselab-Veriton-M2640G: ~
```



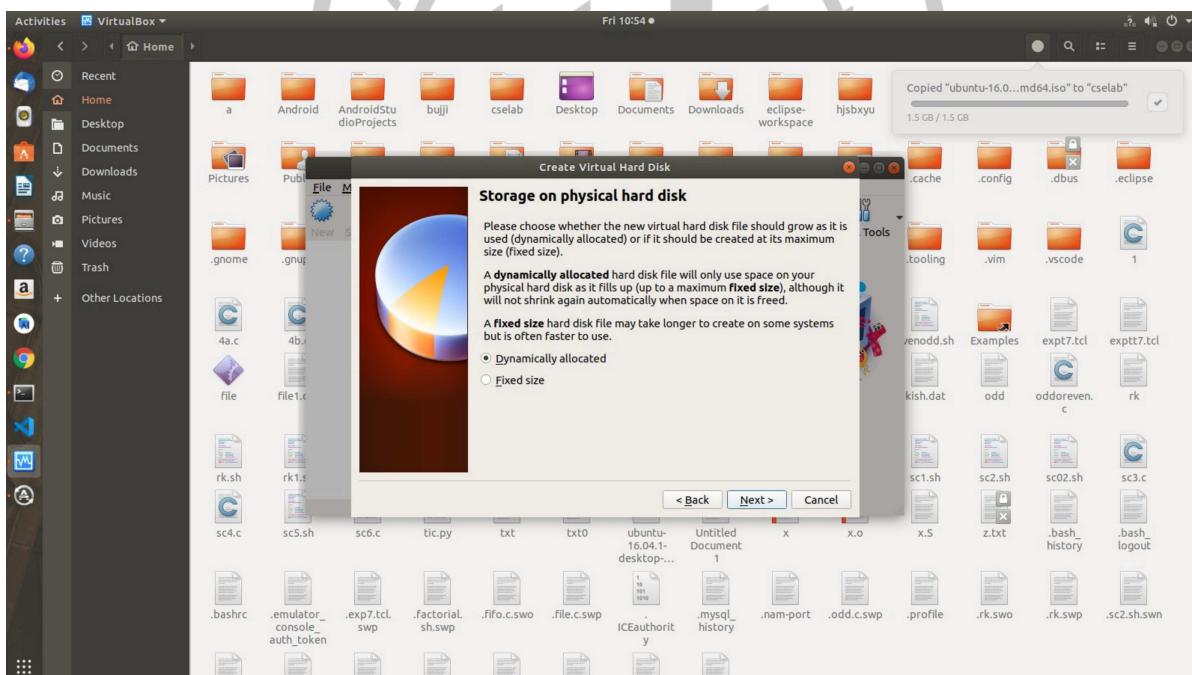
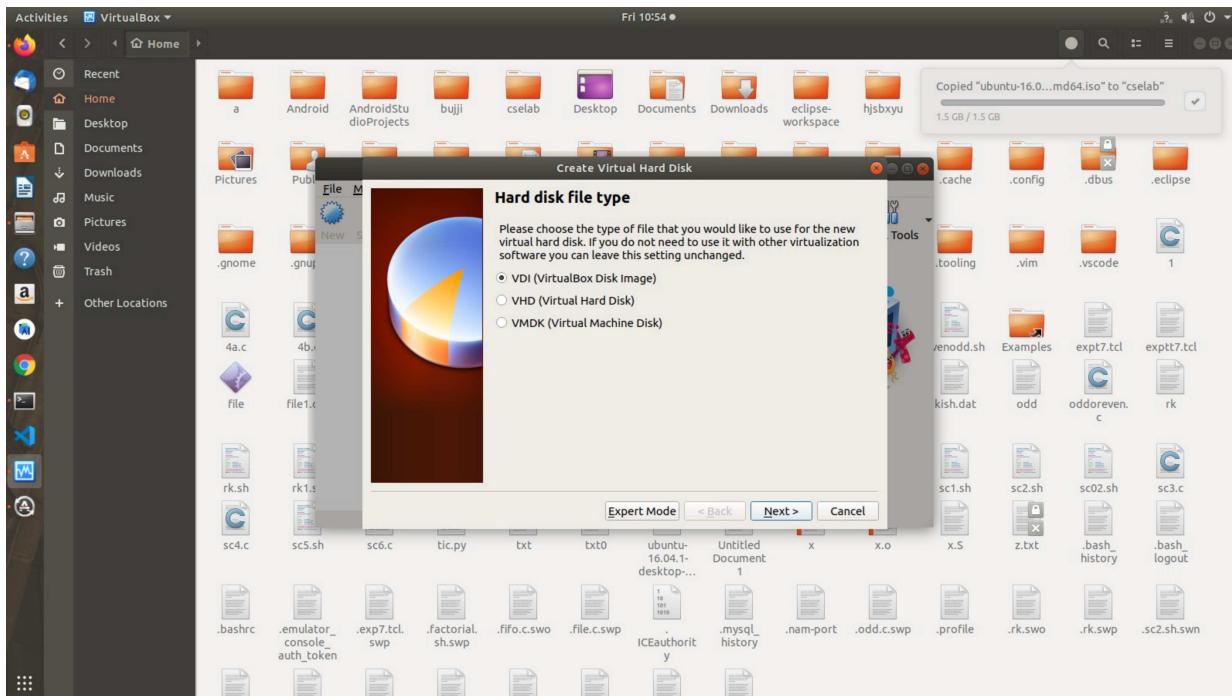
## Step 4: Import the Oracle VM Virtual Extension pack into the Oracle Virtual Box

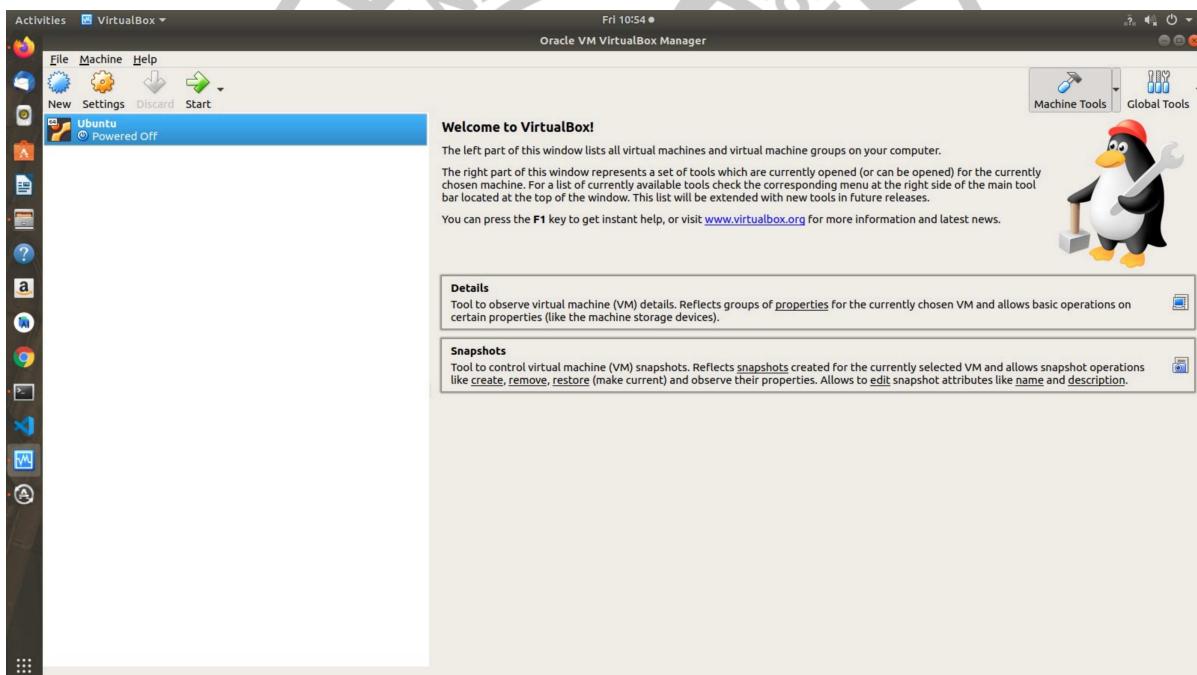


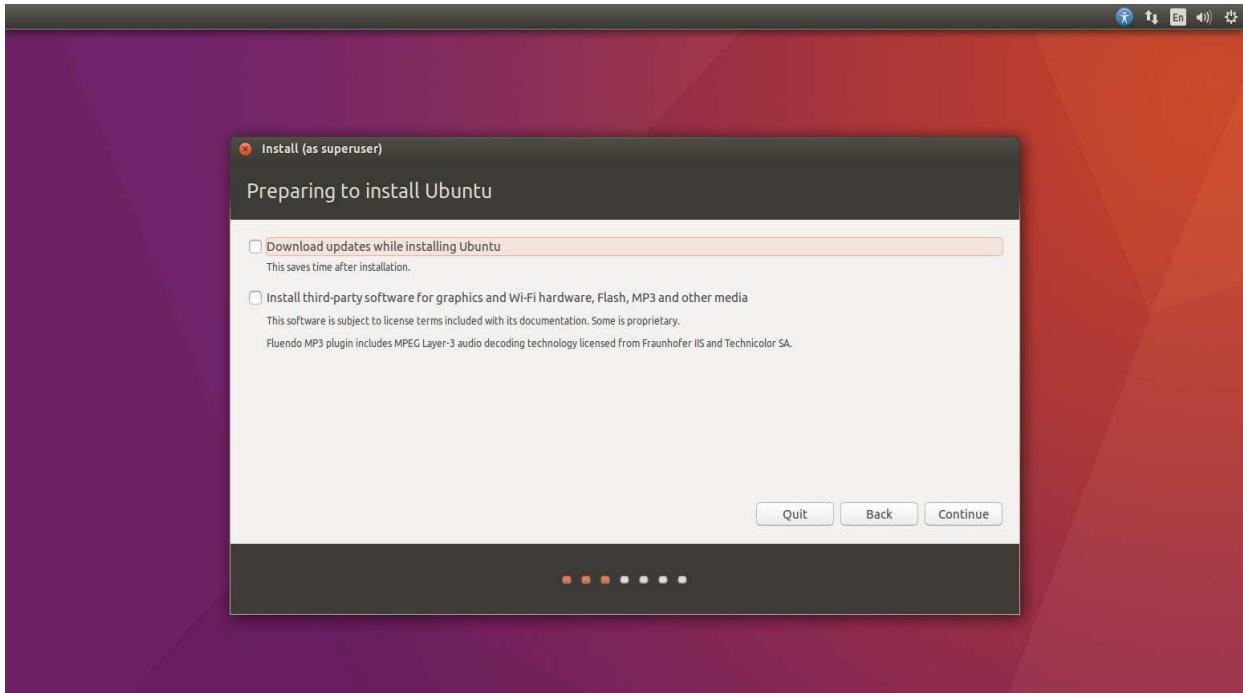


**Oracle VMVirtual Extension pack is imported into the Oracle Virtual Box successfully.**

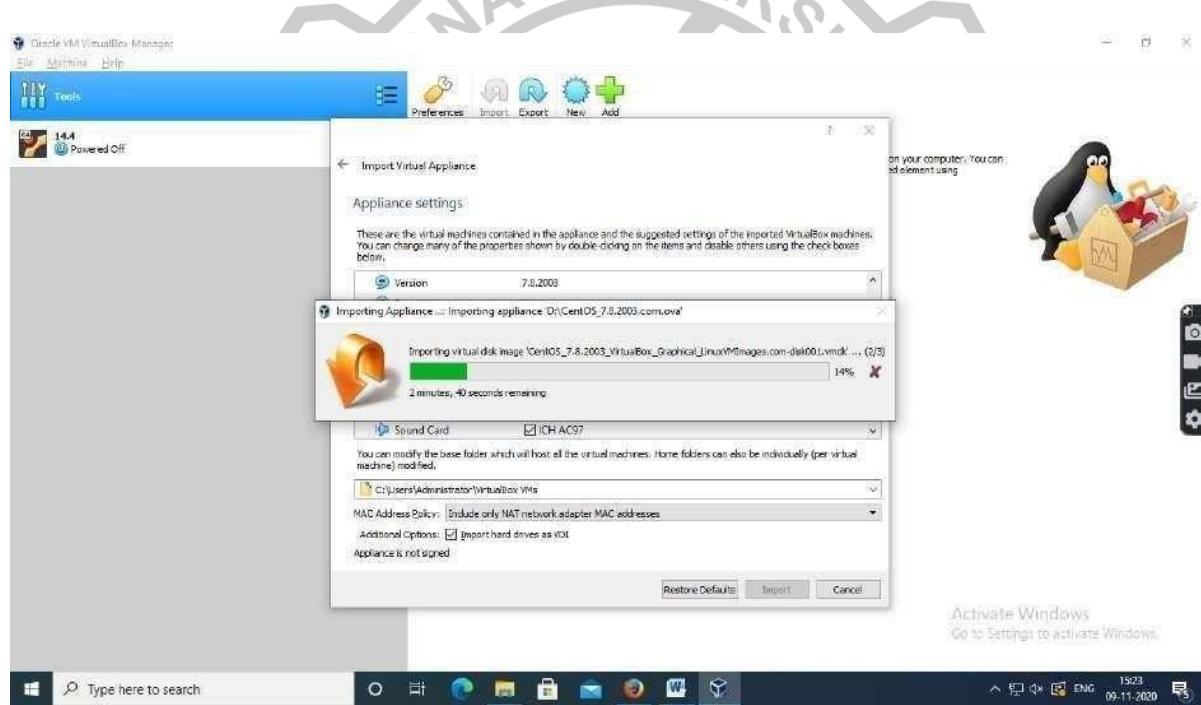
**Step 5:** Click import Appliance and select Ubuntu-7.8.2003.com OVA (Open Virtual Appliance) file from download directory



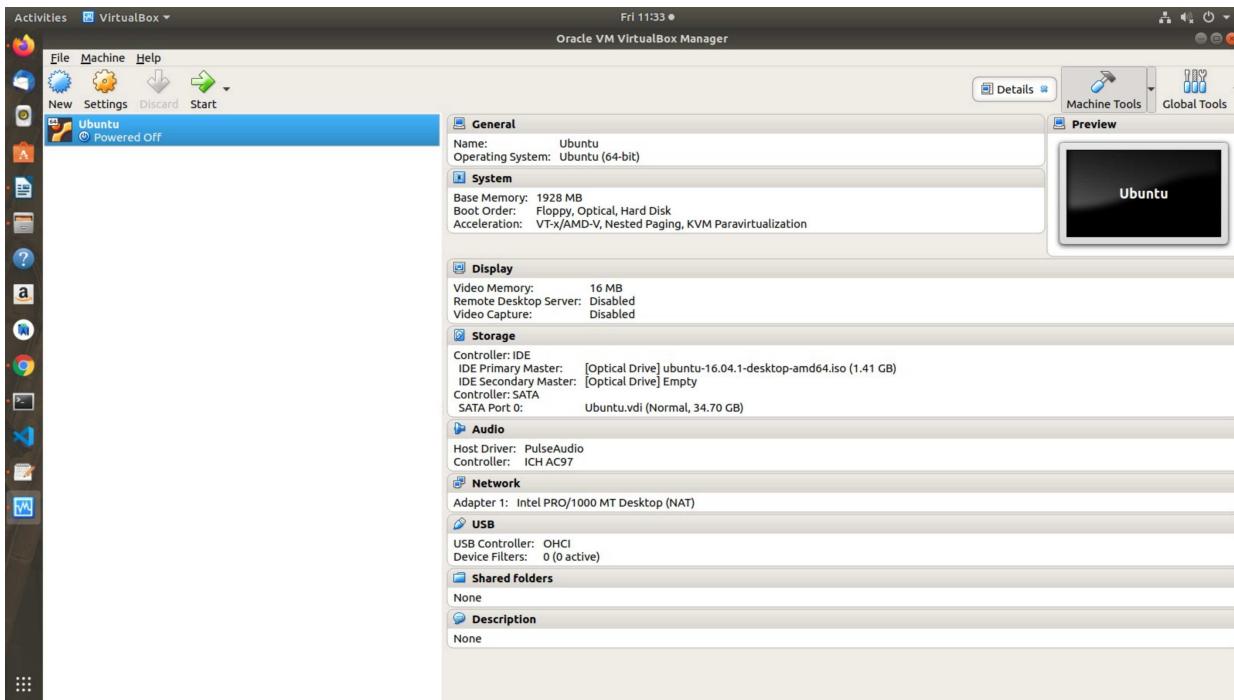




## Installation process started



**Step 6:** Ubuntu-7.8.2003.com is installed successfully and click start button to launch the virtual machine



**Step 7:** Login into Ubuntu

- Login Details
  - User name : ubuntu
  - Password : ubuntu

**PROGRESS THROUGH KNOWLEDGE**

### **Result:**

Oracle VirtualBox is installed with different flavours of Ubuntu 7 on top of windows 10 successfully.

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

**Steps in Installing C or C++ Compiler in Virtual machine and executing simple programs**

Step 1 : Install the C or C++ compiler on Ubuntu-14.04 Virtual Machine by

**\$ sudo apt install g++**

Step 2: Create a file for writing C program.

**\$ sudogedit add.c**

**Source Code:**

```
Sum of two numbers
#include<stdio.h>
int main()
{
    int a,b,c;
    printf("Enter two nos:");
    scanf("%d%d",&a,&b);
    c=0;
    c=a+b;
    printf("Sum of two nos is:
    %d",c); return 0;
}
```

Step 3: Compile the Program

**\$sudo g++ add.c**

Step 4: Run the Program

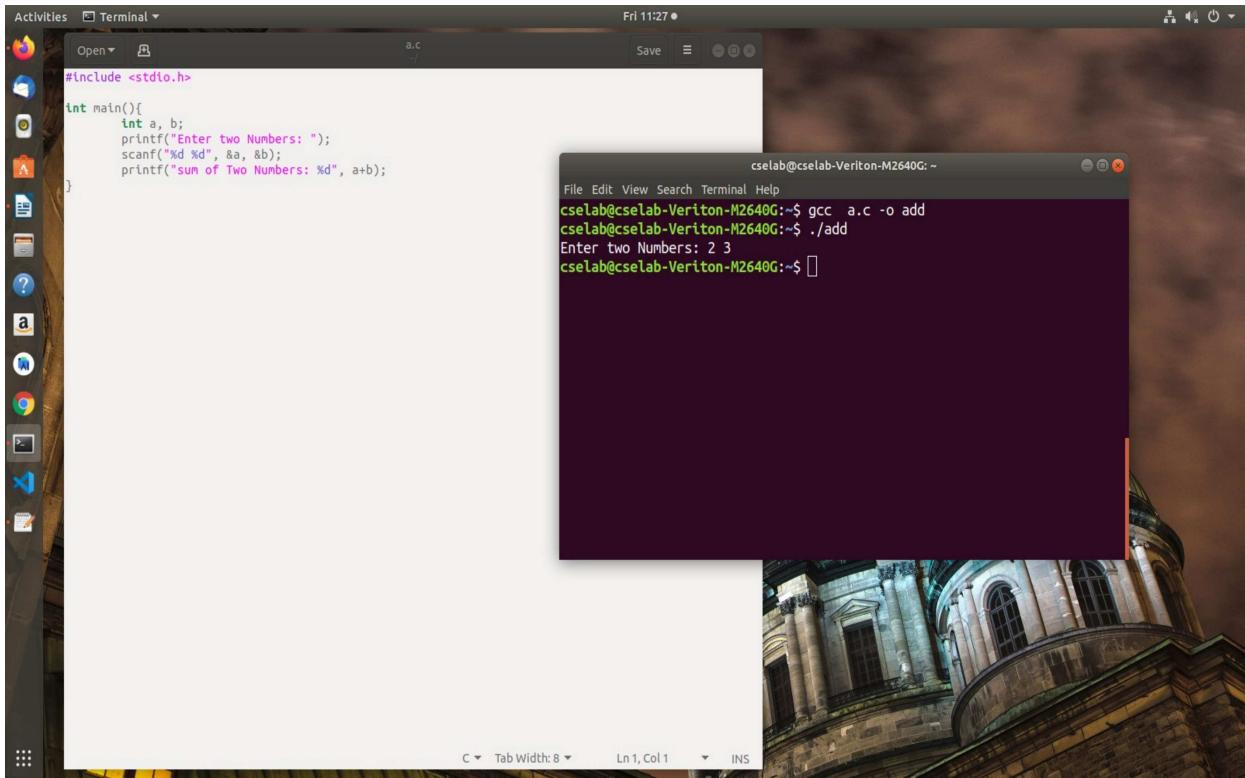
**\$ ./a.out**

Expected Output:

Enter two nos : 2 3

Sum of two nos is: 5

## **Output:**



A screenshot of a Linux desktop environment. On the left is a dock with various icons. In the center is a terminal window titled 'Terminal' with the file 'a.c' open. The code in 'a.c' is:

```
#include <stdio.h>
int main(){
    int a, b;
    printf("Enter two Numbers: ");
    scanf("%d %d", &a, &b);
    printf("sum of Two Numbers: %d", a+b);
}
```

The terminal window shows the output of running the program:

```
File Edit View Search Terminal Help
cselab@cselab-Veriton-M2640G:~$ gcc a.c -o add
cselab@cselab-Veriton-M2640G:~$ ./add
Enter two Numbers: 2 3
cselab@cselab-Veriton-M2640G:~$
```



## **Result:**

The simple C programs are executed with C compiler in the Virtual Machine successfully and different programs are executed and verified.

<b>Ex No. 3 &amp; 4</b>	<b>Install Google App Engine. Create hello world app and other simple web applications using python/java. Use GAE launcher to launch the web applications</b>
-----------------------------	---

### Introduction

#### □ Google Cloud Platform (GCP)

- o **Google Cloud Platform (GCP)**, offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube.
- o Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning.
- o Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.



#### □ Platform as a Service (PaaS)

- o Cloud computing service which provides a computing platform and a solution stack as a service.
- o Consumer creates the software using tools and/or libraries from the provider.
- o Provider provides the networks, servers, storage, etc.

#### □ Google App Engine:

- o Google App Engine was first released as a beta version in April 2008.
- o It is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers.

- o Google's App Engine opens Google's production to any person in the world at no charge.
- o Google App Engine is software that facilitates the user to run his web applications on Google infrastructure.
- o It is more reliable because failure of any server will not affect either the performance of the end user or the service of the Google.
- o It virtualizes applications across multiple servers and data centers.
  - Other cloud-based platforms include offerings such as Amazon Web Services and Microsoft's Azure Services Platform.

## □ **Introduction of Google App Engine**

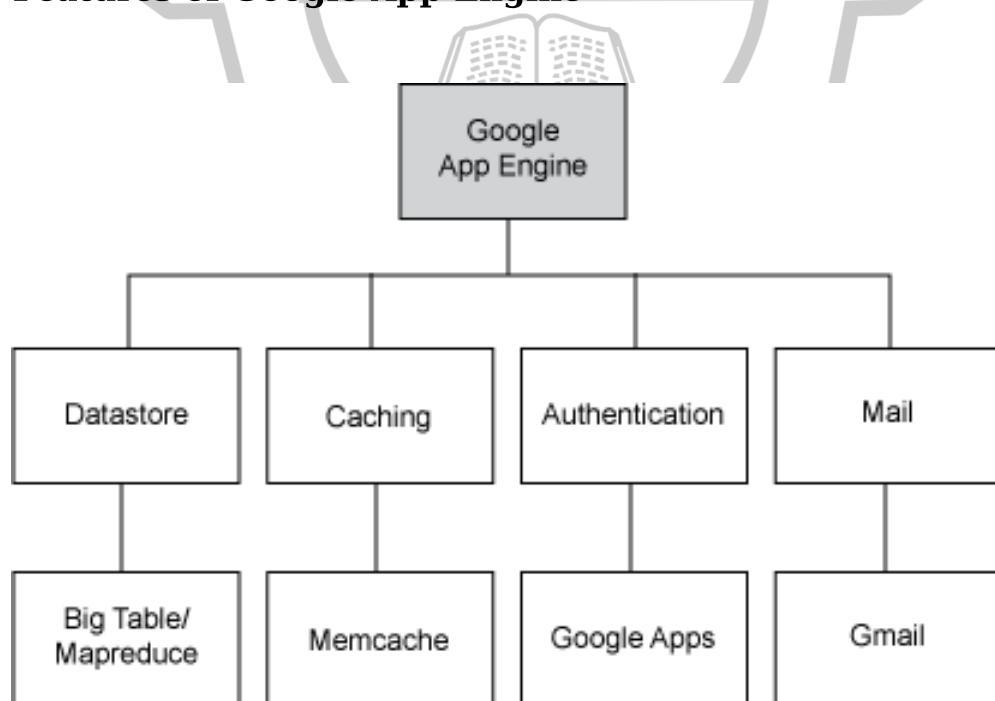
- Google App Engine lets you run your web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve your users.
- You can serve your app from your own domain name (such as <https://www.example.com/>) using Google Apps. Or, you can serve your app using a free name on the appspot.com domain. You can share your application with the world, or limit access to members of your organization.
- Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, you can build your app using standard Java technologies, including the JVM, Java servlets, and the Java programming language—or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a dedicated Python runtime environment, which includes a fast Python interpreter and the Python standard library. The Java and Python runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on the system.
- With App Engine, you only pay for what you use. There are no set-up costs and no recurring fees. The resources your application uses, such as storage and bandwidth, are measured by the gigabyte, and billed at competitive rates. You control the maximum amounts of resources your app can consume, so it always stays within your budget. App Engine costs nothing to get started. All applications can use up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month,

absolutely free. When you enable billing for your application, your free limits are raised, and you only pay for resources you use above the free levels.

## □ Architecture of Google App Engine



## □ Features of Google App Engine



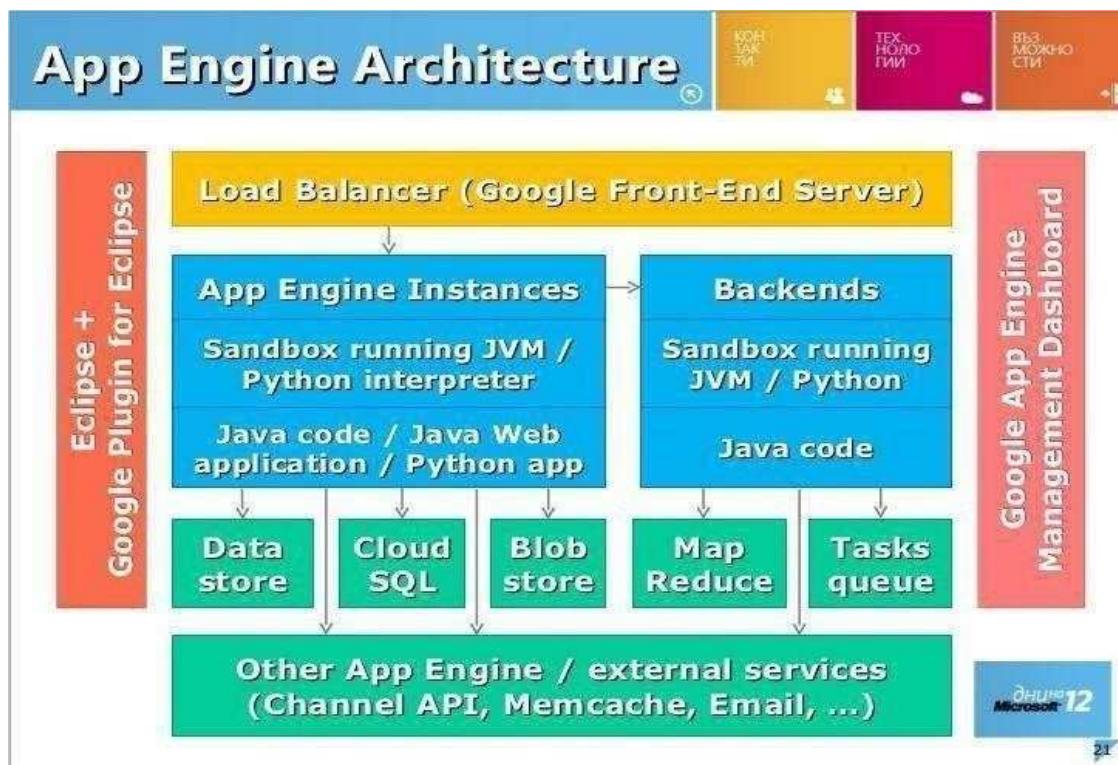
## **□ GAE Application Environment:**

- Google App Engine makes it easy to build an application that runs reliably, even under heavy load and with large amounts of data. App Engine includes the following features:
- Persistent storage with queries, sorting and transactions
- Automatic scaling and load balancing
- APIs for authenticating users and sending email using Google Accounts
- Task queues for performing work outside of the scope of a web request
- Scheduled tasks for triggering events at specified times and regular intervals
- Dynamic web serving, with full support for common web technologies

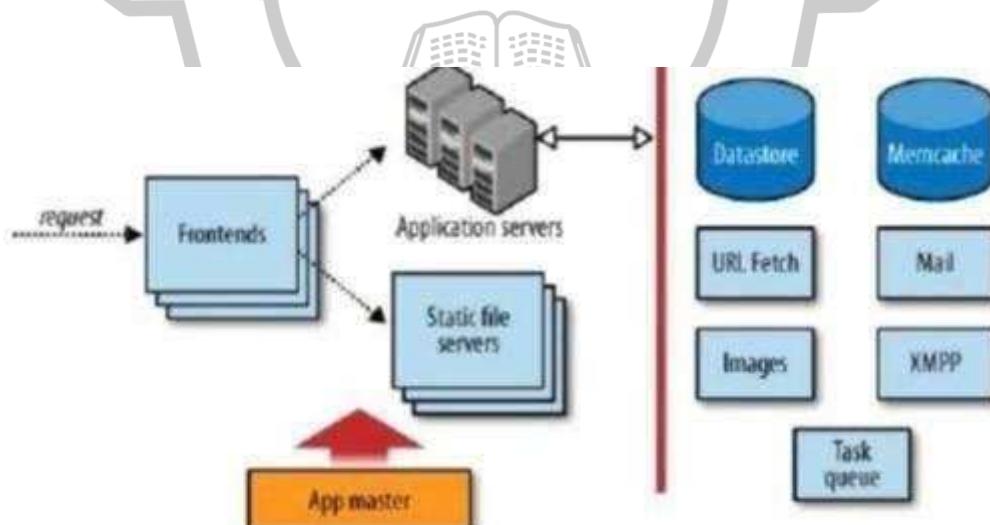
## **□ Java Runtime Environment**

- You can develop your application for the Java runtime environment using common Java web development tools and API standards. Your app interacts with the environment using the Java Servlets standard, and can use common web application technologies such as Java Server Pages
- The Java runtime environment uses Java 6. The App Engine Java SDK supports developing apps using either Java 5 or 6. The environment includes the Java SE Runtime Environment (JRE) 6 platform and libraries. The restrictions of the sandbox environment are implemented in the JVM. An app can use any JVM byte code or library feature, as long as it does not exceed the sandbox restrictions. For instance, byte code that attempts to open a socket or write to a file will throw a runtime exception.
- Your app accesses most App Engine services using Java standard APIs. For the App Engine data store, the Java SDK includes implementations of the Java Data Objects (JDO) and Java Persistence API (JPA) interfaces. Your app can use the JavaMail API to send email messages with the App Engine Mail service. The java.net HTTP APIs access the App Engine URL fetch service.
- App Engine also includes low-level APIs for its services to implement additional adapters, or to use directly from the application. See the documentation for the data store, memcache, URL fetch, mail, images and Google Accounts APIs. Typically, Java developers use the Java programming language and APIs to implement web applications for the JVM. With the use

of JVM-compatible compilers or interpreters, you can also use other languages to develop web applications, such as JavaScript, Ruby.



## Workflow of Google App Engine



## Step1 : Login to [www.cloud.google.com](http://www.cloud.google.com)

The screenshot shows a web browser window with multiple tabs open. The active tab displays the Google Cloud Platform homepage at [www.cloud.google.com](http://www.cloud.google.com). The page features a search bar at the top with the query "google cloud platform". Below the search bar, there's a navigation menu with links like "Cloud Console", "Try it free", "Free Tier", and "Pricing". To the right, there are sections for "Products" and "Google Cloud Platform". A sidebar on the right contains icons for various Google services: App Engine, BigQuery, Kubernetes, and G Suite. Below the main content, there's a "People also ask" section and a "What can you do with Google Cloud Platform?" section featuring a large industrial factory graphic.

## Step2 : Goto Console

The screenshot shows a web browser window displaying the Google Cloud homepage at [cloud.google.com](http://cloud.google.com). The page has a yellow header bar with the text "See the latest news from Next OnAir as it happens. Get the updates." Below this, there's a large call-to-action section with the heading "Solve more with Google Cloud" and a subtext "Meet your business challenges head on with cloud computing services from Google." At the bottom of this section is a blue "Go to console" button. To the right, there's a promotional box for "Google Cloud Next" featuring two speakers, Urs Hözle and Ajay Hammerly. The bottom of the page features a horizontal navigation bar with links: "Modernize your workloads on world-", "Protect your data with multilayered", "Drive decision-making with intelligent", and "Adopt hybrid and multi-cloud". The status bar at the bottom of the screen shows the date as "13-09-2020" and the time as "18:34".

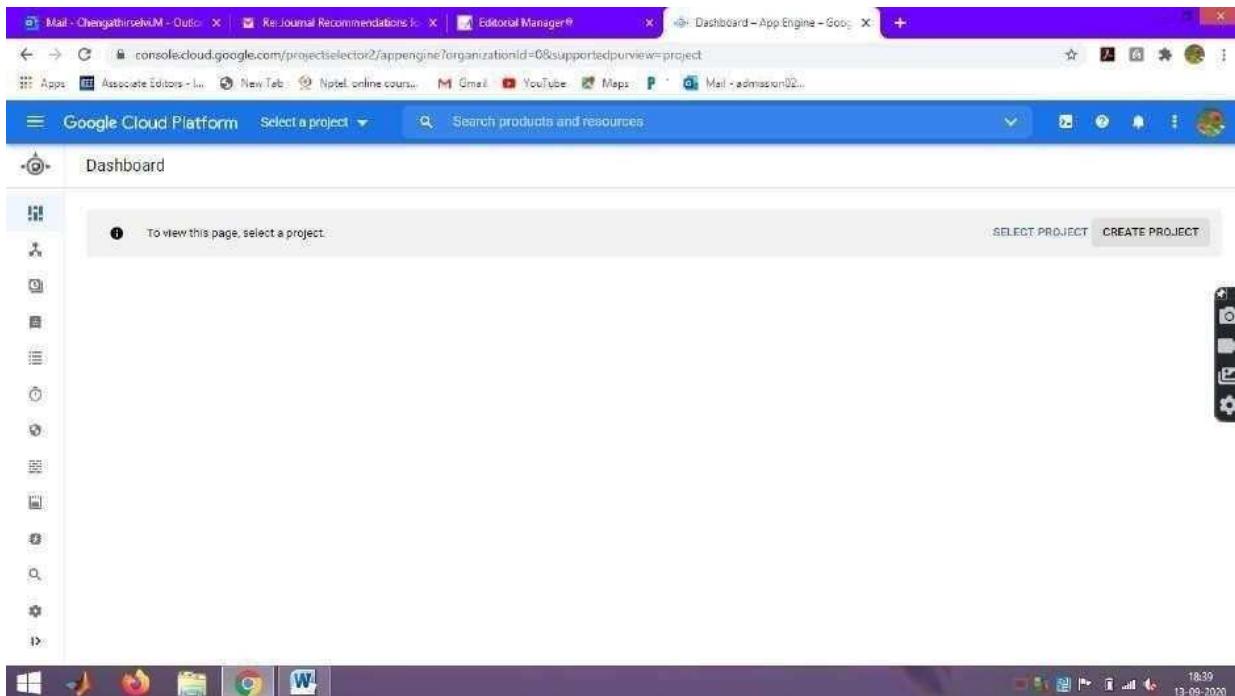
### Step 3 : Google Cloud Platform is shown

The screenshot shows the Google Cloud Platform Home dashboard. On the left, there's a sidebar with 'Project info' (Project name: Koet-kathir, Project ID: koet-kathir, Project number: 625091604908), 'ADD PEOPLE TO THIS PROJECT' button, 'Go to project settings' link, 'Resources' section (Storage: 2 buckets), and 'Trace' section. The main area has a 'API APIs' card showing 'Requests (requests/sec)' from 17:48 to 18:30, with a note: 'No data is available for the selected time frame.' Below it is a 'Go to APIs overview' link. To the right, there are cards for 'Google Cloud Platform status' (All services normal), 'Billing' (Estimated charges INR 0.00 for the period 1–13 Sep 2020), and 'Monitoring' (Set up alerting policies, Create uptime checks). The bottom of the screen shows a Windows taskbar with various icons.

### Step 4 : Click Dashboard in the Google Cloud Platform

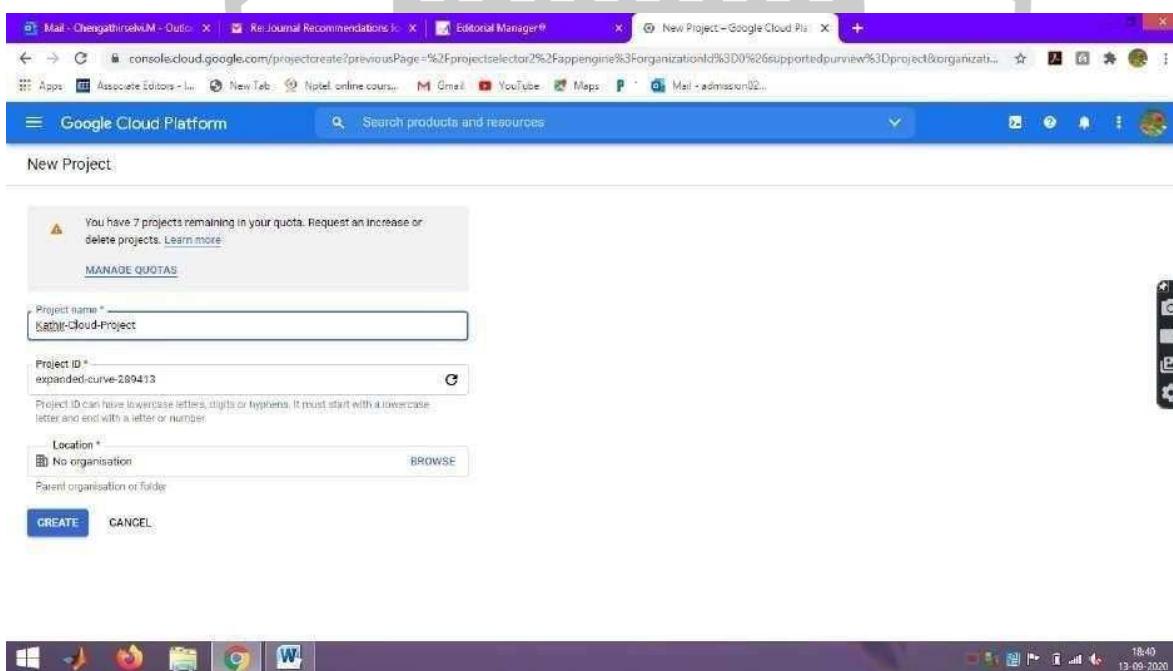
The screenshot shows the Google Cloud Platform Dashboard page. The left sidebar lists various services: Home, Security, IAM & Admin, COMPUTE (App Engine, Compute Engine, Kubernetes Engine, Cloud Functions, Cloud Run), and others like Anthos, Firewall rules, Outlays, Memcache, Search, and Settings. A modal window titled 'Dashboard' is open over the main content. It contains a search bar 'Search products and resources', a 'CREATE FOLDER' button, and tabs for 'MOVES' and 'DELETE'. Below these are sections for 'Charges' and 'Labels'. A message says 'No resource selected'. At the bottom of the modal, there are tabs for 'PERMISSIONS', 'LABELS', and 'ACTIVITY'. A note at the bottom of the modal says 'Please select at least one resource.' The bottom of the screen shows a Windows taskbar with various icons.

## Step 5 : Dashboard in the Google Cloud Platform

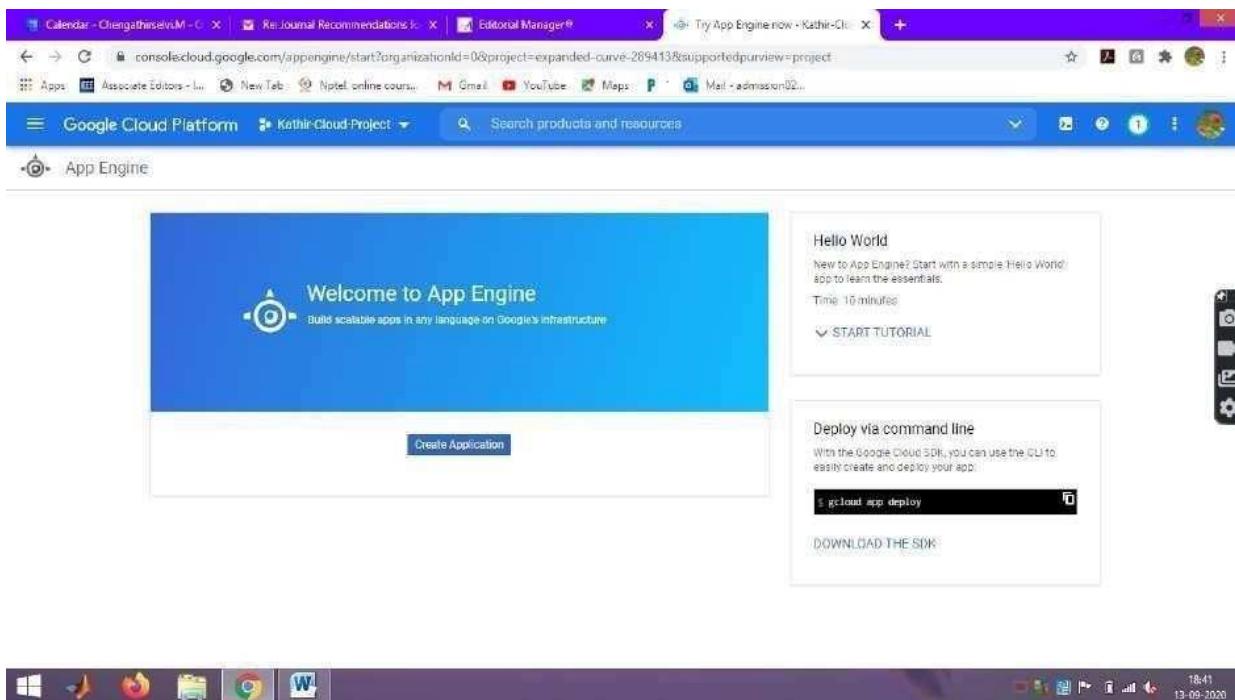


## Step 6 : Click New Project and give unique Project Name.

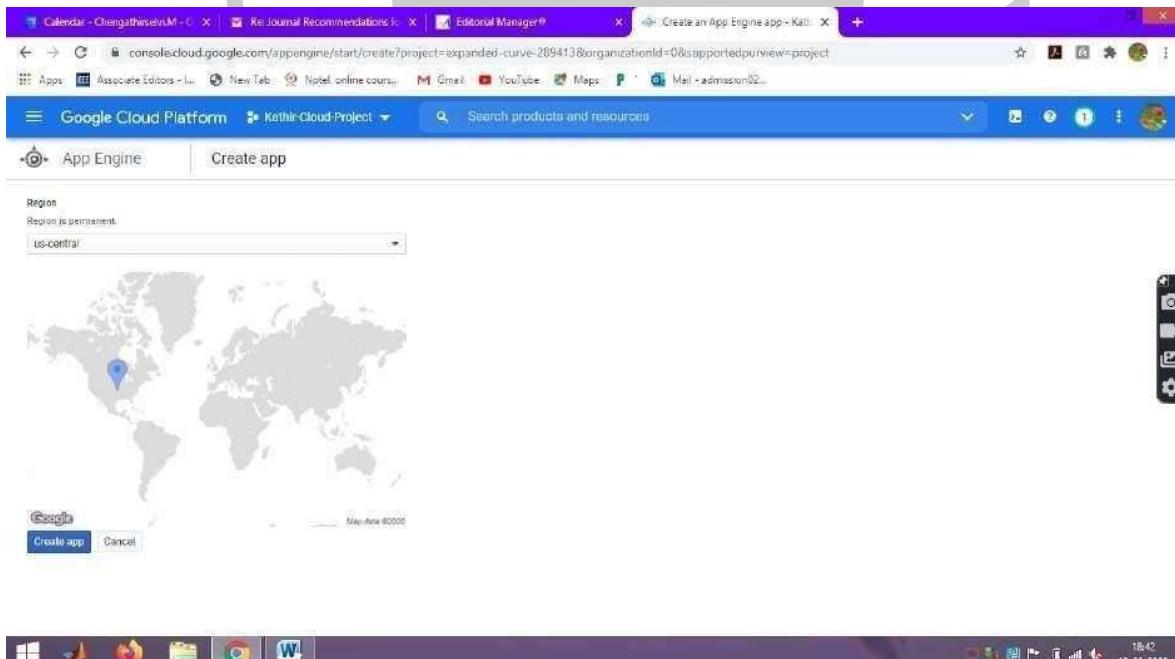
Example : kcet-cloud-project



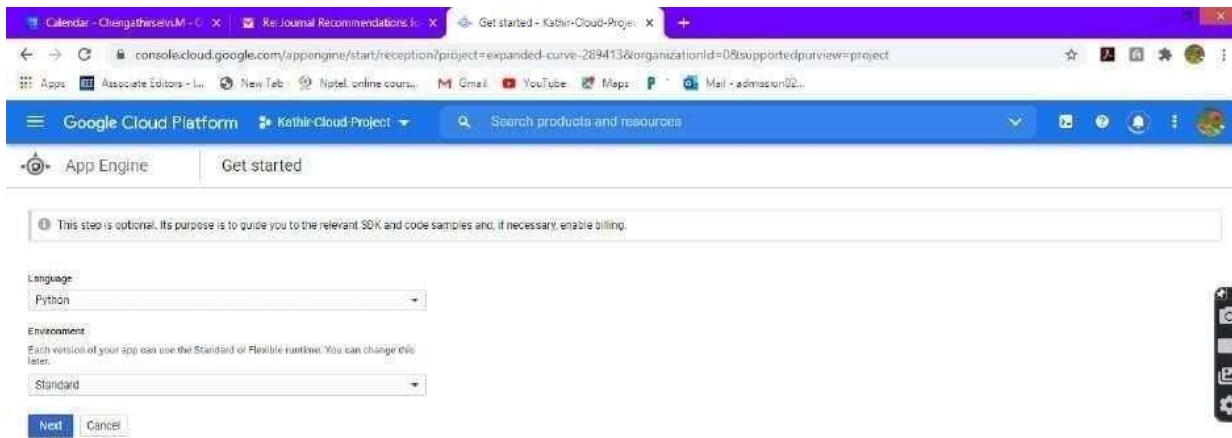
## Step 7 : Google App Engine is initiated



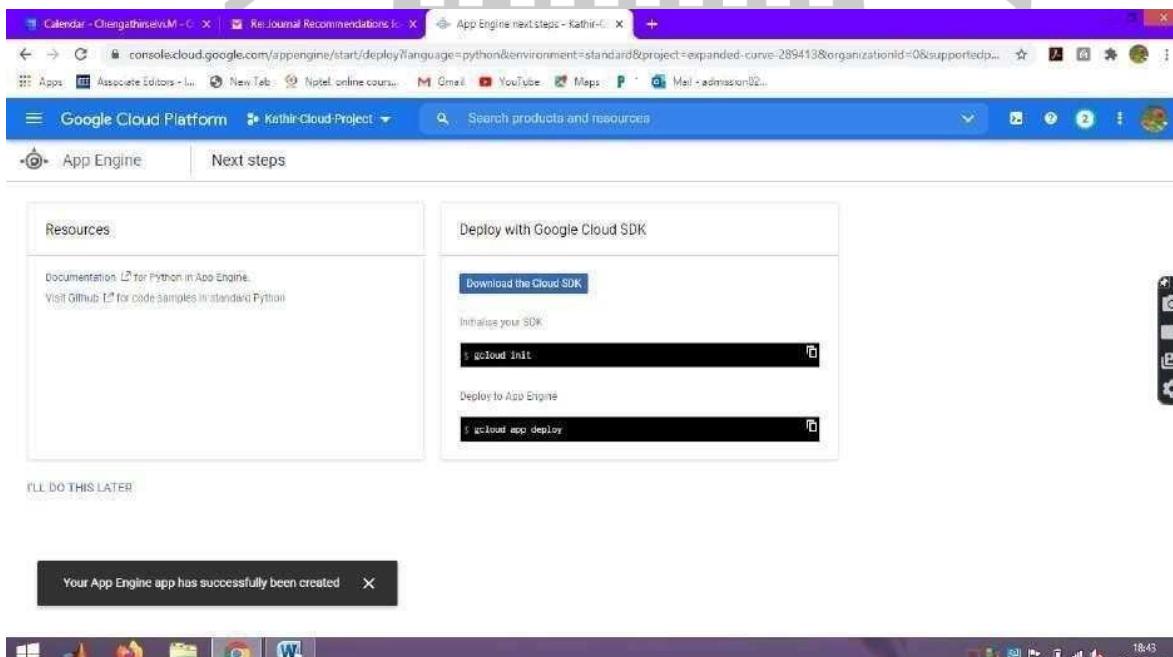
## Step 8 : Click create Application



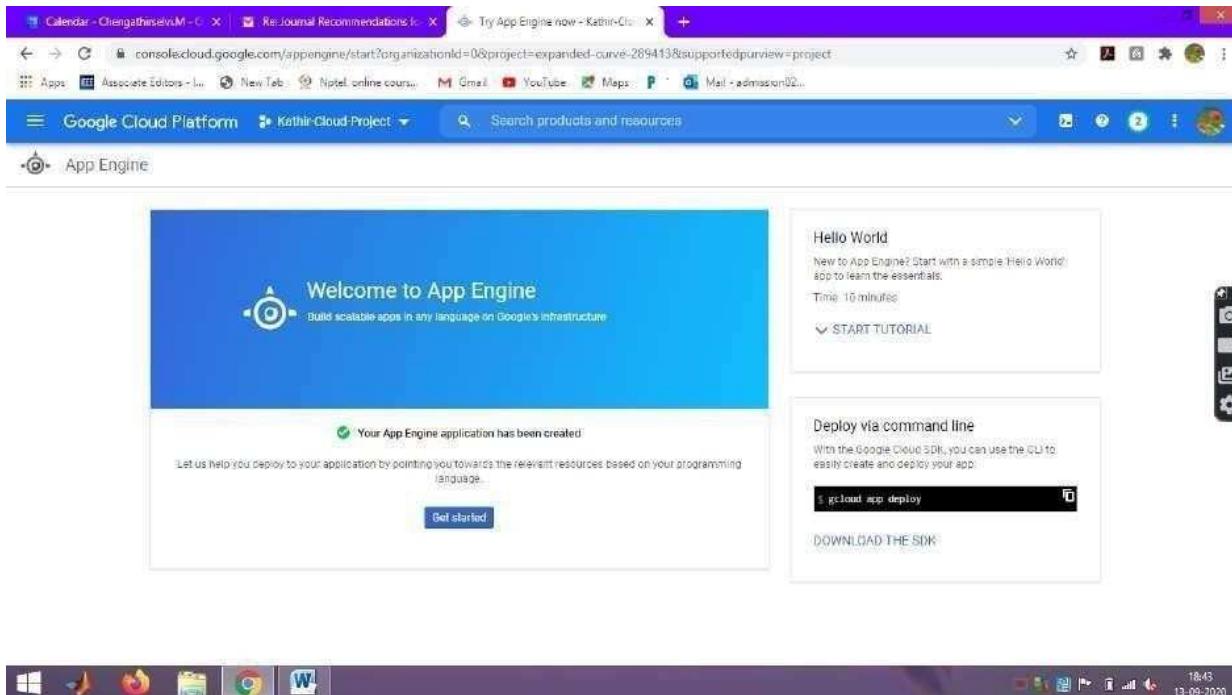
## Step 9 : Create app and Select Language Python



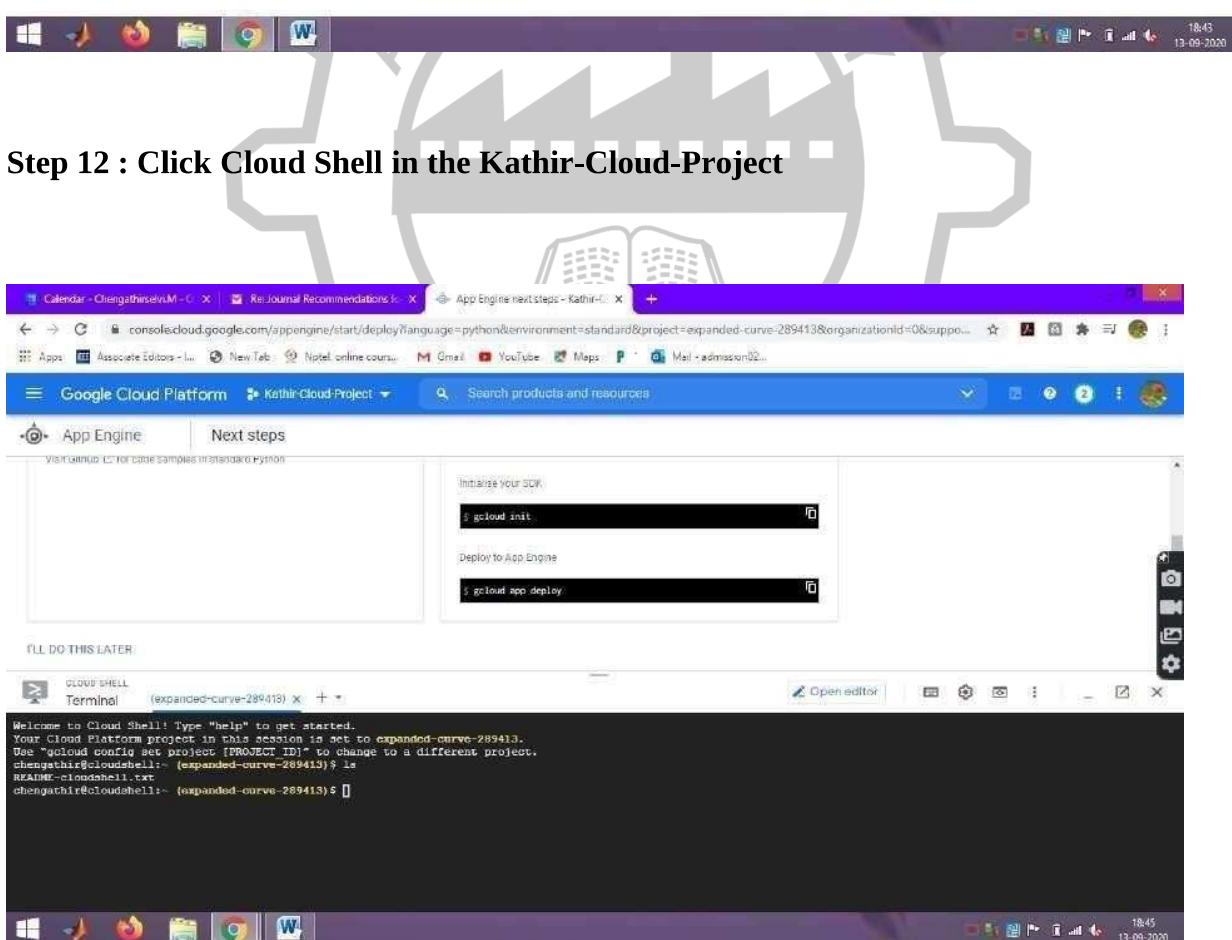
## Step 10 : Python app is created in Google App Engine



## Step 11 : Python app Engine application is created

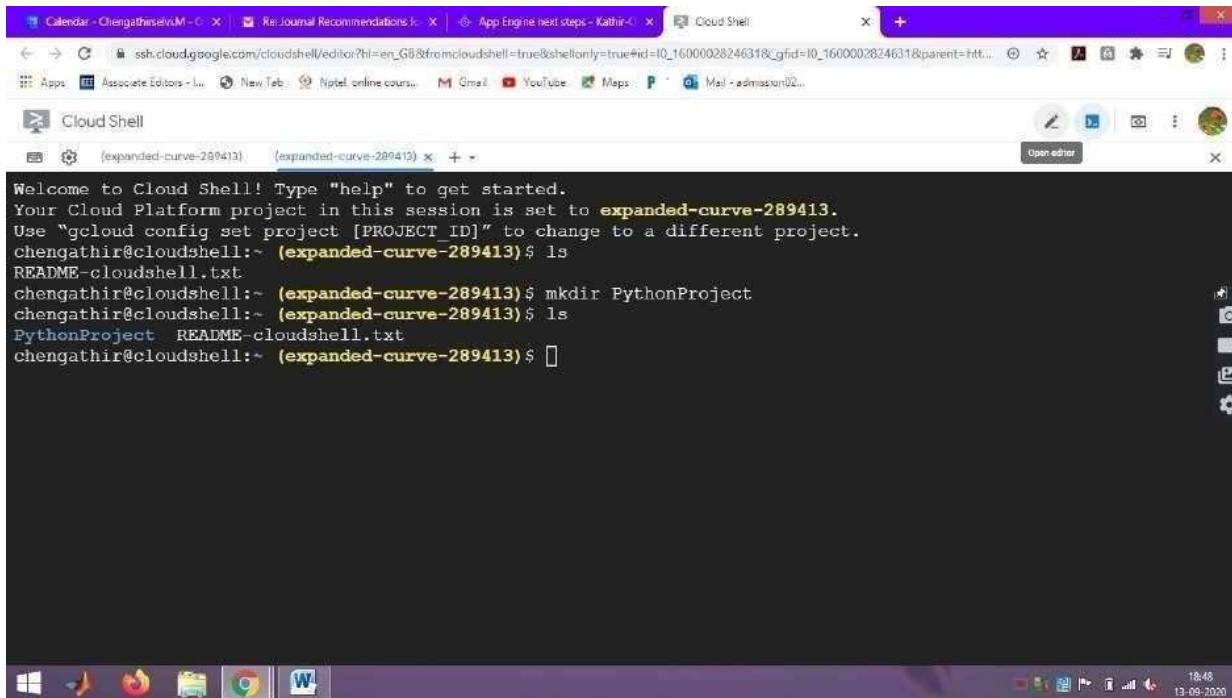


## Step 12 : Click Cloud Shell in the Kathir-Cloud-Project



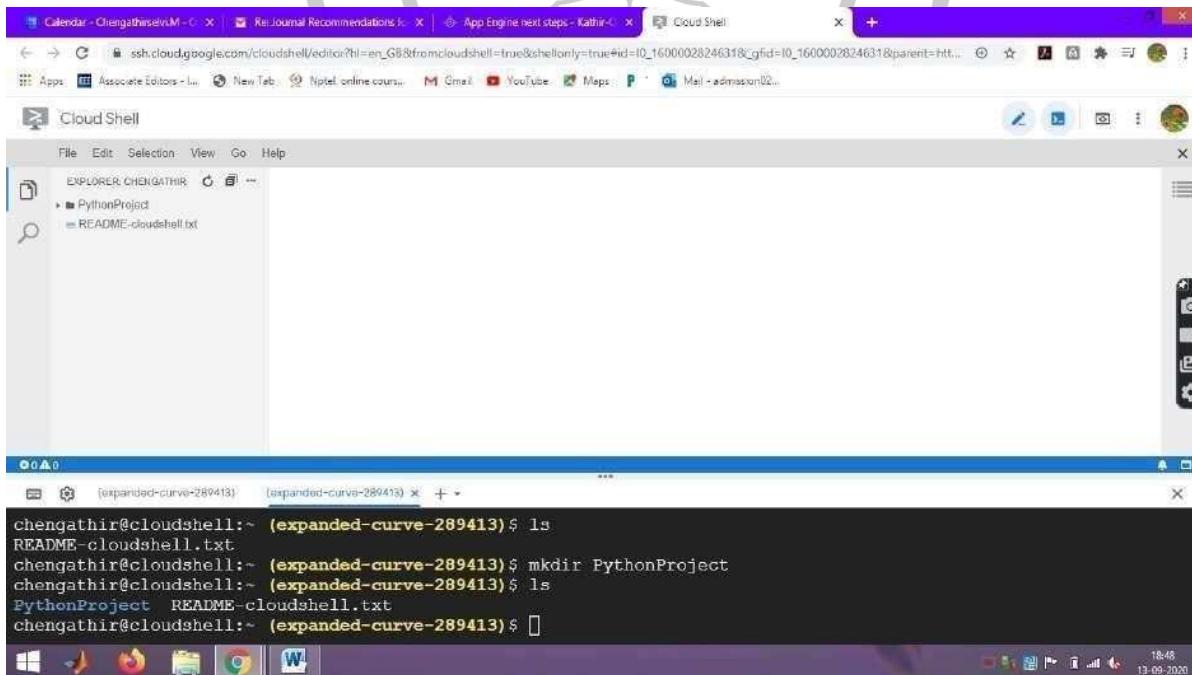
## Step 13 : Create a Directory PythonProject using mkdir command

Syntax : mkdir PythonProject



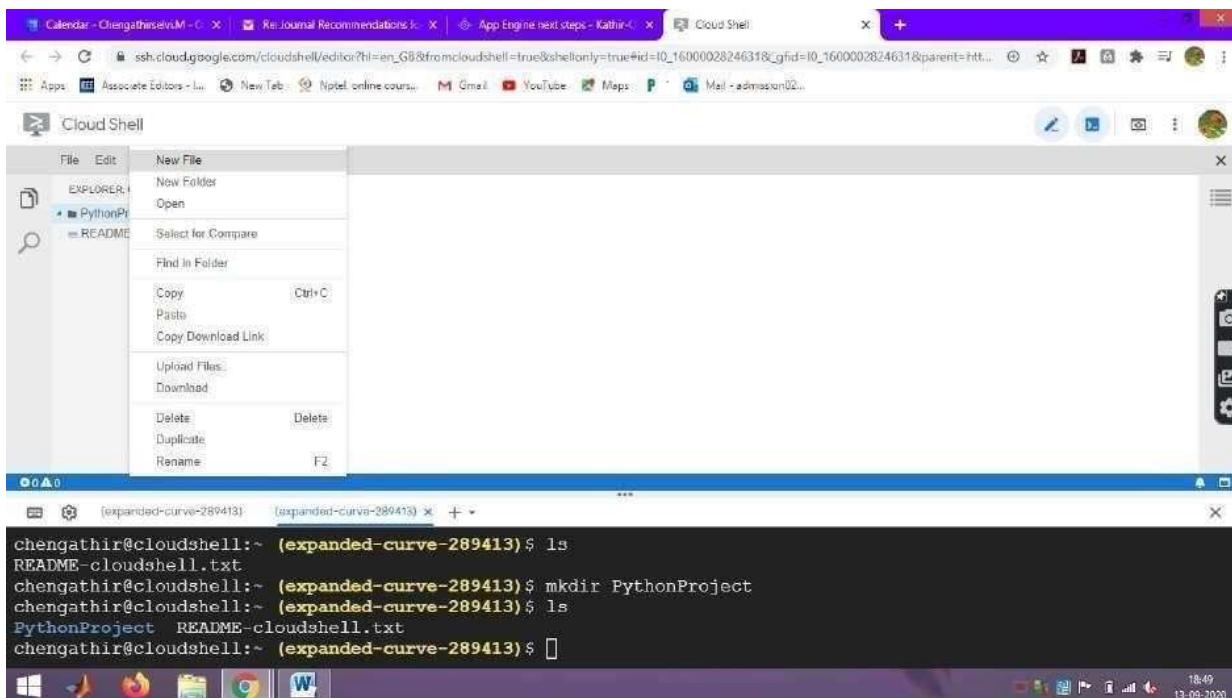
```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to expanded-curve-289413.  
Use "gcloud config set project [PROJECT ID]" to change to a different project.  
chengathir@cloudshell:~ (expanded-curve-289413)$ ls  
README-cloudshell.txt  
chengathir@cloudshell:~ (expanded-curve-289413)$ mkdir PythonProject  
chengathir@cloudshell:~ (expanded-curve-289413)$ ls  
PythonProject README-cloudshell.txt  
chengathir@cloudshell:~ (expanded-curve-289413)$
```

## Step 14 : Click Editor to create Python application

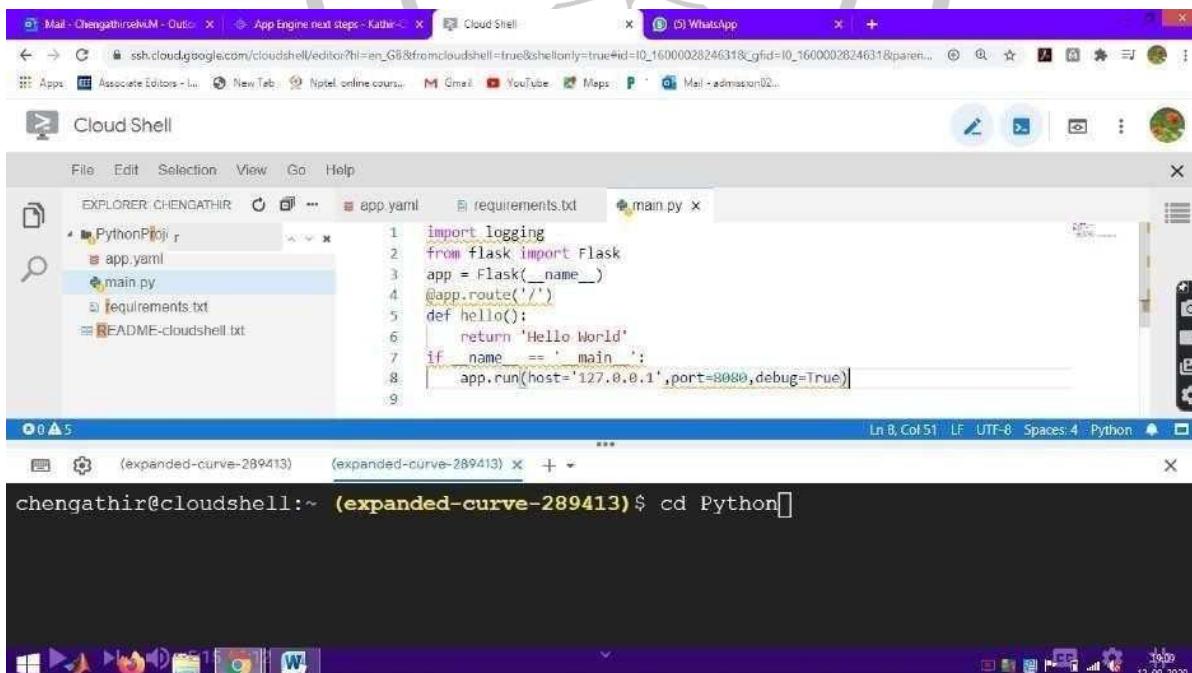


```
File Edit Selection View Go Help  
EXPLORER: CHENGATHIR  
PythonProject  
README-cloudshell.txt  
chengathir@cloudshell:~ (expanded-curve-289413)$ ls  
README-cloudshell.txt  
chengathir@cloudshell:~ (expanded-curve-289413)$ mkdir PythonProject  
chengathir@cloudshell:~ (expanded-curve-289413)$ ls  
PythonProject README-cloudshell.txt  
chengathir@cloudshell:~ (expanded-curve-289413)$
```

## Step 15 : Click New File in the PythonProject Folder (Python file)



## Step 16 : Create main.py file



## main.py file

```
import logging

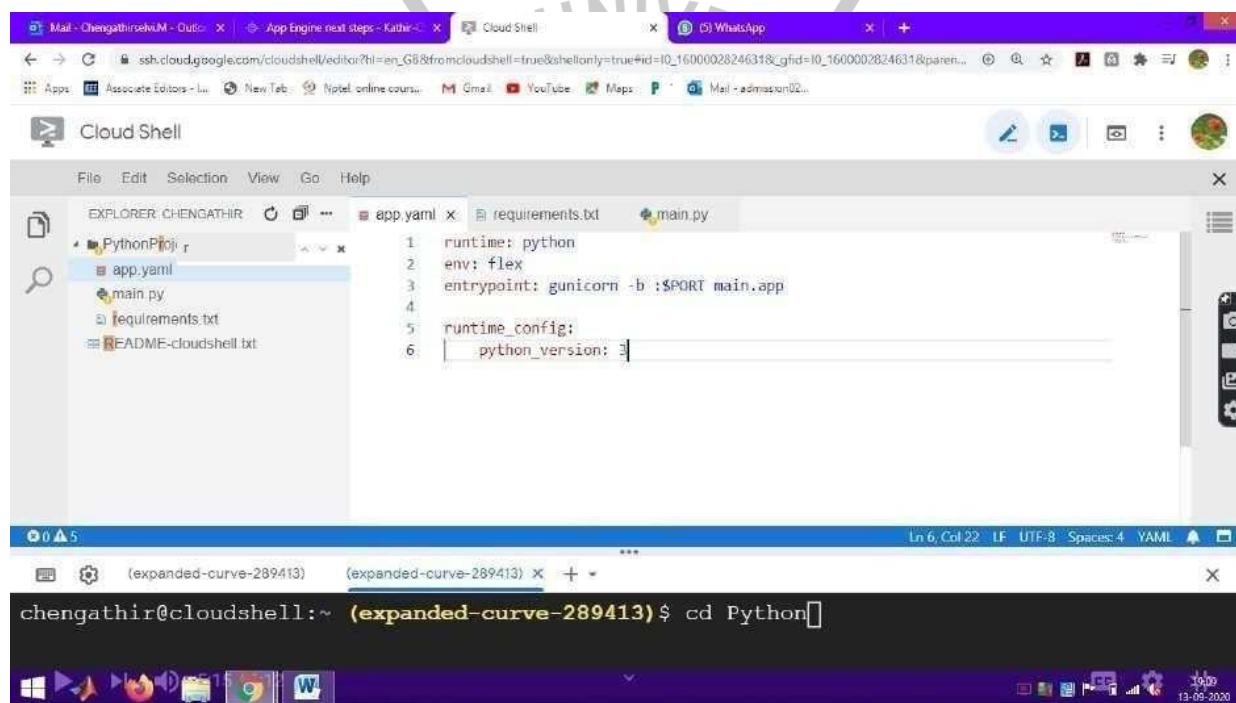
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return 'HelloWorld'

if __name__ == '__main__':
    app.run(host='127.0.0.1', port=8080,
            debug=True)
```

## Step 17 : Create app.yaml file

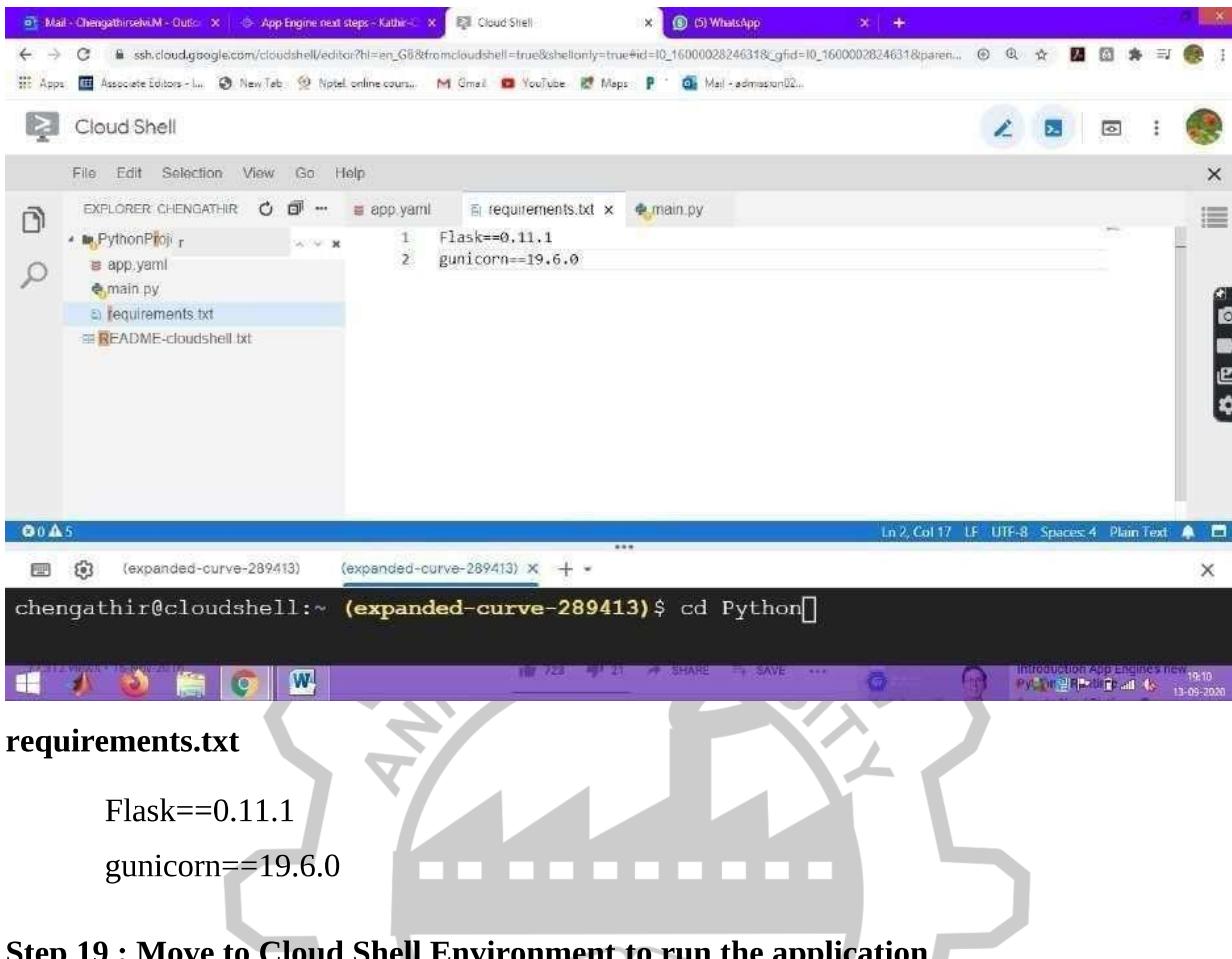


## app.yaml

```
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT main:app

runtime_config:
  python_version: 3
```

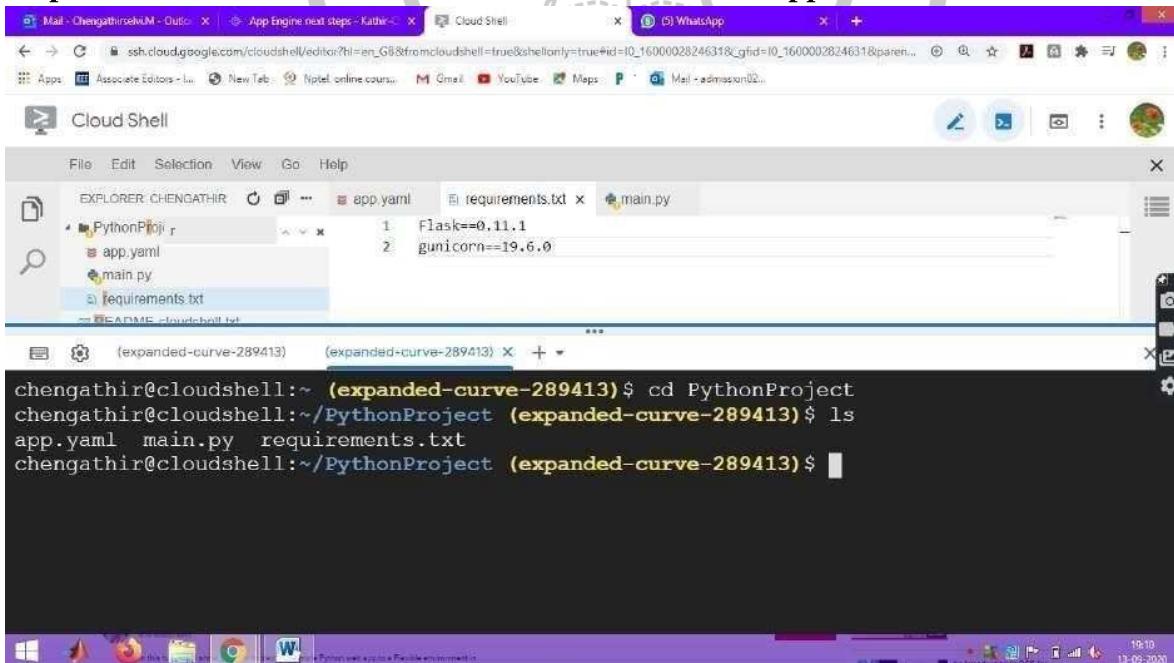
## Step 18 : Create requirements.txt file



The screenshot shows a Cloud Shell environment with a terminal window open. The terminal shows the command `cd PythonProject` being run. Below the terminal, the contents of the `requirements.txt` file are displayed:

```
Flask==0.11.1
gunicorn==19.6.0
```

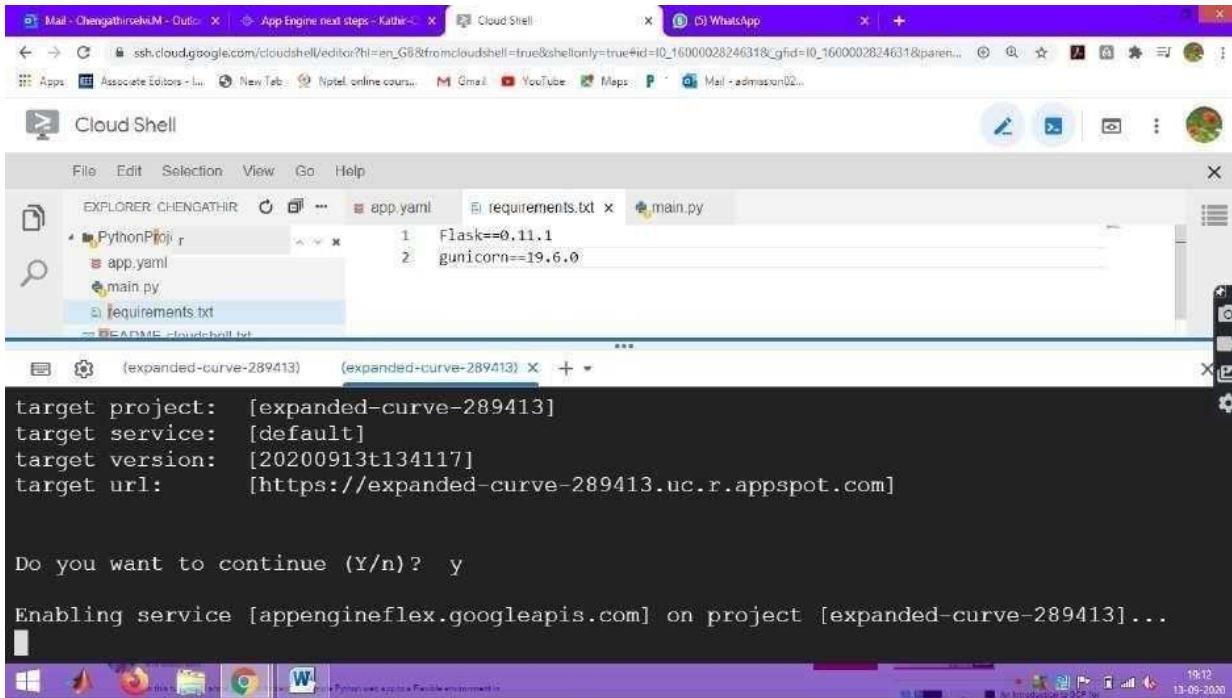
## Step 19 : Move to Cloud Shell Environment to run the application



The screenshot shows a Cloud Shell environment with a terminal window open. The terminal shows the command `ls` being run, which lists the files in the current directory: `app.yaml main.py requirements.txt`.

## Step 20 : Move to Cloud Shell Environment to run the application

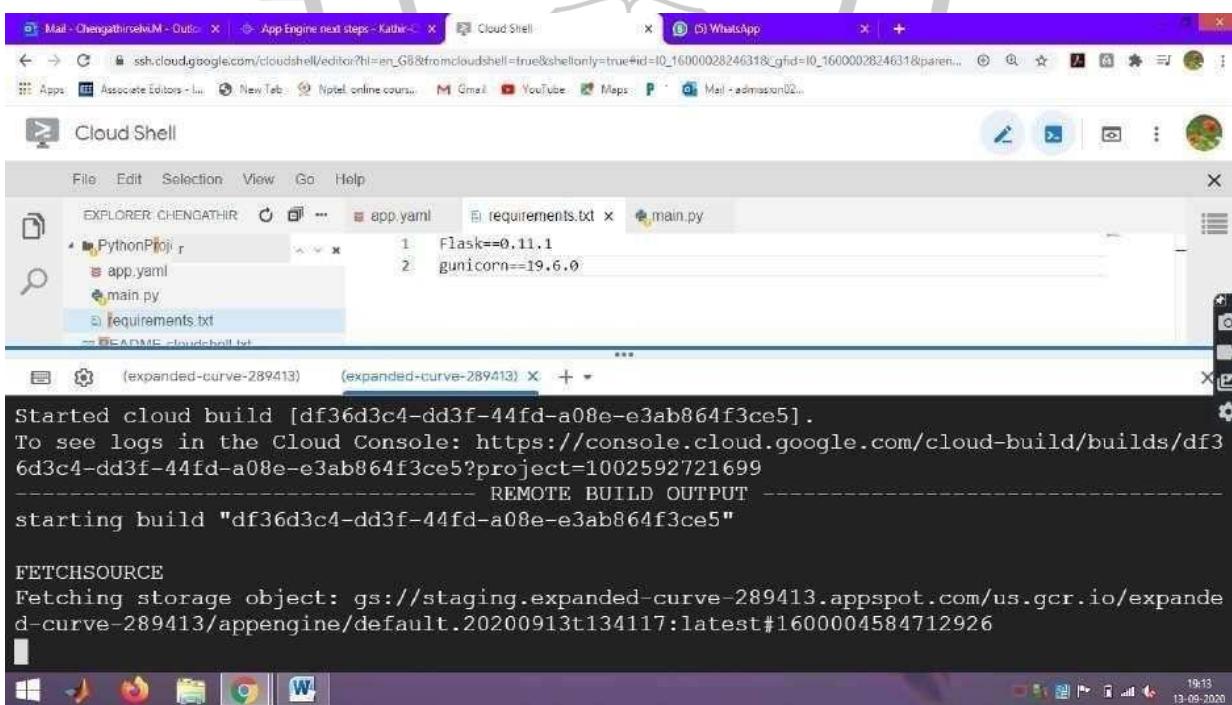
Syntax : **gcloud app deploy**



```
target project: [expanded-curve-289413]
target service: [default]
target version: [20200913t134117]
target url: [https://expanded-curve-289413.uc.r.appspot.com]

Do you want to continue (Y/n)? y
Enabling service [appengineflex.googleapis.com] on project [expanded-curve-289413]...
```

Continue the application. It enable service on the given project



```
Started cloud build [df36d3c4-dd3f-44fd-a08e-e3ab864f3ce5].
To see logs in the Cloud Console: https://console.cloud.google.com/cloud-build/builds/df36d3c4-dd3f-44fd-a08e-e3ab864f3ce5?project=1002592721699
----- REMOTE BUILD OUTPUT -----
starting build "df36d3c4-dd3f-44fd-a08e-e3ab864f3ce5"

FETCHSOURCE
Fetching storage object: gs://staging.expanded-curve-289413.appspot.com/us.gcr.io/expande...
```

It started building the object and fetching the storage object for the created application

Cloud Shell

File Edit Selection View Go Help

EXPLORER: CHENGATHIR app.yaml requirements.txt main.py

PythonProject  
app.yaml  
main.py  
requirements.txt  
README-cloudshell.txt

89e14614ab6b: Layer already exists  
18ecc1cf8b2f: Pushed  
15942b628b0d: Layer already exists  
ed16353646db: Pushed  
latest: digest: sha256:a56ba6724428bce0a0f89a6258cab5d985097ed4dd85027c24bcd176ca06d4b6 size: 3457  
DONE

Updating service [default] (this may take several minutes)...

Windows taskbar icons: File Explorer, Task View, Edge, Chrome, File Manager, Power User, Taskbar settings.

It is updating the service

Cloud Shell

File Edit Selection View Go Help

EXPLORER: CHENGATHIR app.yaml requirements.txt main.py

PythonProject  
app.yaml  
main.py  
requirements.txt  
README-cloudshell.txt

1 runtime: python  
2 env: Flex  
3 entrypoint: gunicorn -b :\$PORT main:app  
4  
5 runtime\_config:  
6 python\_version: 3

DONE

Updating service [default] (this may take several minutes)...done.  
Setting traffic split for service [default]...done.  
Deployed service [default] to [<https://expanded-curve-289413.uc.r.appspot.com>]

You can stream logs from the command line by running:  
\$ gcloud app logs tail -s default

To view your application in the web browser run:  
\$ gcloud app browse

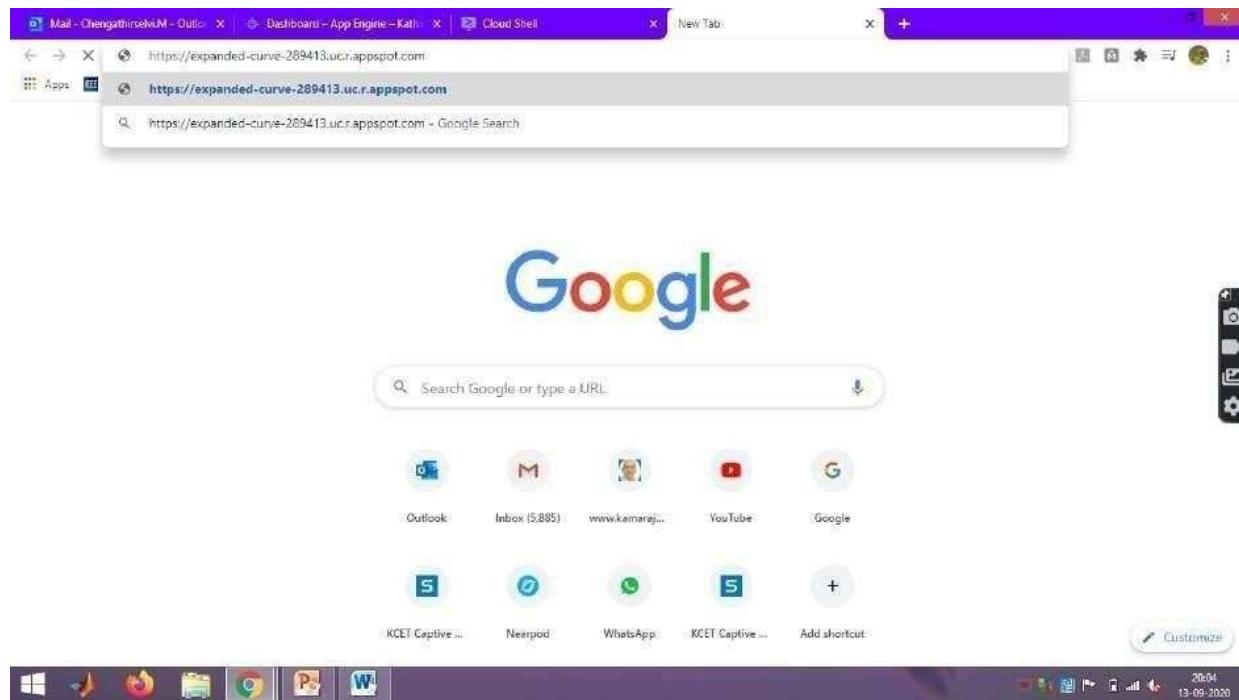
chengathir@cloudshell:~/PythonProject (expanded-curve-289413)\$

Windows taskbar icons: File Explorer, Task View, Edge, Chrome, File Manager, Power User, Taskbar settings.

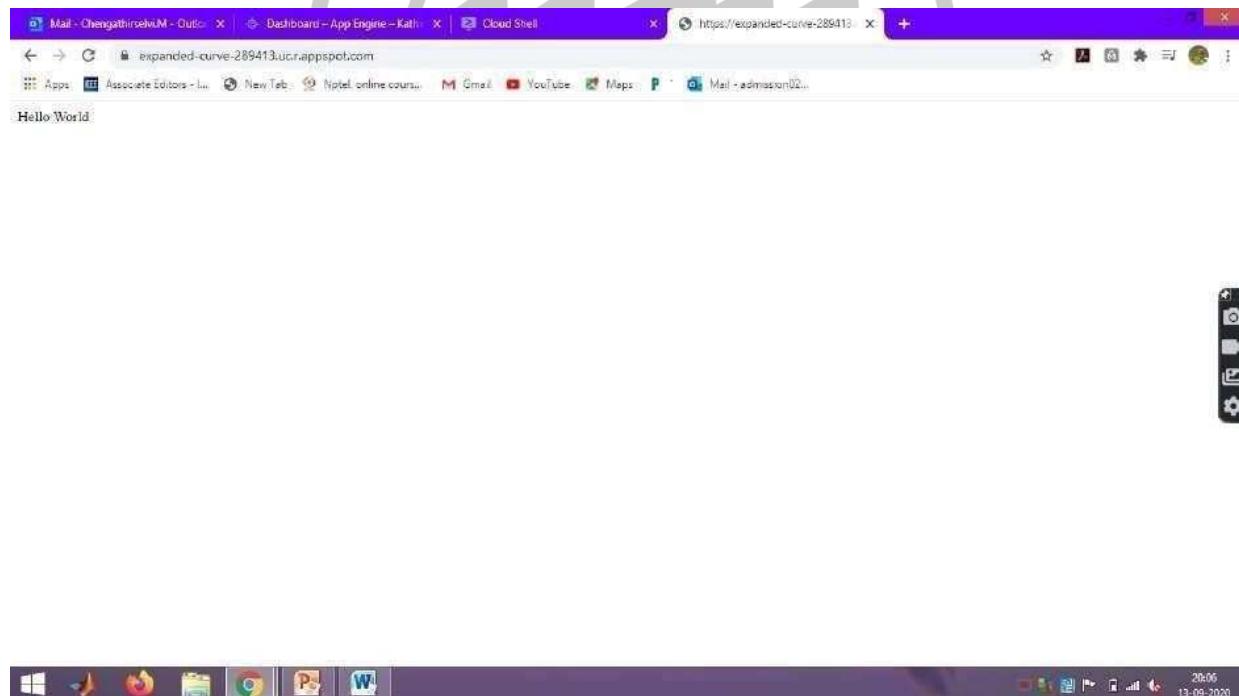
The application is successfully deployed and URL is

<https://expanded-curve-289413.uc.r.appspot.com>

## Step 21 : Run your program in the browser



## Step 22 : Hello World Program is sucessfully run in the browser



## 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 and used GAE Launcher to launch the web applications.

**Ex No. 5a****Simulate a cloud scenario using CloudSim****Introduction:****❖ CloudSim**

- A Framework for modeling and simulation of Cloud Computing Infrastructures and services
- Originally built at the Cloud Computing Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia
- It is completely written in JAVA

**❖ Main Features of CloudSiM**

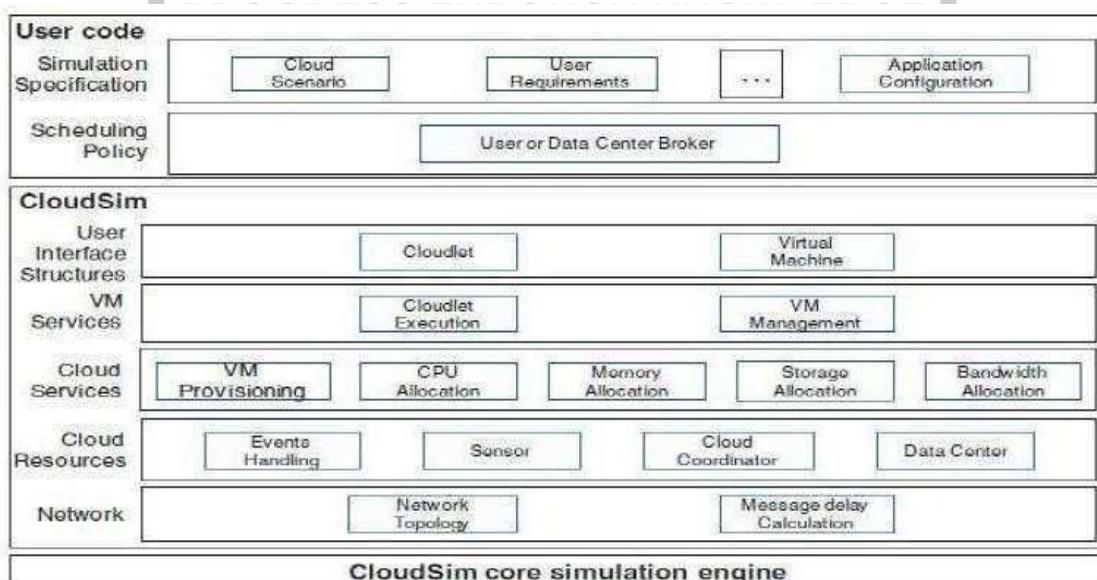
- Modeling and simulation
- Data centre network topologies and message-passing applications
- Dynamic insertion of simulation elements
- Stop and resume of simulation
- Policies for allocation of hosts and virtual machines

**❖ Cloudsim – Essentials**

- JDK 1.6 or above <http://tinyurl.com/JNU-JAVA>
- Eclipse 4.2 or above <http://tinyurl.com/JNU-Eclipse>
- Alternatively NetBeans <https://netbeans.org/downloads>
- Up & Running with cloudsim guide: <https://goo.gl/TPL7Zh>

**❖ Cloudsim-Directory structure**

- cloudsim/ -- top level CloudSim directory
- docs/ -- CloudSim API Documentation
- examples/ -- CloudSim examples
- jars/ -- CloudSim jar archives
- sources/ -- CloudSim source code

**❖ Cloudsim - Layered Architecture**

❖ **Cloudsim - Component model classes**

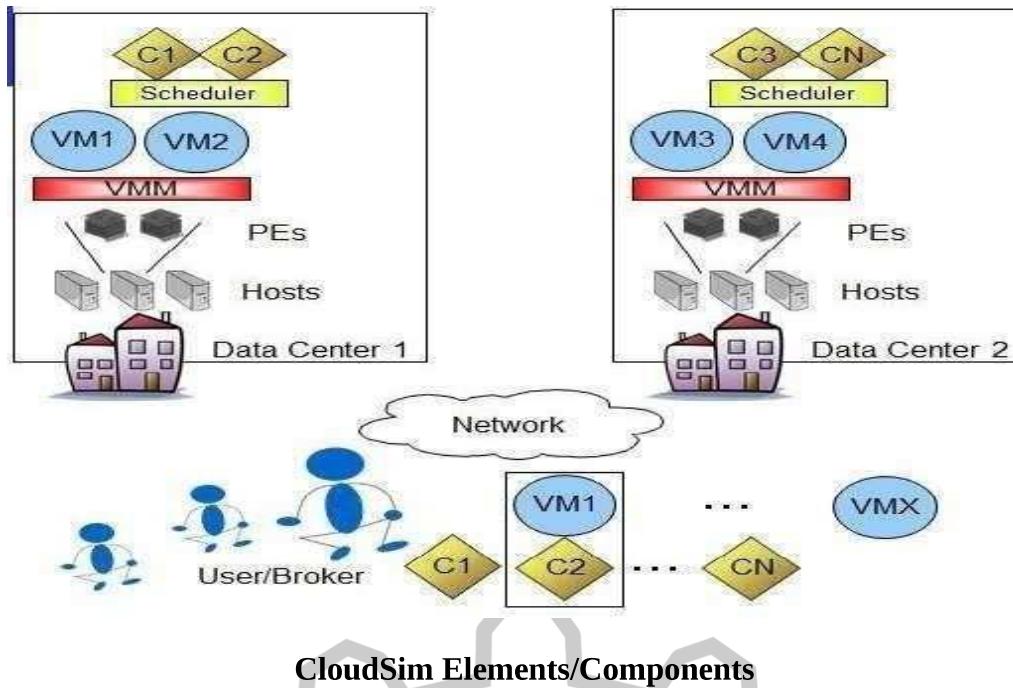
- o CloudInformationService.java
- o Datacenter.java, Host.java, Pe.java
- o Vm.java, Cloudlet.java
- o DatacenterBroker.java
- o Storage.java, HarddriveStorage.java, SanStorage.java

❖ **Cloudsim - Major blocks/Modules**

- o org.cloudbus.cloudsim
- o org.cloudbus.cloudsim.core
- o org.cloudbus.cloudsim.core.predicates
- o org.cloudbus.cloudsim.distributions
- o org.cloudbus.cloudsim.lists
- o org.cloudbus.cloudsim.network
- o org.cloudbus.cloudsim.network.datacenter
- o org.cloudbus.cloudsim.power
- o org.cloudbus.cloudsim.power.lists
- o org.cloudbus.cloudsim.power.models
- o org.cloudbus.cloudsim.provisioners
- o org.cloudbus.cloudsim.util

❖ **Cloudsim - key components**

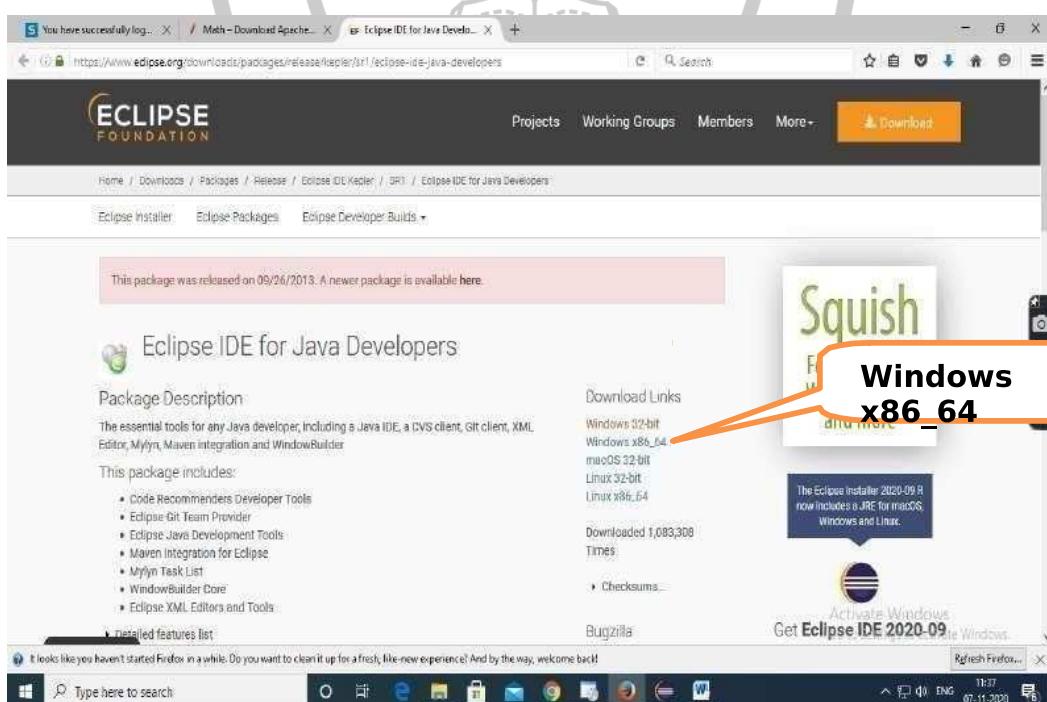
- o Datacenter
- o DataCenterCharacteristics
- o Host
- o DatacenterBroker
- o RamProvisioner
- o BwProvisioner
- o Storage
- o Vm
- o VMAllocationpolicy
- o VmScheduler
- o Cloudlet
- o CloudletScheduler
- o CloudInformationService
- o CloudSim
- o CloudSimTags
- o SimEvent
- o SimEntity
- o CloudsimShutdown
- o FutureQueue
- o DefferedQueue
- o Predicate and associative classes.



### Procedure to import Eclipse, Cloudsim in your system

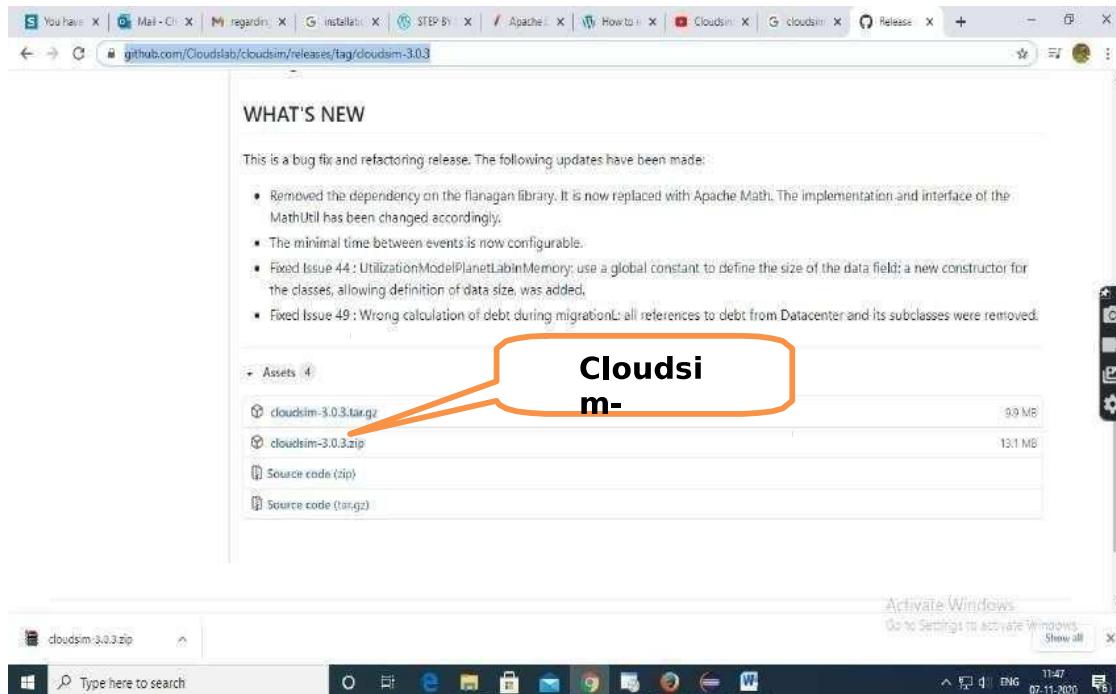
**Step 1:** Link to download Eclipse and download Eclipse for Windows 64bit into your Local machine

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

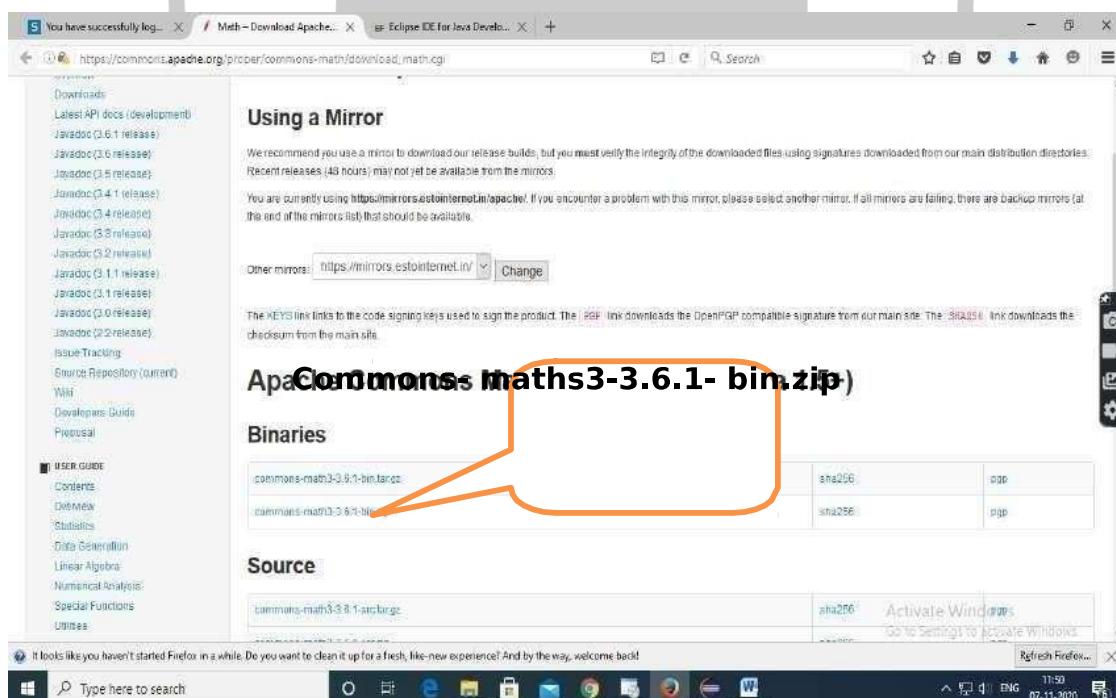


## Step 2: Download cloudsim-3.0.3 from git hub repository in your local machine

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

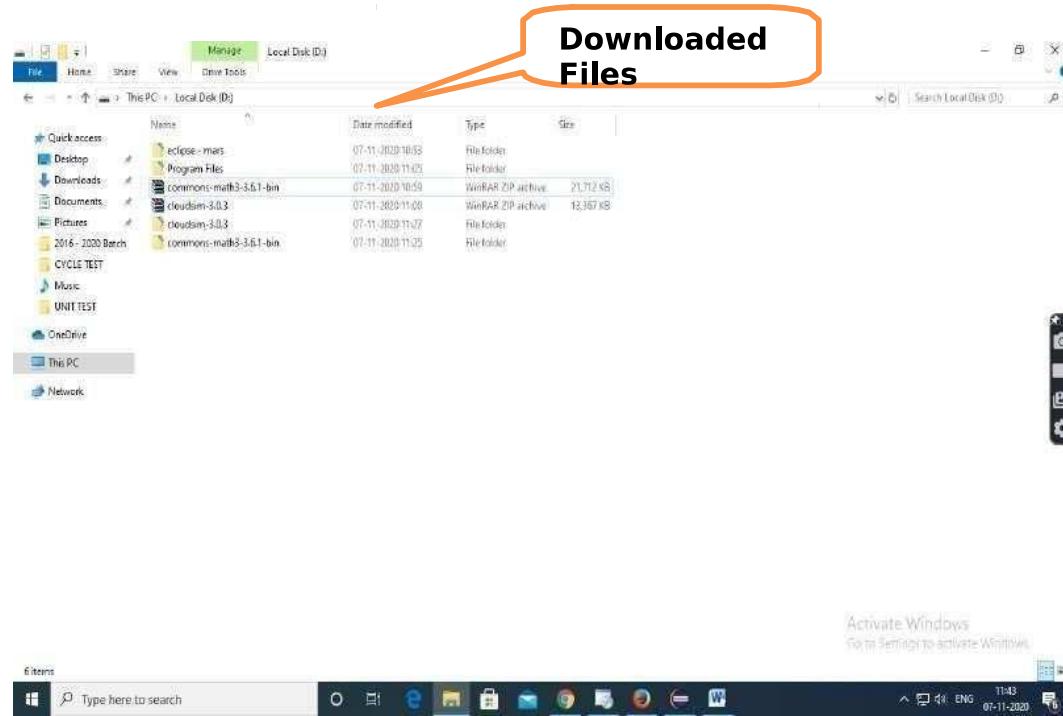


## Step 3: Download commons-math3-3.6.1 from git hub repository in your local machine

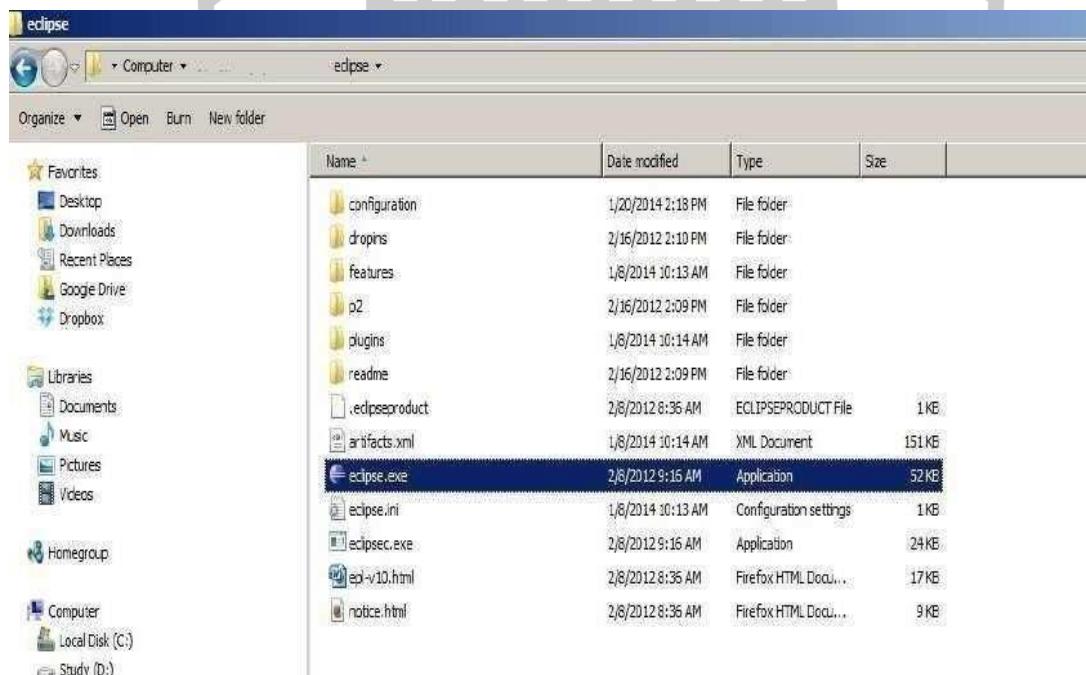


[https://commons.apache.org/proper/commons-math/download\\_math.cgi](https://commons.apache.org/proper/commons-math/download_math.cgi)

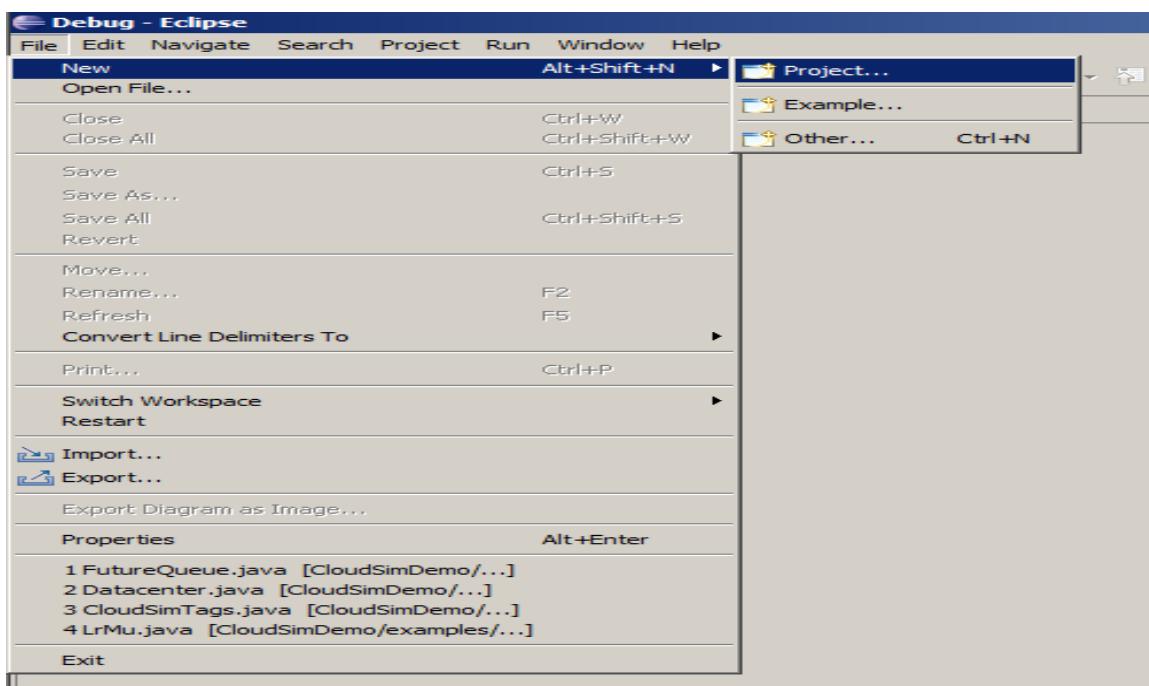
**Step 4:** Downloaded Eclipse, cloudsim-code-master and Apache Commons Math 3.6.1 in your local machine and extract cloudsim-3.0.3 and Apache Commons Math 3.6.1



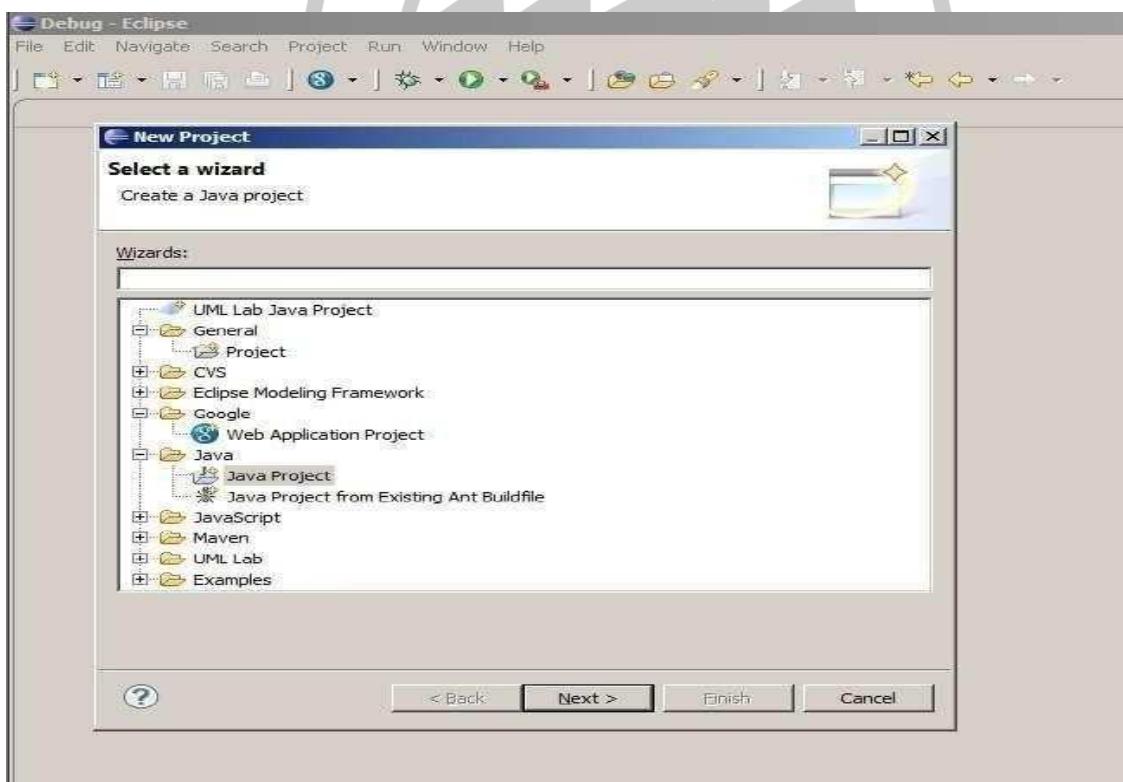
**Step 5:** First of all, navigate to the folder where you have unzipped the eclipse folder and open Eclipse.exe



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

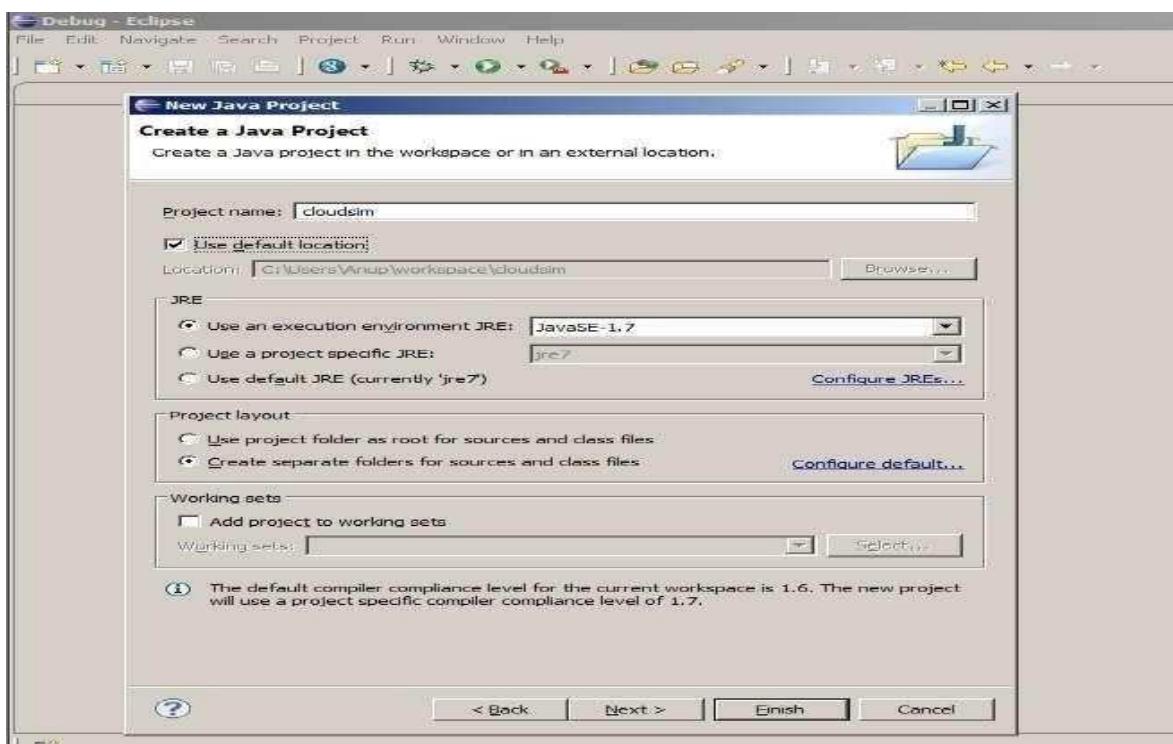


**Step 7:** A New Project wizard should open. There are a number of options displayed and you have to find & select the Java Project option, once done click 'Next'

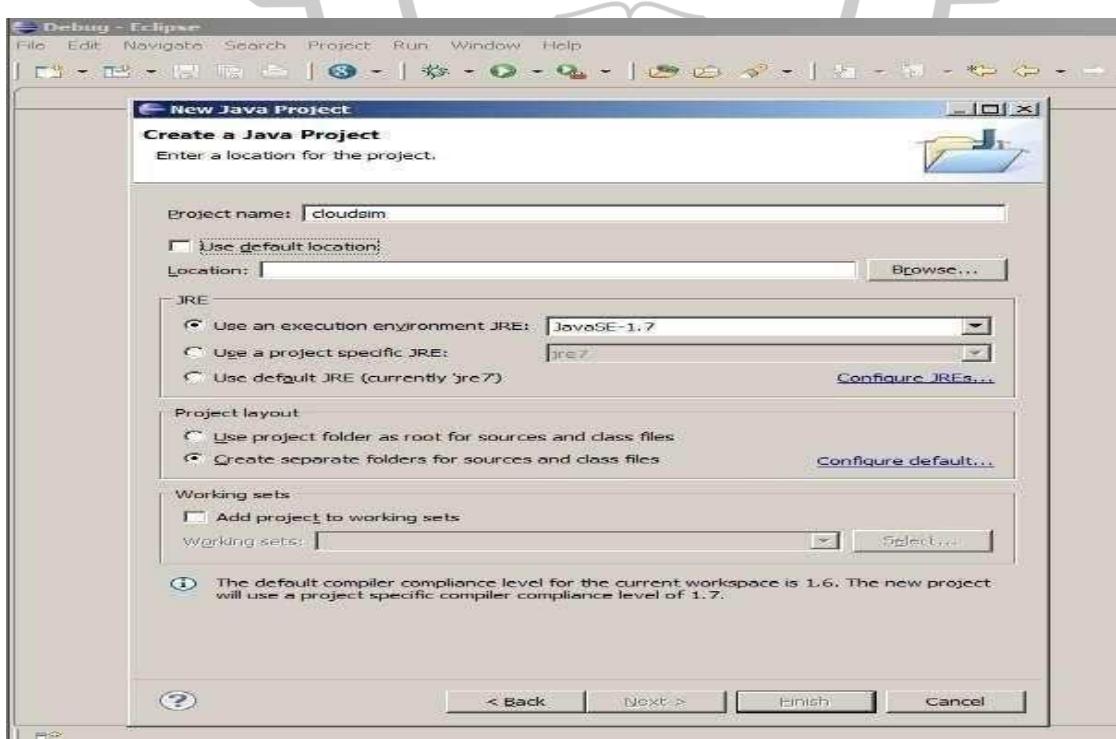


**Step 8:** Now a detailed new project window will open, here you will provide the project name and the path of CloudSim project source code, which will be done as follows:

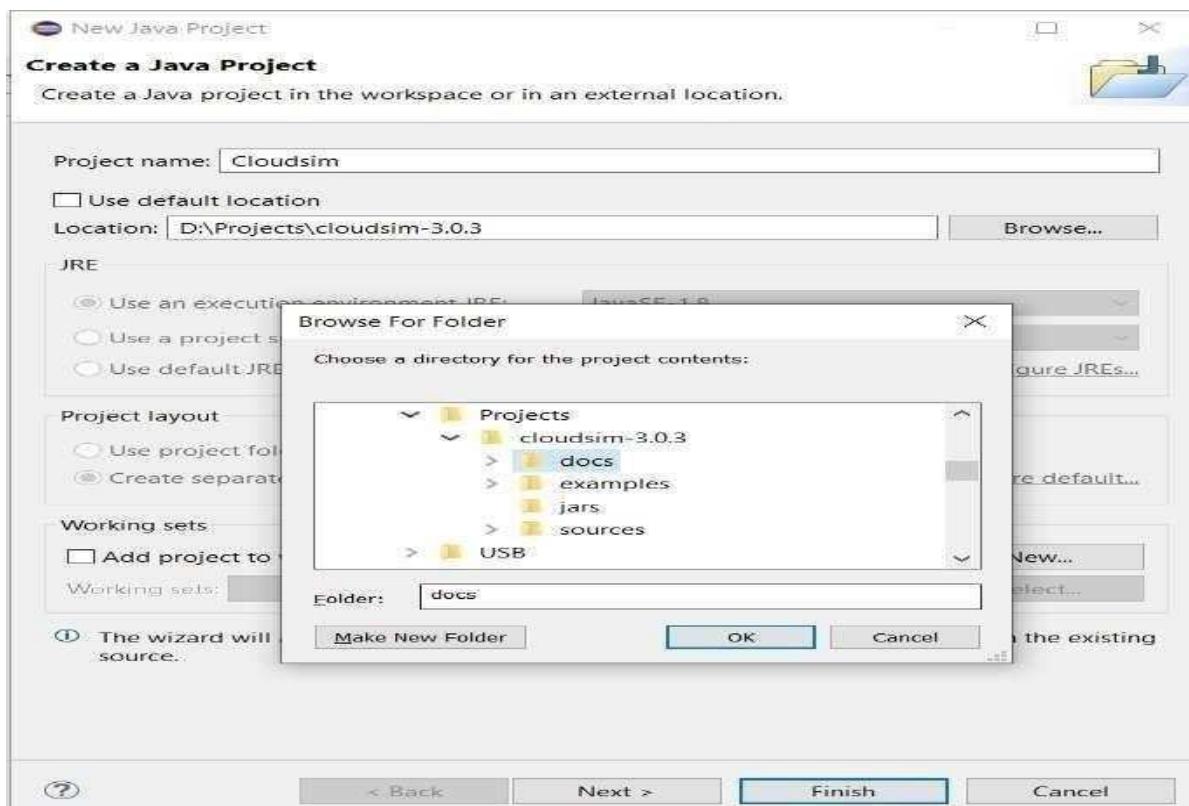
**Project Name: CloudSim.**



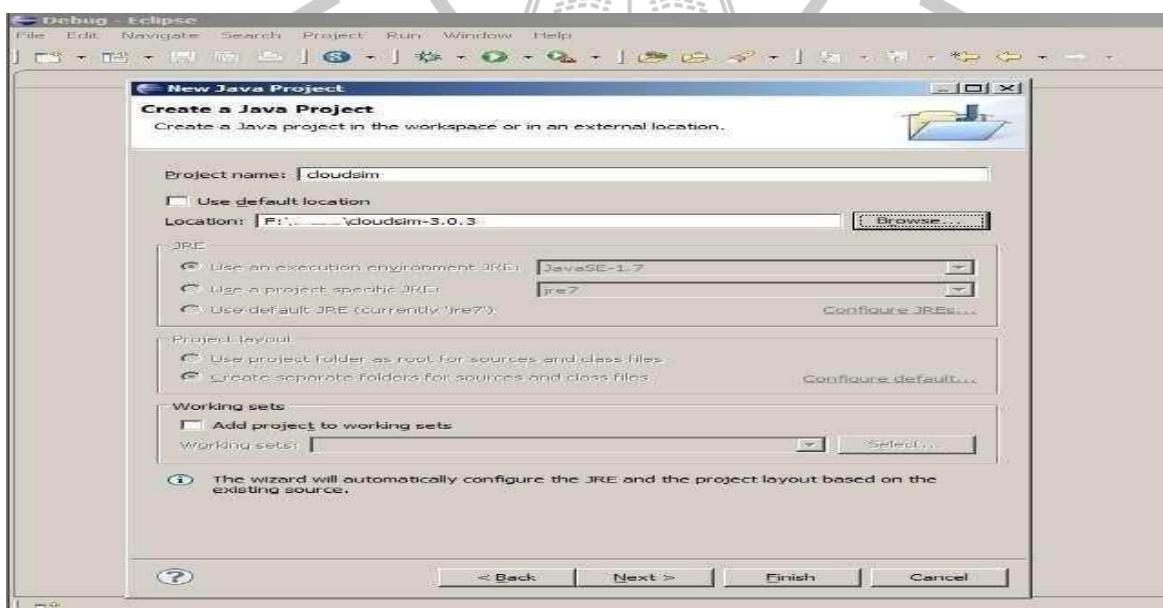
**Step 9:** Unselect the 'Use default location' option and then click on 'Browse' to open the path where you have unzipped the Cloudsim project and finally click Next to set project settings.



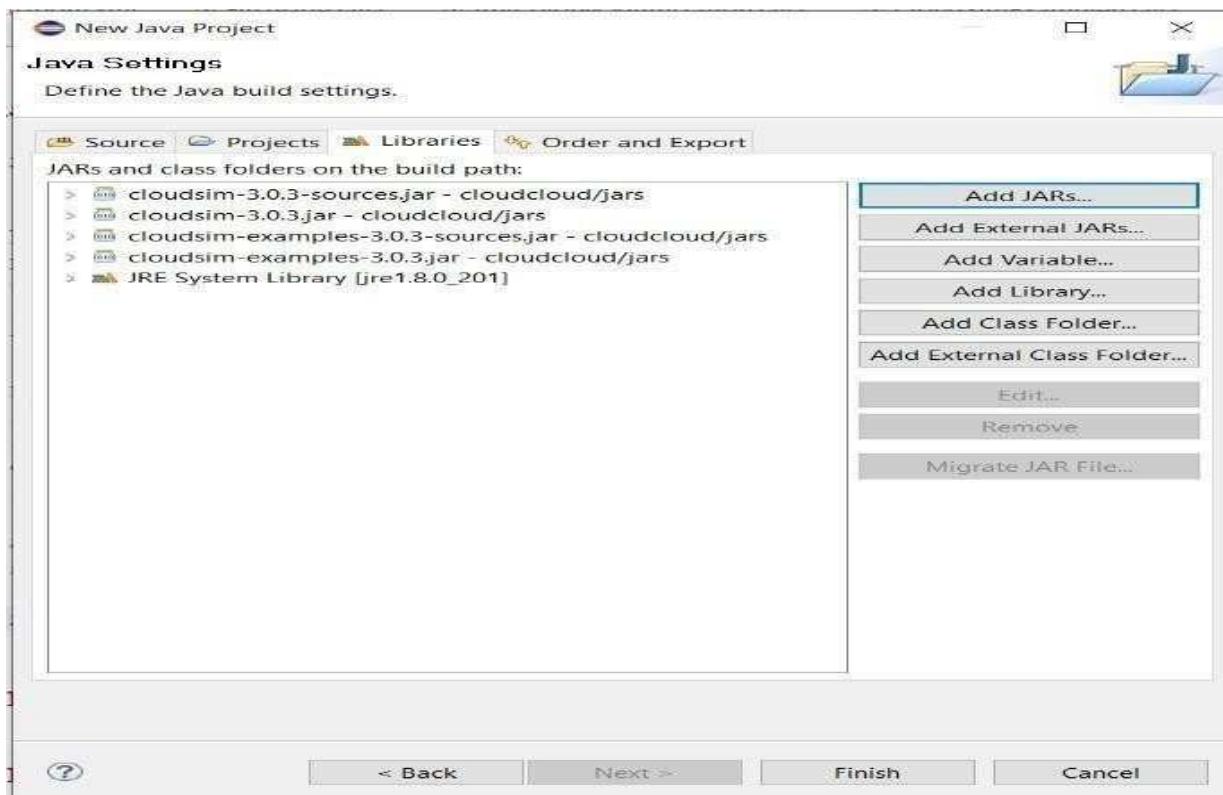
**Step 10:** Make sure you navigate the path till you can see the bin, docs, examplesetc folder in the navigation plane.



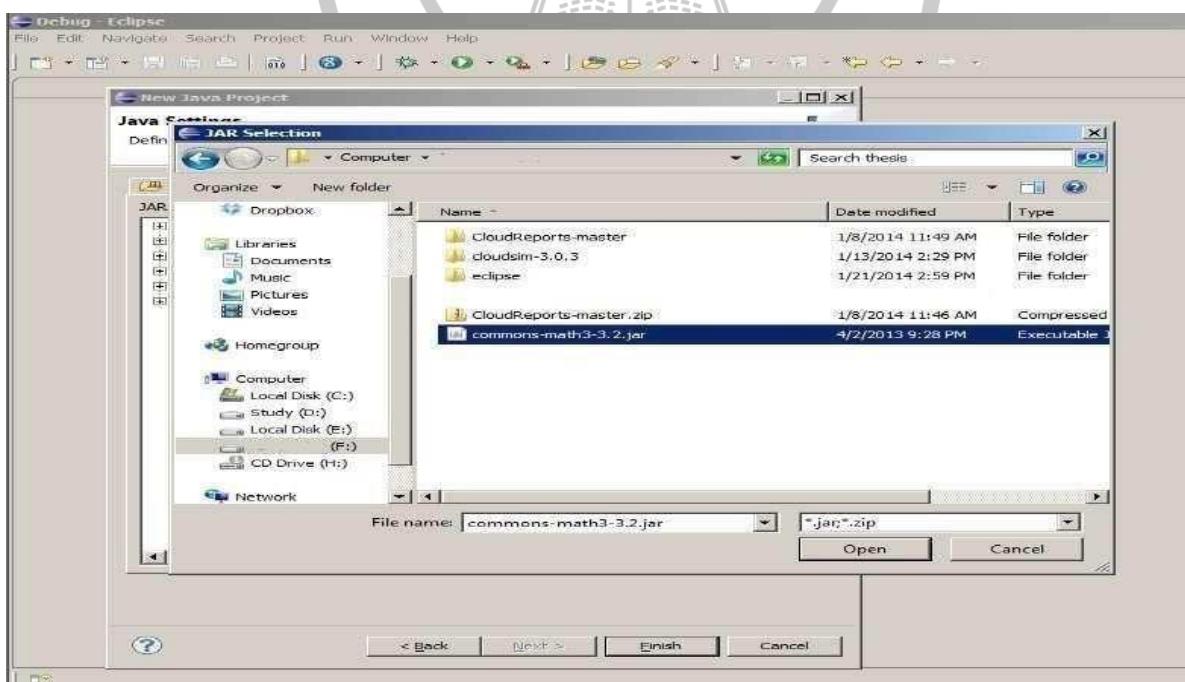
**Step 11:** Once done finally, click 'Next' to go to the next step i.e. setting up of project settings



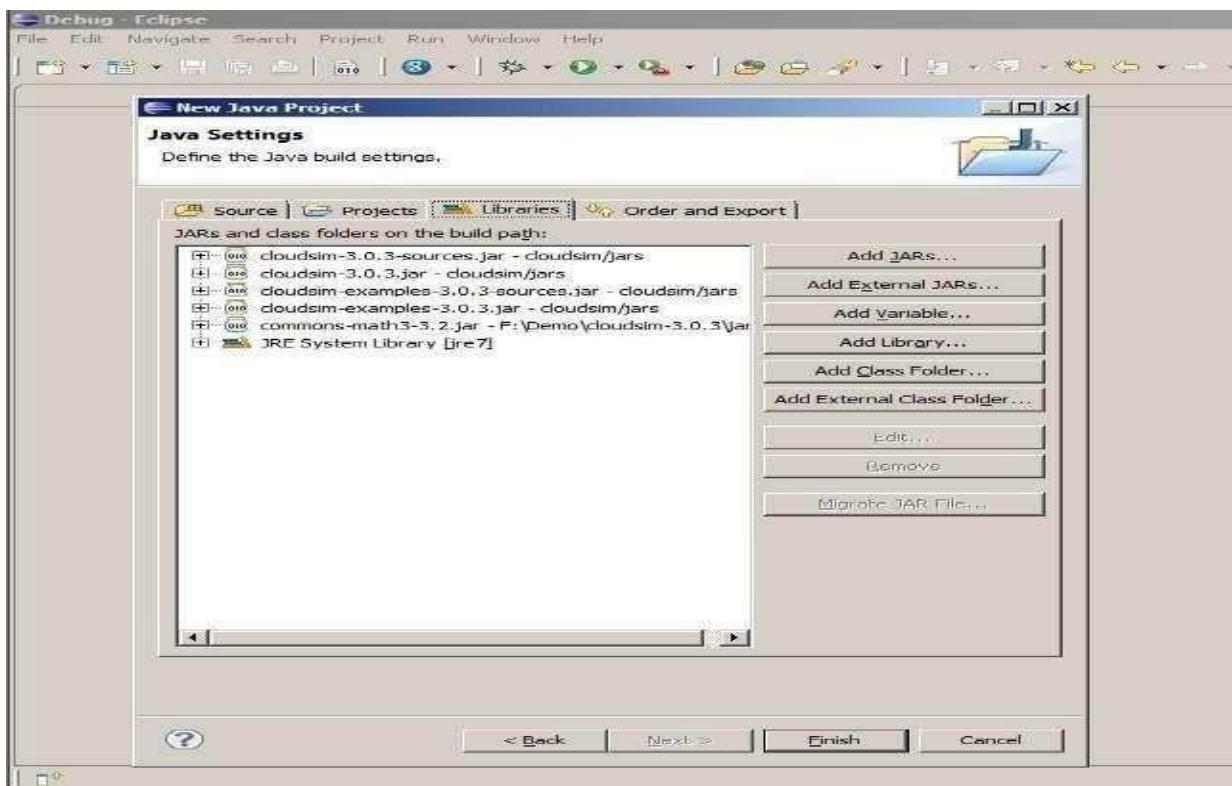
**Step 12:** Now open ‘Libraries’ tab and if you do not find commons-math3-3.x.jar (here ‘x’ means the minor version release of the library which could be 2 or greater) in the list then simply click on Add External Jar (commons-math3-3.x.jar will be included in the project from this step)



**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.

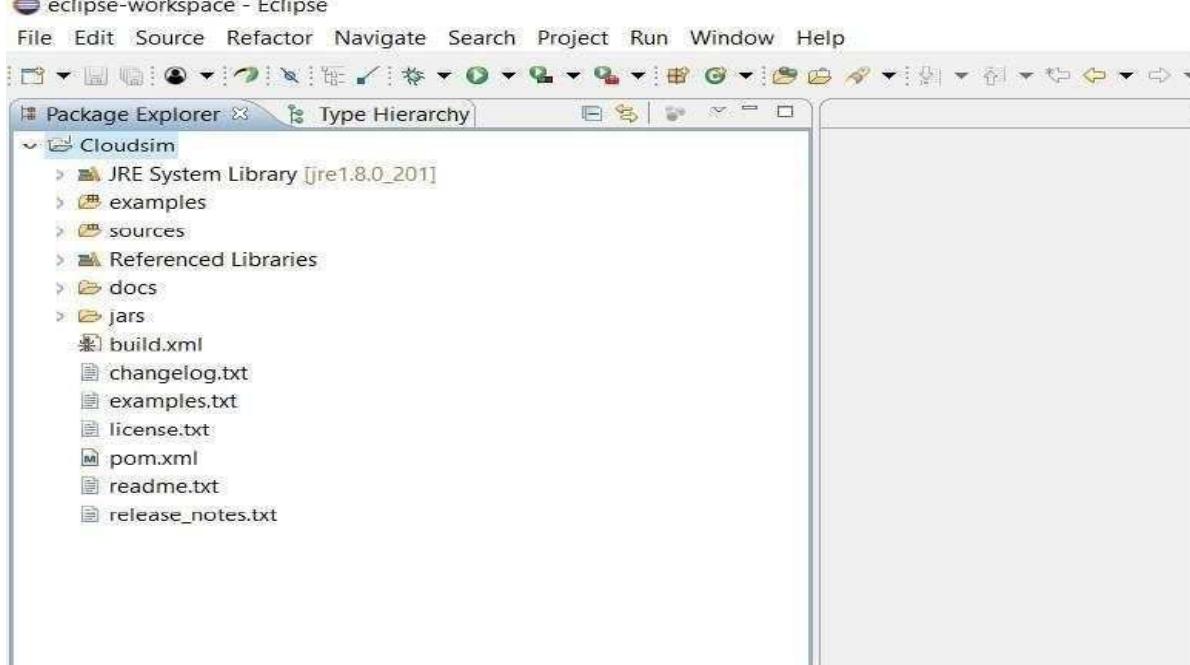


**Step 14:** Ensure external jar that you opened in the previous step is displayed in the list and then click on Finish‘ (your system may take 2-3 minutes to configure the project)



**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 double click to open the CloudsimExample1.java



The screenshot shows the Eclipse IDE interface. The top window is titled "eclipse-workspace - Eclipse" and displays the "Package Explorer" view. In the "examples" package under "org.cloudbus.cloudsim.examples", the file "CloudSimExample1.java" is selected and highlighted in blue. The bottom window is titled "eclipse-workspace - Cloudsim\examples\org\cloudbus\cloudsim\examples\CloudSimExample1.java - Eclipse" and shows the Java code for "CloudSimExample1". The code is as follows:

```

1 package org.cloudbus.cloudsim.examples;
2
3 /**
4 * Title: CloudSim Toolkit
5 */
6
7 /**
8 * A simple example showing how to create a datacenter with one host and run one
9 * cloudlet on it.
10 */
11
12 public class CloudSimExample1 {
13
14     /**
15      * The cloudlet list.
16     */
17     private static List<Cloudlet> cloudletList;
18
19     /**
20      * The vmlist.
21     */
22     private static List<Vm> vmlist;
23
24     /**
25      * Creates main() to run this example.
26      *
27      * @param args the args
28      */
29     @SuppressWarnings("unused")
30     public static void main(String[] args) {
31         Log.println("Starting CloudSimExample1...");
32
33         try {
34             // First step: Initialize the CloudSim package. It should be called
35             // before creating any entities.
36             int num_user = 1; // number of cloud users
37             Calendar calendar = Calendar.getInstance();
38             boolean trace_flag = false; // mean trace events
39
40             // Initialize the CloudSim library
41             CloudSim.init(num_user, calendar, trace_flag);
42
43             // Create a Datacenter
44             Datacenter dc = Datacenter.createDatacenter("Datacenter1");
45             dc.setHost(new Host("Host1", 1000));
46
47             // Create a Vm
48             Vm vm = Vm.createVm("Vm1");
49             vm.setHost(dc.getHostByName("Host1"));
50             vm.setMemory(1024);
51             vm.setCpu(1);
52             vm.setOsType("Ubuntu");
53
54             // Create a Cloudlet
55             Cloudlet cl = Cloudlet.createCloudlet("Cloudlet1");
56             cl.setVm(vm);
57             cl.setArrivalTime(0);
58             cl.setCloudletLength(100);
59             cl.setPriority(1);
60
61             // Add the cloudlet to the list
62             cloudletList.add(cl);
63
64             // Run the simulation
65             dc.run();
66
67         } catch (Exception e) {
68             e.printStackTrace();
69         }
70     }
71 }

```

```

Java - Kathy_CloudsimProjectCloudsim 1.0.2/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer [CloudSimExample1.java] CloudSimExample1.java [CloudSimExample1.java]
Kathy_CloudsimProject:
  org.cloudbus.cloudsim.examples
    CloudSimExample1.java
    CloudSimExample2.java
    CloudSimExample3.java
    CloudSimExample4.java
    CloudSimExample5.java
    CloudSimExample6.java
    CloudSimExample7.java
    CloudSimExample8.java
  org.cloudbus.cloudsim.examples.network
  org.cloudbus.cloudsim.examples.network.datacenter
  org.cloudbus.cloudsim.examples.power
  org.cloudbus.cloudsim.examples.power.planetLab
  org.cloudbus.cloudsim.examples.power.random
  workload.planetLab
  sources
  Referenced Libraries
JRE System Library [jre1.8.0_201]
Cloudsim 1.0.2
  docs
  jars
  build.xml
  changelog.txt
  examples.txt
  license.txt
  pom.xml
  readme.txt
  release_notes.txt
20 package org.cloudbus.cloudsim.examples;
21
22 import java.util.List<Cloudlet> cloudletList;
23
24 /**
25  * A simple example showing how to create
26  * a datacenter with one host and two VMs
27  * running on it. The cloudlets run in
28  * VMs with the same HIVE requirements.
29  * The Cloudlets will take the same time to
30  * complete the execution.
31 */
32 public class CloudSimExample1 {
33
34     /**
35      * The cloudlet list.
36      */
37     private static List<Cloudlet> cloudletList;
38
39     /**
40      * The VM list.
41      */
42     private static List<Vm> vmList;
43
44     /**
45      * Create main() to run this sample.
46      */
47     public static void main(String[] args) {
48         Log.println("Starting CloudSimExample1...*");
49     }
50 }

```

**Step 17:** Now navigate to the Eclipse menu \_Run ->Run\_ or directly use a keyboard shortcut ‘Ctrl + F11’ to execute the \_CloudSimExample1.java\_.

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

File Edit Source Refactor Navigate Search Project **Run** Window Help

Run Ctrl+F11

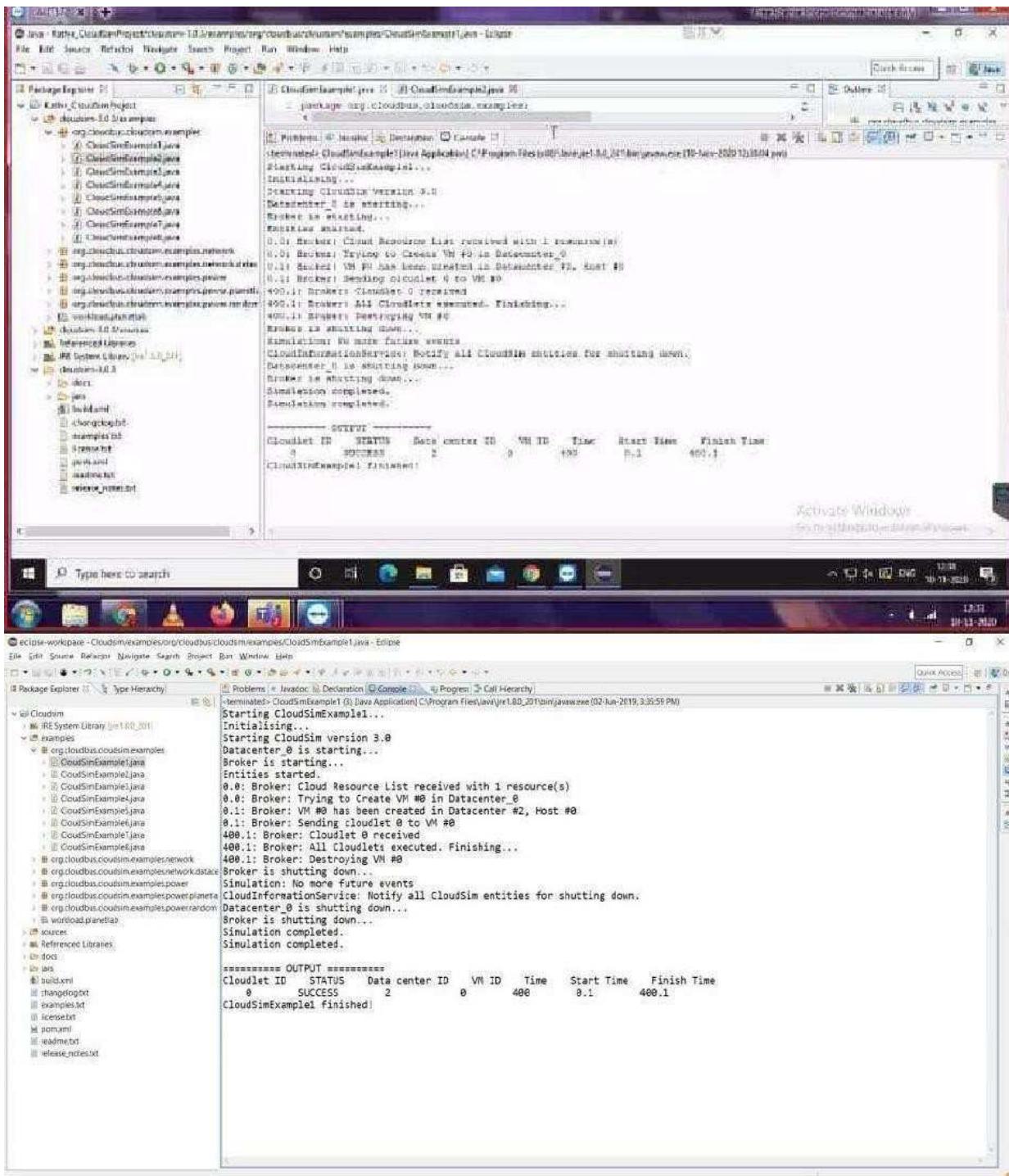
- Run F11
- Debug
- Coverage Ctrl+Shift+F11
- Run History
- Run As
- Run Configurations...
- Debug History
- Debug As
- Debug Configurations...
- Coverage History
- Coverage As
- Coverage Configurations...
- Breakpoint Types
- Toggle Breakpoint Ctrl+Shift+B
- Toggle Tracepoint
- Toggle Line Breakpoint
- Toggle Watchpoint
- Toggle Method Breakpoint
- Skip All Breakpoints Ctrl+Alt+B
- Remove All Breakpoints
- Add Java Exception Breakpoint...
- Add Class Load Breakpoint...
- All References...
- All Instances... Ctrl+Shift+N
- Instance Count...
- Watch
- Inspect
- Display
- Execute
- Force Return
- External Tools

```

Cloudsim.examples > CloudSim
sim.examples;
Toolkit
rmatis;
g how to create a da
le1 {
/*
oudlet> cloudletList;
> vmList;
un this example.
S.
sed")
n(String[] args) {
rting CloudSimExample1
60 // First step: Initialize the Cloud
61 // before creating any entities.
62 int num user = 1: // number of clo

```

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



The screenshot shows two separate Eclipse IDE windows, both titled "Java - KaliCloudProjectCloudsim 1.0 Examples/org.cloudsim/examples/CloudSimExample1 - Java Application".

**Top Window Output:**

```
<terminated> CloudSimExample1 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe [02-Jun-2019, 3:35:59 PM]
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Broker is started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter_0, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.0: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

=====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
CloudSimExample1 finished!
```

**Bottom Window Output:**

```
<terminated> CloudSimExample1 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe [02-Jun-2019, 3:35:59 PM]
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Broker is started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.0: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

=====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
CloudSimExample1 finished!
```

## Result:

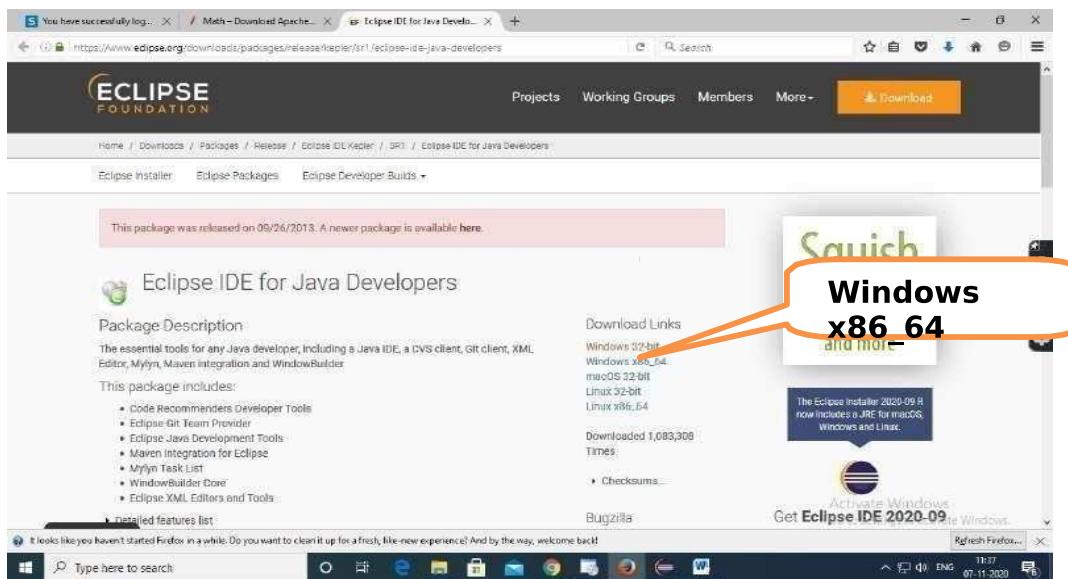
Thus the cloudsim is simulated using Eclipse Environment successfully.

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

### **Procedure to import Eclipse, running scheduling algorithms in your system**

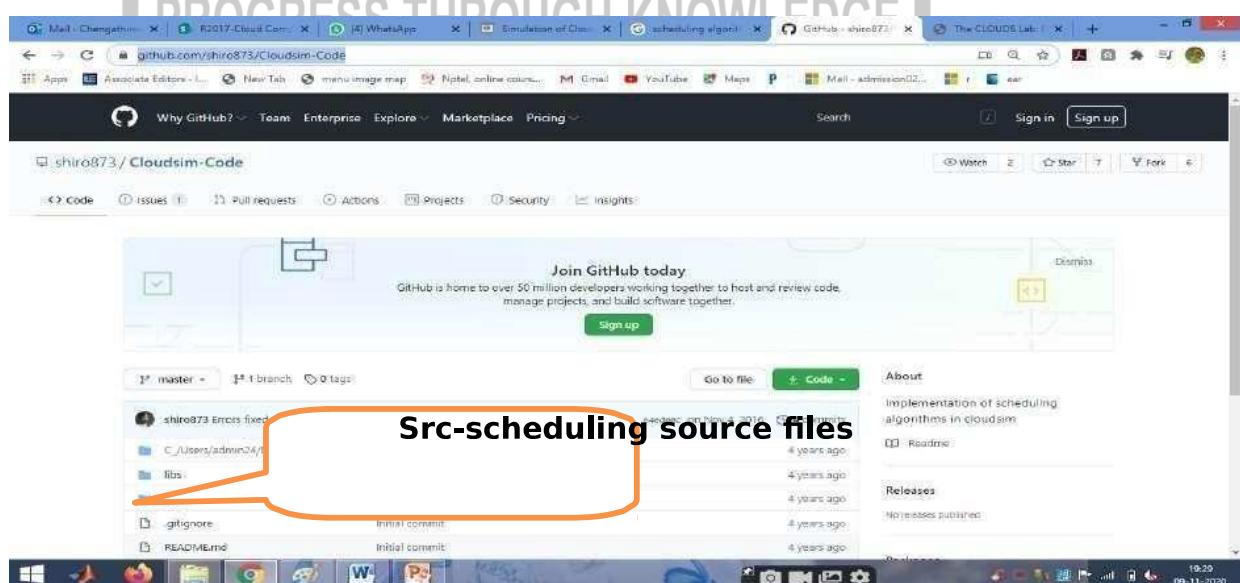
**Step 1:** Link to download Eclipse and download Eclipse for Windows 64bit into your Local machine

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



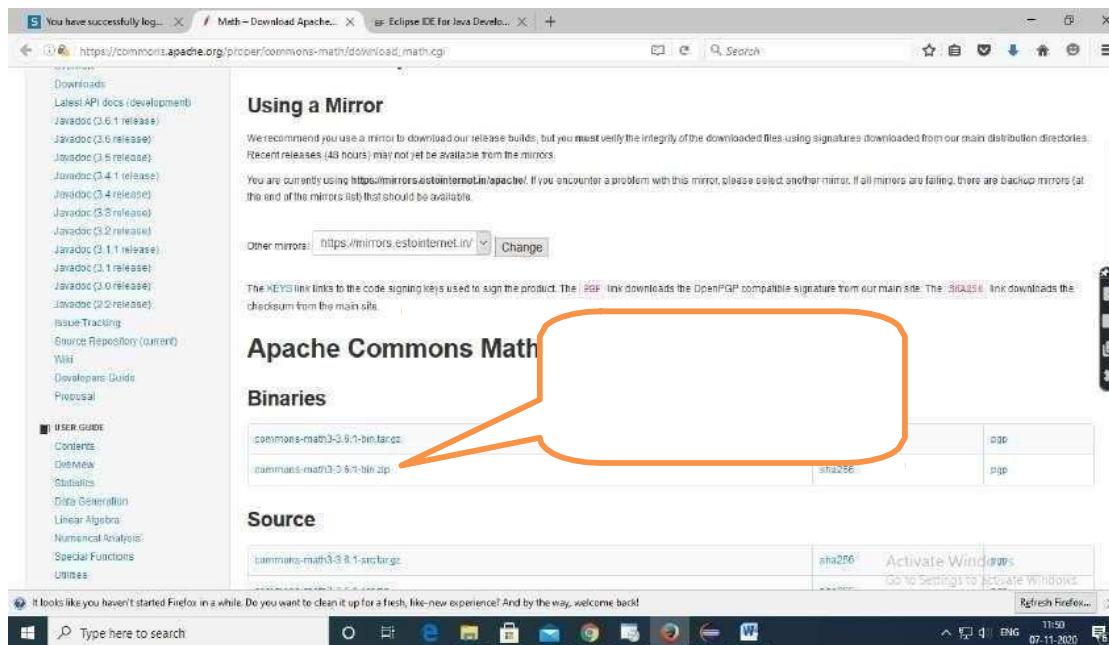
**Step 2:** Download scheduling source code **cloudsim-code-master** from git hub repository in your local machine

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

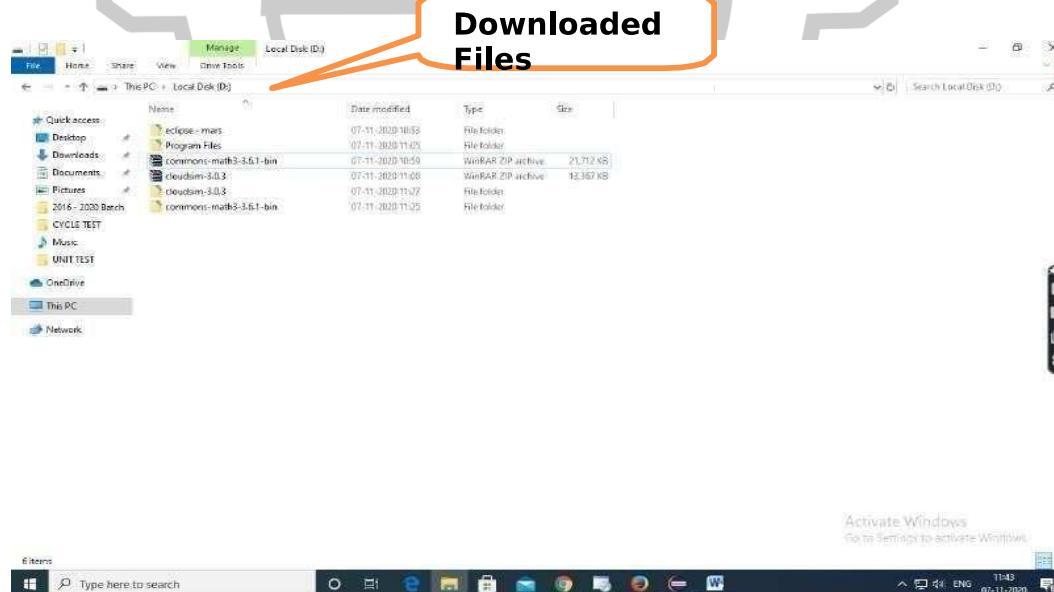


**Step 3:** Download commons-math3-3.6.1 from git hub repository in your local machine

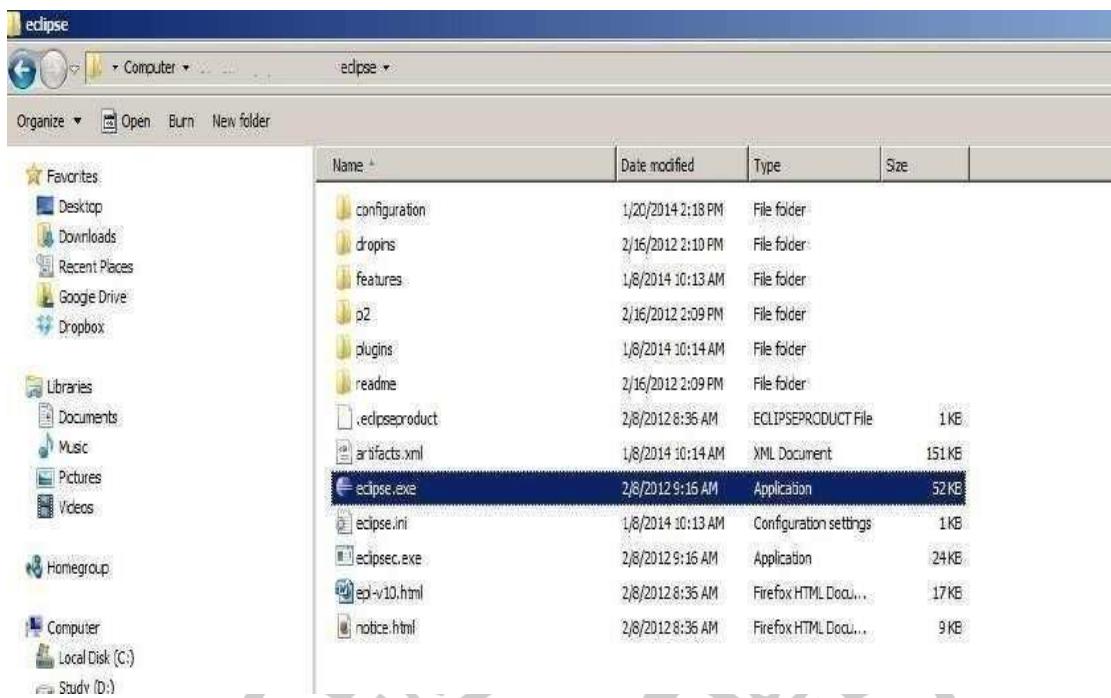
[https://commons.apache.org/proper/commons-math/download\\_math.cgi](https://commons.apache.org/proper/commons-math/download_math.cgi)



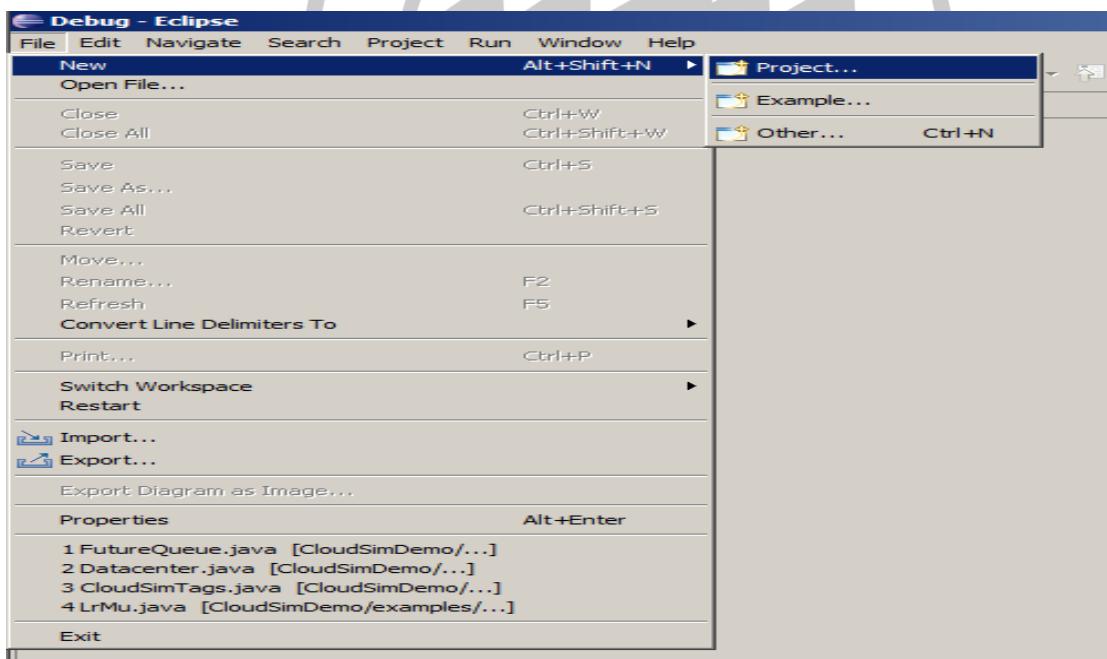
**Step 4:** Downloaded Eclipse, cloudsim-3.0.3 and Apache Commons Math 3.6.1 in your local machine and extract cloudsim-3.0.3 and Apache Commons Math 3.6.1



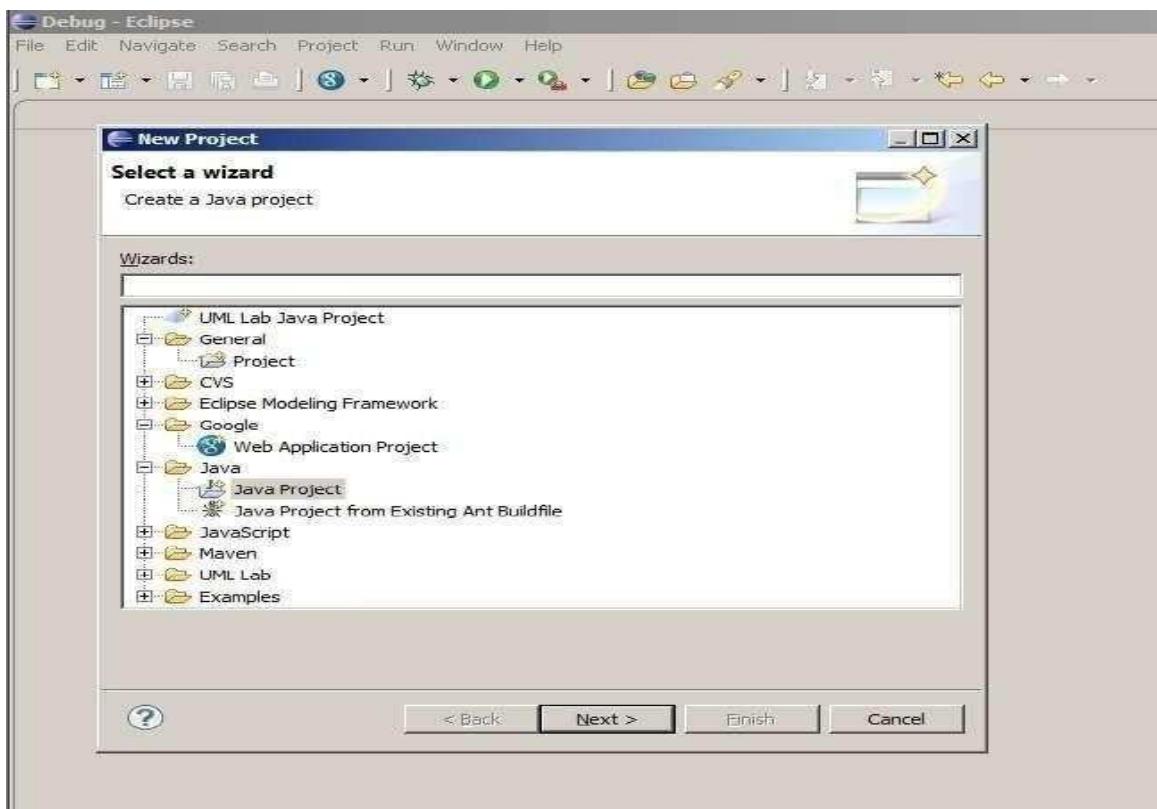
**Step 5:** First of all, navigate to the folder where you have unzipped the eclipse folder and open Eclipse.exe



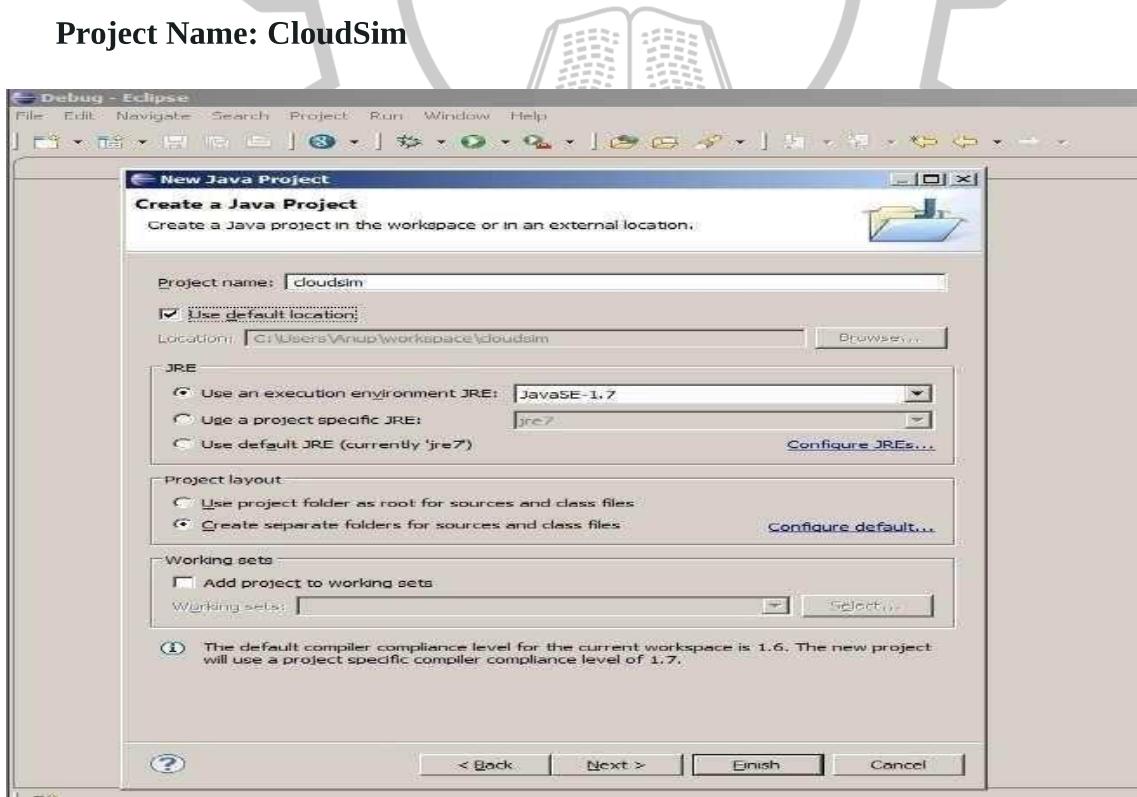
**Step 6:** Now within Eclipse window navigate the menu: *File -> New -> Project*, to open the new project wizard



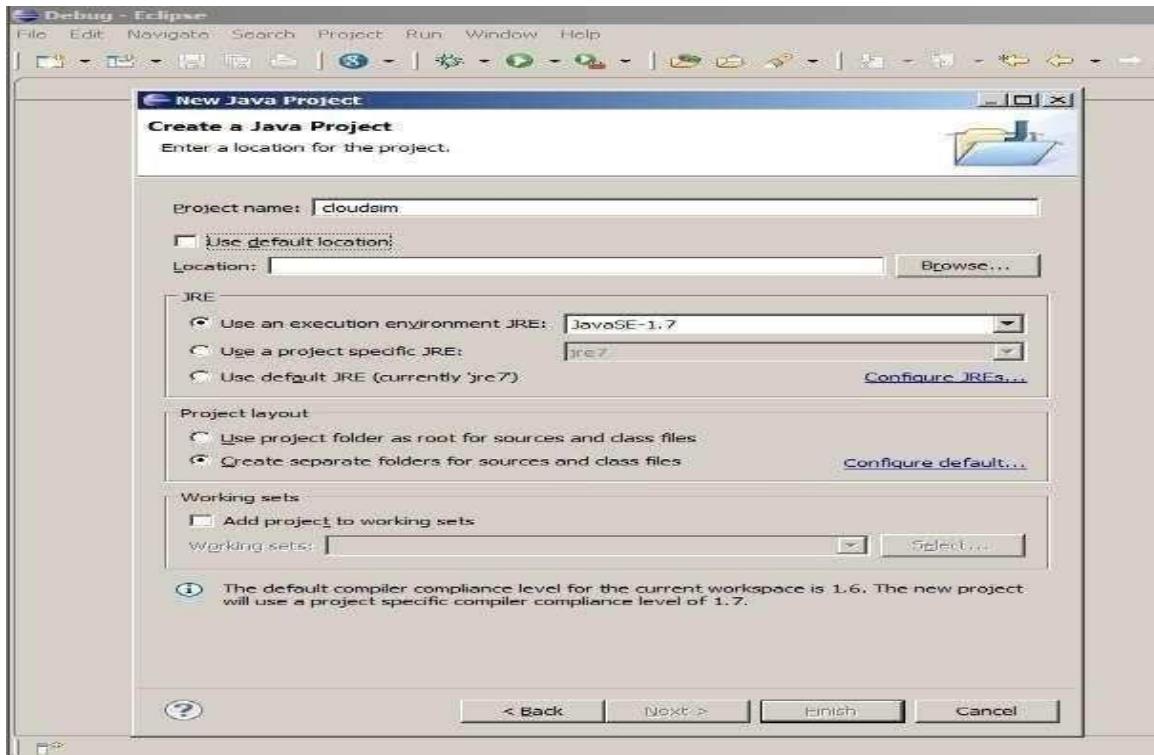
**Step 7:** A New Project wizard should open. There are a number of options displayed and you have to find & select the Java Project option, once done click ‘Next’



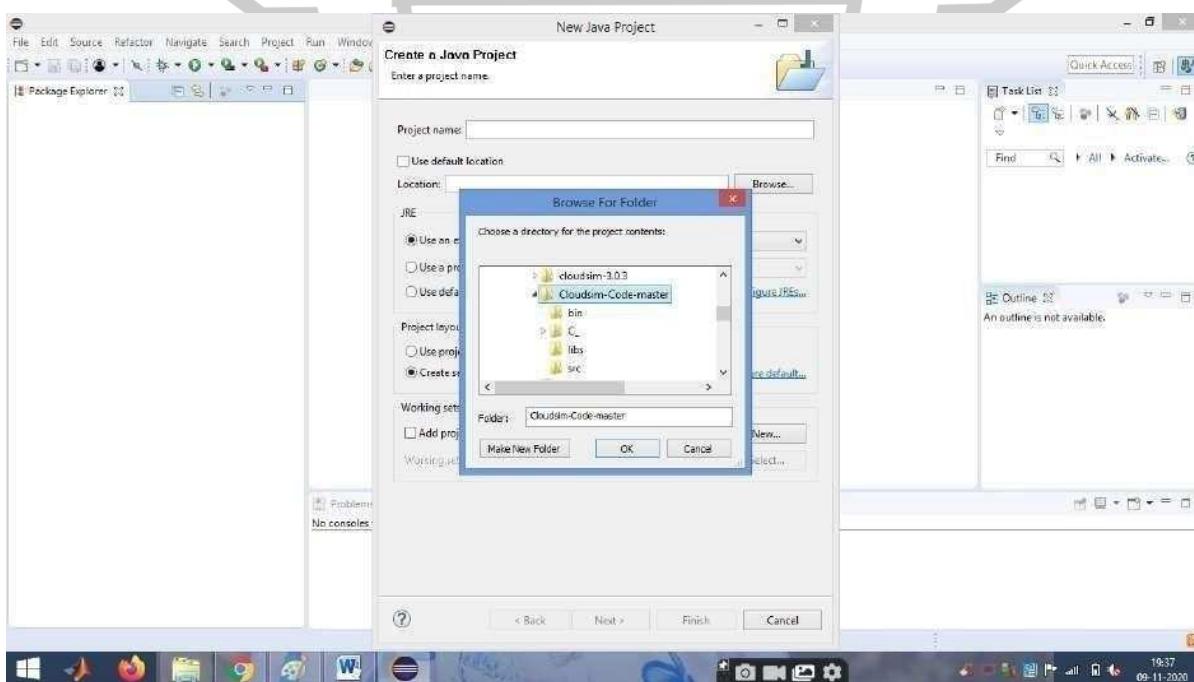
**Step 8:** Now a detailed new project window will open, here you will provide the project name and the path of CloudSim-master-code project source code, which will be done as follows:



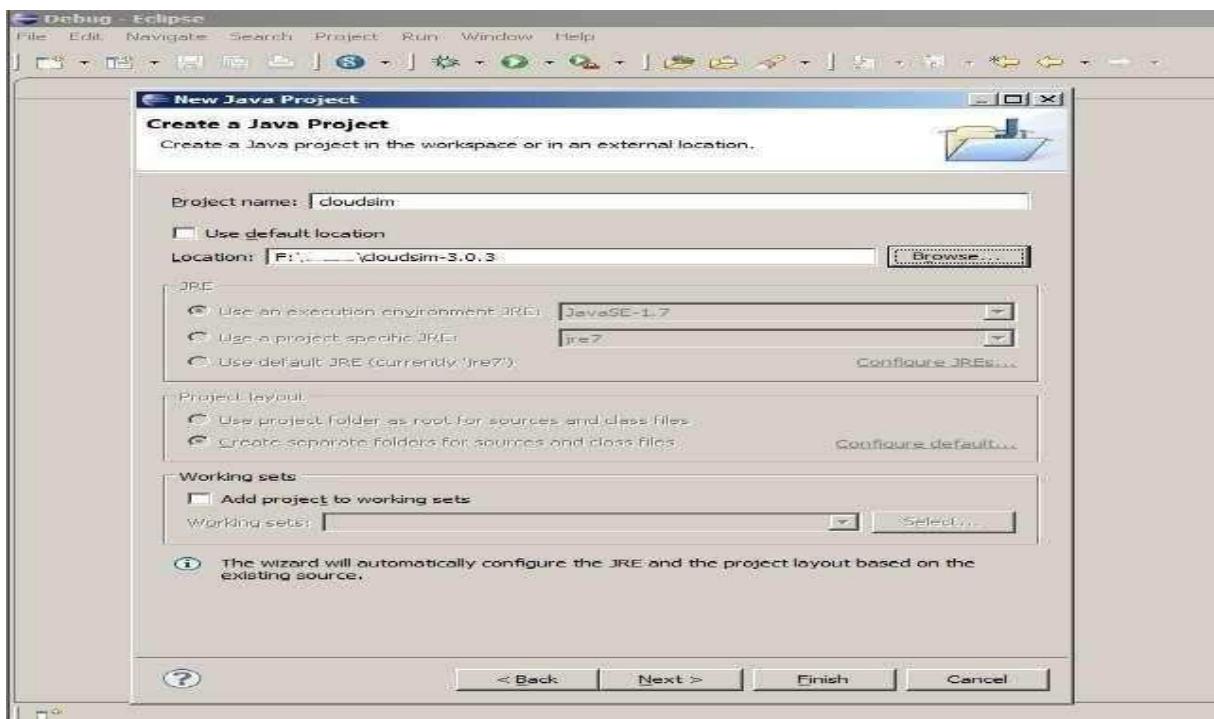
**Step 9:** Unselect the ‘Use default location’ option and then click on ‘Browse’ to open the path where you have unzipped the Cloudsim-code-master project and finally click Next to set project settings.



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

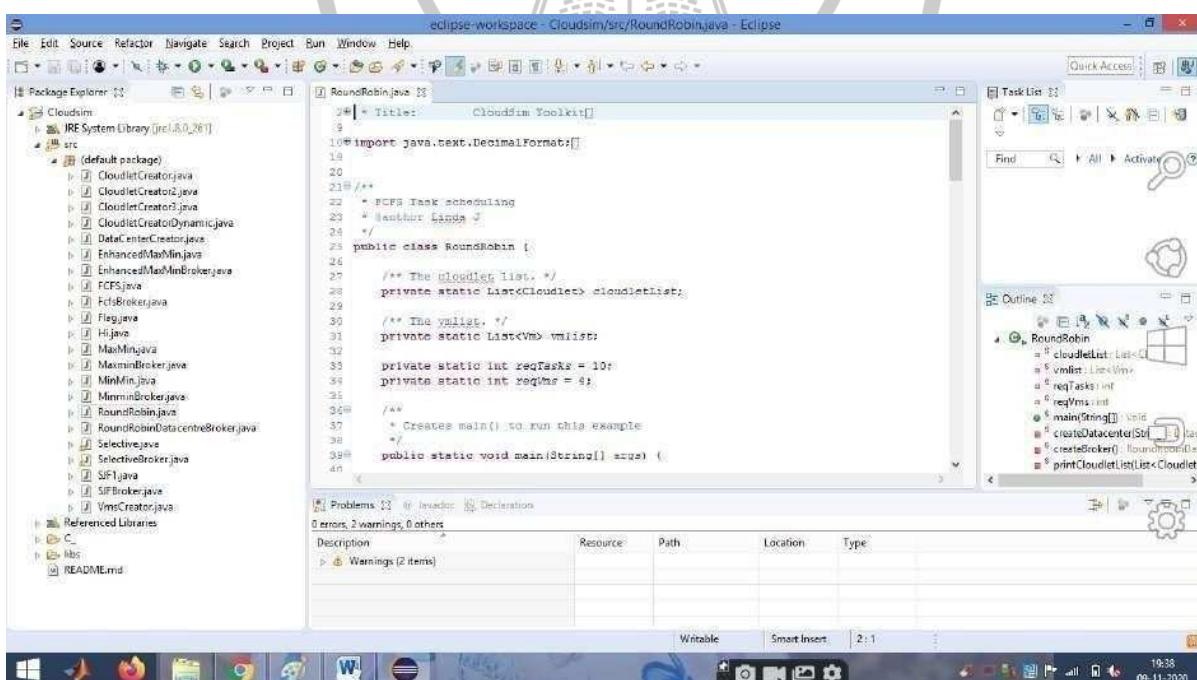


**Step 11:** Once done finally, click ‘Next’ to go to the next step i.e. setting up of project settings

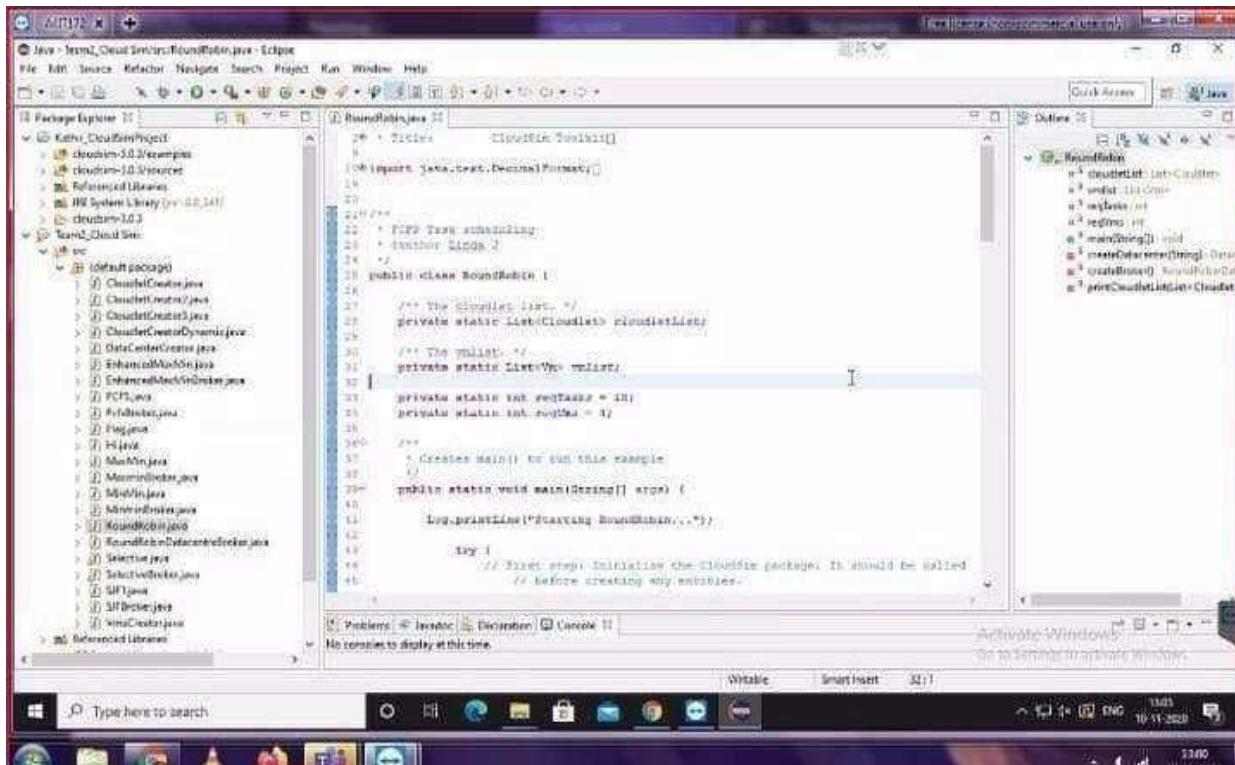


**Step 12:** Once the project is configured you can open the ‘Project Explorer’ and start exploring the Cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured Cloudsim project, which may take some time depending on the configuration of the computer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

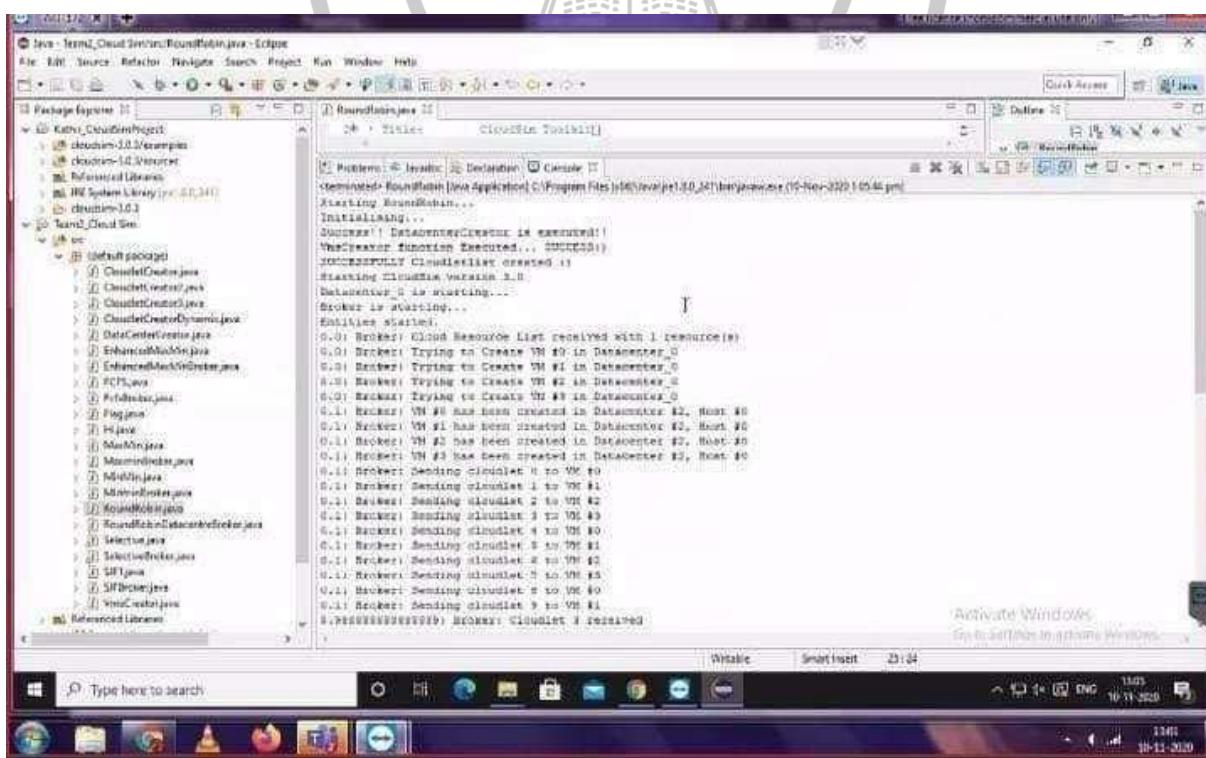
Following is the final screen which you will see after Cloudsim is configured.



**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 RoundRobin.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 Eclipse Environment successfully.

## **Ex No.6 Procedure File Transfer in Client & Server using virtual machine.**

### **Aim:**

To procedure File Transfer in Client & Server using virtual machine

### **Steps:**

Steps to perform File Transfer in Client & Server using virtual machine.

Step 1: Open a virtual machine to do file transfer.

Step 2: Write the java program for FTP Client and FTP Server.

Step 3: Run the program.

### **Source Code:**

#### **FTPClient.java**

```
import java.io.*;
import java.net.*;
import java.util.*;
public class FTPClient{
    public static void main(String args[])throws IOException {
        try {
            int number;
            Socket s=new Socket("127.0.0.1",10087);
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the file
name:"); String fn=sc.next();
            DataOutputStream dos=new
DataOutputStream(s.getOutputStream()); dos.writeUTF(fn);
            DataInputStream dis=new DataInputStream(s.getInputStream());
            String input=(String)dis.readUTF();
            FileInputStream fis=new FileInputStream(input);
            System.out.println("Even Numbers in the" +fn+" are");
            int i=0;
            while((i=fis.read())!=-1){
                System.out.println((char)i);
            }
            s.close();
        }
        catch(Exception e){
            System.out.println("Port not available "+e);
        }
    }
}
```

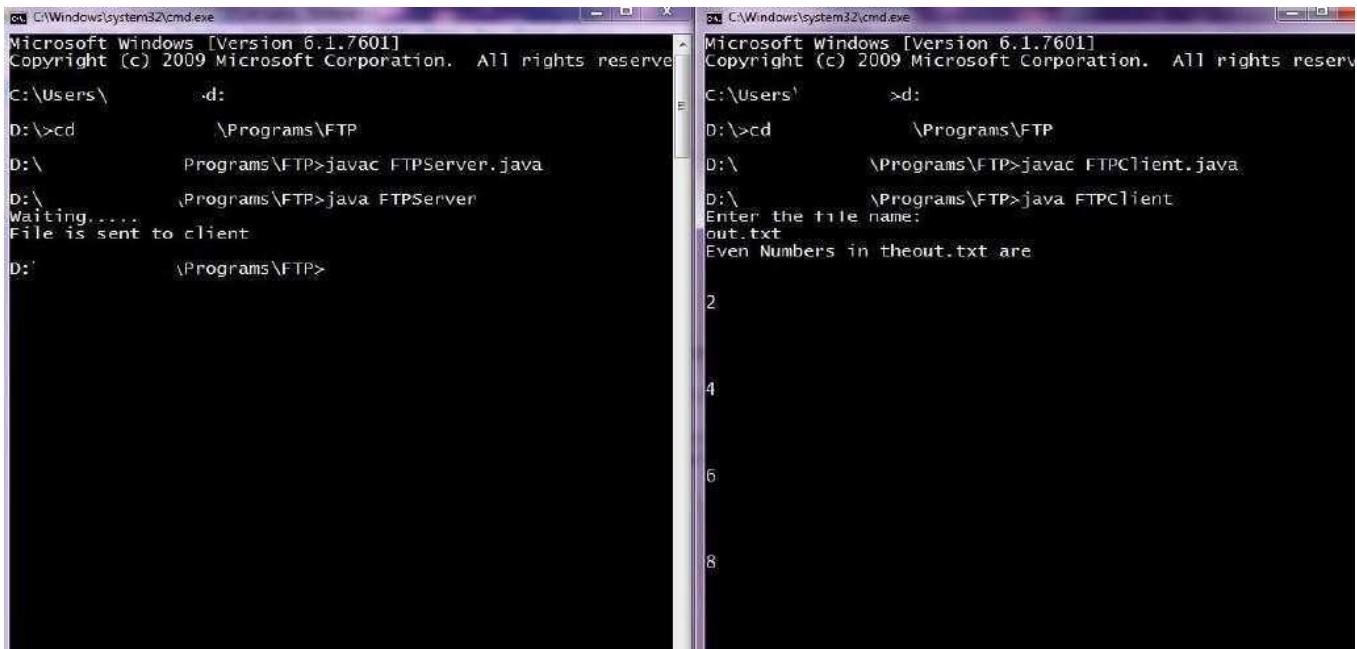
### **FTPServer.java**

```
import java.io.*;
import java.net.*;
import java.util.*;
public class FTPServer{
    public static void main(String args[])throws IOException{
        try{
            int num;
            Scanner sc=new Scanner(System.in);
            ServerSocket ss=new ServerSocket(10087);
            Socket s=ss.accept();
            System.out.println("Waiting.....");
            DataInputStream dis=new DataInputStream(s.getInputStream());
            String input=(String)dis.readUTF();
            DataOutputStream dos=new
            DataOutputStream(s.getOutputStream()); FileInputStream fis = new
            FileInputStream("out.txt"); FileOutputStream fos = new
            FileOutputStream(input); while((num=fis.read())!= -1) {
                if(num%2==0) {
                    fos.write(num);
                }
            }
            dos.writeUTF(input);
            System.out.println("File is sent to
client"); ss.close();
            s.close();
        }
        catch(Exception e) {
            System.out.println("Port not available"+e);
        }
    }
}
```

### **Out.txt**

1  
2  
3  
4  
5  
6  
7  
8  
9

## **Output:**



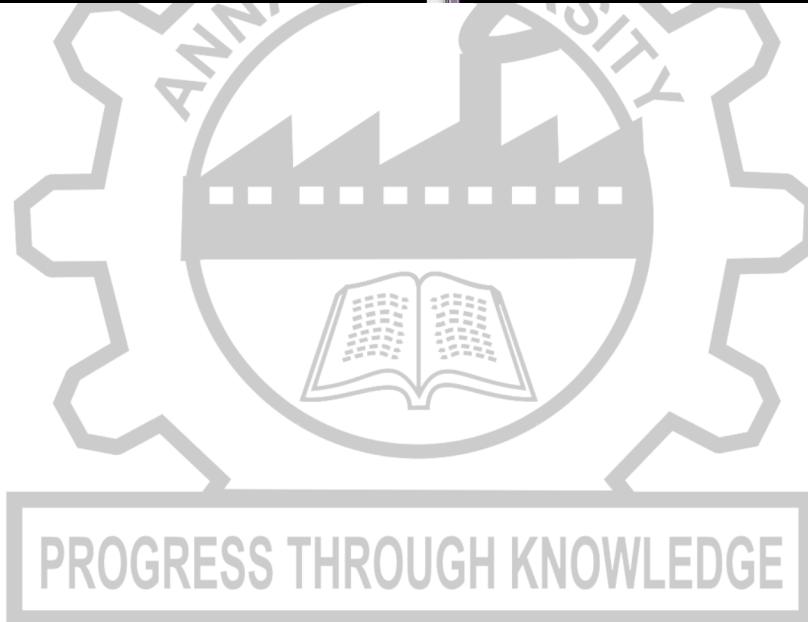
The image shows two side-by-side Microsoft Windows command-line windows. Both windows have the title bar "Microsoft Windows [Version 6.1.7601]" and the status bar "C:\Windows\system32\cmd.exe".

The left window (Server) shows the following command sequence:

```
C:\Users> cd \Programs\FTP  
D:\Programs\FTP> javac FTPServer.java  
D:\Programs\FTP> java FTPServer  
Waiting.....  
File is sent to client  
D:\Programs\FTP>
```

The right window (Client) shows the following command sequence:

```
C:\Users> cd \Programs\FTP  
D:\Programs\FTP> javac FTPClient.java  
D:\Programs\FTP> java FTPClient  
Enter the file name:  
out.txt  
Even Numbers in the out.txt are  
2  
4  
6  
8
```



## **Result:**

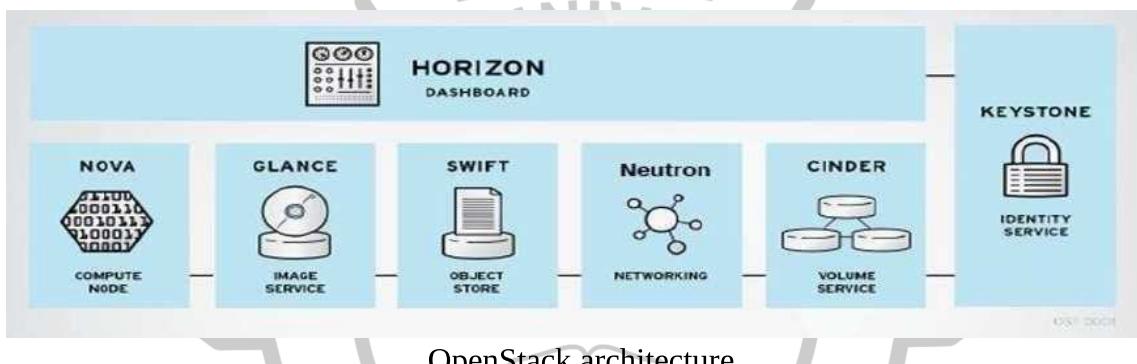
Thus the program to the File transfer operation using virtual machine was successfully executed and verified.

## **Ex No. 7 | Find a procedure to launch virtual machine using Openstack**

### **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 and control 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



- 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**

**PROGRESS THROUGH KNOWLEDGE**

- **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 integration 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 Ubuntu 7 OVA(Open Virtual Appliance) from
  - Link : <https://linuxvmimages.com/images/Ubuntu-7>
- Import Ubuntu 7 OVA(Open Virtual Appliance) into Oracle Virtual Box



### Step 3: Login into CentOS 7

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

**#sudosu-**

```

File Edit View Search Terminal Help
[edureka@localhost ~]$ su
Password:
[root@localhost edureka]# [REDACTED]

```

### Step 4: Installation Steps for OpenStack

**Step5:** Command to disable and stop firewall

```

# systemctl disable
firewalld #systemctl stop
firewalld

```

```
[root@localhost ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
Removed symlink /etc/systemd/system/basic.target.wants/firewalld.service.
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]#
```

I

#### Step 6: Command to disable and stop Network Manager

**# systemctl disable NetworkManager**

**# systemctl stop NetworkManager**

```
[root@localhost ~]# systemctl disable NetworkManager
Removed symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
Removed symlink /etc/systemd/system/dbus-org.freedesktop.NetworkManager.service.
Removed symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
[root@localhost ~]# systemctl stop NetworkManager
[root@localhost ~]#
```

I

#### Step 7: Enable and start Network

**#systemctl enable network**

**#systemctl start network**

```
[root@localhost ~]# systemctl enable network
network.service is not a native service, redirecting to /sbin/chkconfig.
Executing /sbin/chkconfig network on
[root@localhost ~]# systemctl start network
[root@localhost ~]#
```

**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 Ubuntu 7 run the below command.

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

```
[root@localhost ~]# yum install -y centos-release-openstack-newton
```

**Step 9:** Update Current packages

```
#yum update -y
```

```
[root@localhost ~]# yum update -y
Loaded plugins: fastestmirror, langpacks
centos-ceph-jewel
centos-openstack-newton
centos-qemu-ev
(1/3): centos-ceph-jewel/7/x86_64/primary_db      2.9 kB  00:00:00
(2/3): centos-qemu-ev/7/x86_64/primary_db        2.9 kB  00:00:00
(3/3): centos-openstack-newton/x86_64/primary_db   2.9 kB  00:00:00
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net                   63 kB  00:00:01
 * extras: centos.excellmedia.net                 52 kB  00:00:00
 * updates: mirrors.viethosting.com               853 kB  00:00:02
```

## Step 10: Install OpenStack Release for Ubuntu

```
#yum install -y openstack-packstack
```

```
[root@localhost ~]# yum install -y openstack-packstack
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * extras: centos.excellmedia.net
 * updates: mirrors.viethosting.com
[...]
```

## Step 11: Start packstack to install OpenStack Newton

```
#packstack --allinone
```

```
[root@localhost ~]# packstack --allinone
Welcome to the Packstack setup utility

The installation log file is available at: /var/tmp/packstack/20170314-065810-b8cxch/openstack-setup.log
Packstack changed given value to required value /root/.ssh/id_rsa.pub

Installing:
Clean Up                                [ DONE ]
Discovering ip protocol version          [ DONE ]
Setting up ssh keys                      [ DONE ]
Preparing servers                        [ DONE ]
Pre installing Puppet and discovering hosts' details [ DONE ]
Preparing pre-install entries           [ DONE ]
Setting up CACERT                         [ DONE ]
Preparing AMQP entries                   [ DONE ]
Preparing MariaDB entries                [ DONE ]
Fixing Keystone LDAP config parameters to be undef if empty [ DONE ]
Preparing Keystone entries               [ DONE ]
Preparing Glance entries                 [ DONE ]
Checking if the Cinder server has a cinder-volumes vg [ DONE ]
Preparing Cinder entries                 [ DONE ]
Preparing Nova API entries              [ DONE ]
[...]
```

SUBSCRIBE

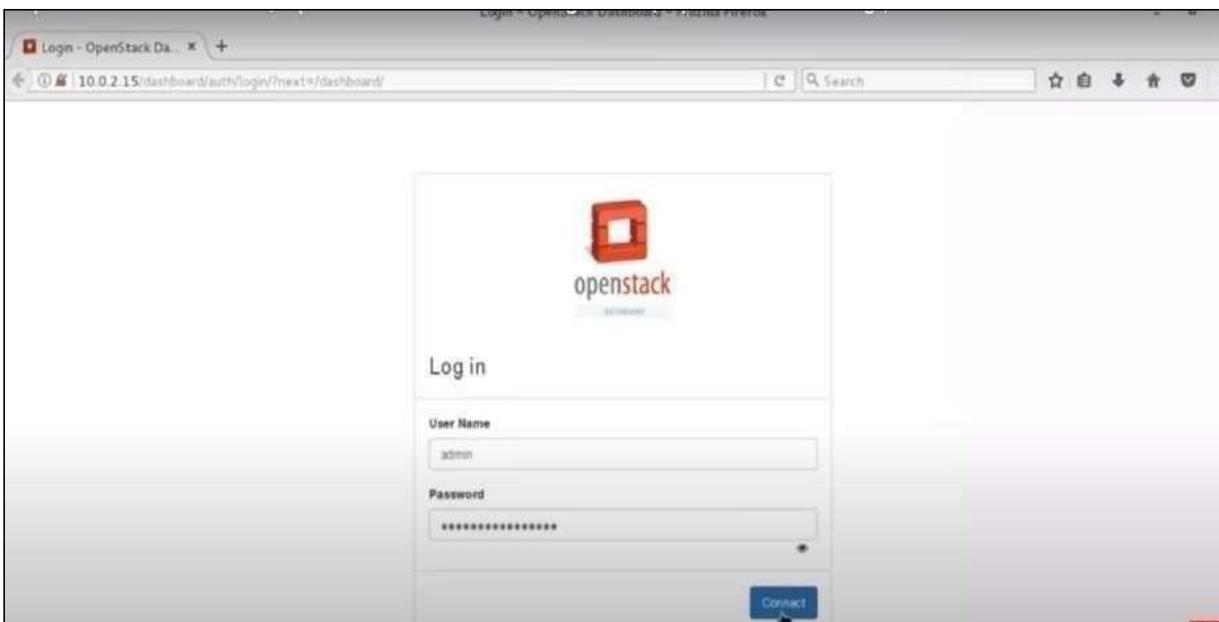
## Step 12: Note the user name and password from keystonec\_admin

```
#cat keystonec_admin
```

```
[root@localhost ~]# ls
anaconda-ks.cfg      keystonec_admin  packstack-answers-20170314-065812.txt
initial-setup-ks.cfg  keystonec_demo
[root@localhost ~]# cat keystonec_admin
unset OS_SERVICE_TOKEN
export OS_USERNAME=admin
export OS_PASSWORD=cdc897f8cb7f4dda
export OS_AUTH_URL=http://10.0.2.15:5000/v2.0
export PS1='[\u@\h \w(keystone_admin)]\$ '

export OS_TENANT_NAME=admin
export OS_REGION_NAME=RegionOne
[root@localhost ~]#
```

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



**OpenStack is successfully launched in your machine**

A screenshot of the OpenStack dashboard projects page. The URL in the address bar is 10.0.2.15/dashboard/identity/. The left sidebar shows navigation links for Project, Admin, and Identity. The main content area is titled "Projects" and displays a table of existing projects. The table has columns for Name, Description, Project ID, Domain Name, Enabled, and Actions. Three projects are listed: "services" (Description: Tenant for the openstack services), "admin" (Description: admin tenant), and "demo" (Description: default tenant). The "Actions" column for each row contains a "Manage Members" button.

Projects - OpenStack... +

10.0.2.15/dashboard/identity/

openstack admin

Resource Usage

	Name	Description	Project ID	Domain Name	Enabled	Actions
Hypervisors	services	Tenant for the openstack services	2a6451398c8340bba1254a079b74a483	Default	Yes	Manage Members
Host Aggregates	admin	admin tenant	58095a14406547d88982e8f69d82bc94	Default	Yes	Manage Members
Instances	demo	default tenant	a5d60a950a484c98977d5d5c171562d	Default	Yes	Manage Members
Volumes						
Flavors		Displaying 3 items				
Images						
Networks						
Routers						
Floating IPs						
Defaults						
Metadata Definitions						
System Information						

10.0.2.15/dashboard/admin/networks/

SUBSCRIBE



## Result:

Thus the OpenStack Installation is executed successfully.

## **Ex. No. 8a      Install Hadoop single node cluster**

### **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:~$ su
```

Password:

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

#### **Step 3:**

**PROGRESS THROUGH KNOWLEDGE**

**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.

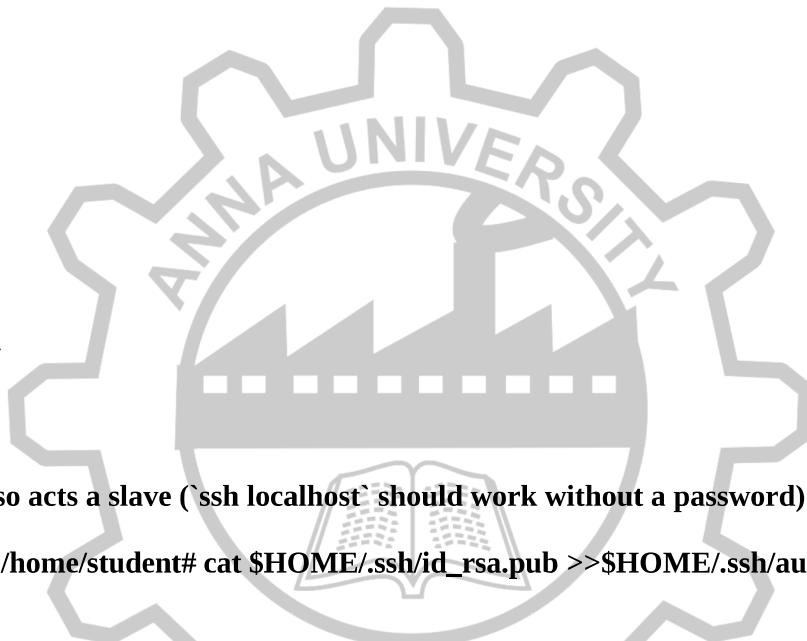
The key fingerprint is:

```
77:a1:20:bb:db:95:6d:89:ce:44:25:32:b6:81:5d:d5 root@a4cse196
```

The 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_keys
```

**Step 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 default

Full Name []:

Room Number []:

Work Phone []:

Home Phone []:

Other []:

Is the information correct? [Y/n]

y root@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# **sudo 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 ~/.bashrc
```

**Add to ~/.bashrc file:**

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

```
export 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>**

**Result:**

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

## **Ex. No. 8 b Word Count Program Using Map And Reduce**

### **Aim:**

tasks. To Count the number of words using JAVA for demonstrating the use of Map and Reduce

### **Procedure:**

1. Analyze the input file content
2. Develop the code
  - a. Writing a map function
  - b. Writing a reduce function
  - c. Writing the Driver class
3. Compiling the source
4. Building the JAR file
5. Starting the DFS
6. Creating Input path in HDFS and moving the data into Input path
7. Executing the program

### **Program: WordCount.java**

```
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

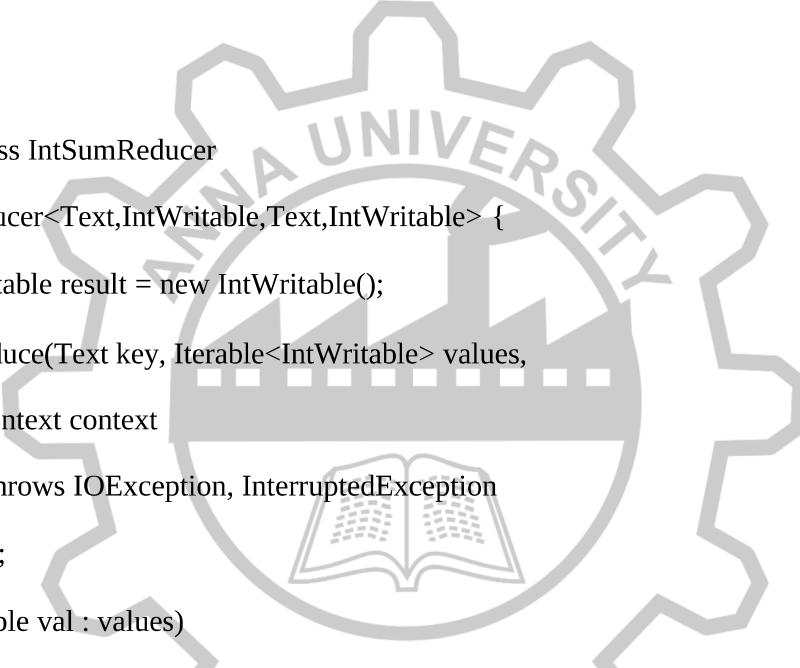
    { int sum = 0;
        for (IntWritable val : values)

            { 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");
```

The watermark features the Anna University logo, which is a circular emblem. The outer ring contains the text "ANNA UNIVERSITY". Inside the circle is an open book with visible text lines. Below the circle is a rectangular banner with the text "PROGRESS THROUGH KNOWLEDGE".

PROGRESS THROUGH KNOWLEDGE

```
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);

}

}

Save the program as
```

**WordCount.java**

**Step 1:** Compile the java program

For compilation we need this hadoop-core-1.2.1.jar file to compile the mapreduce program.

<https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core/1.2.1>

Assuming both jar and java files in same directory run the following command to compile

**root@a4cseh160:/#javac -classpath hadoop-core-1.2.1.jar WordCount.java**

**Step 2:** Create a jar file

**Syntax:**

**PROGRESS THROUGH KNOWLEDGE**

jar cf jarfilename.jar MainClassName\*.class

**Output:**

**root@a4cseh160:/#jar cf wc.jar WordCount\*.class**

**Step 3:** Make directory in hadoop file system

**Syntax:**

hdfs dfs -mkdir directoryname

**Output:**

**root@a4cseh160:/# hdfs dfs -mkdir /user**

**Step 4:** Copy the input file into hdfs

**Syntax:**

```
hdfs dfs -put sourcefile destpath
```

**Output:**

```
root@a4cseh160:/#hdfs dfs -put /input.txt /user
```

**Step 5: To run a program**

**Syntax:**

```
hadoop jar jarfilename main_class_name inputfile outputpath
```

**Output:**

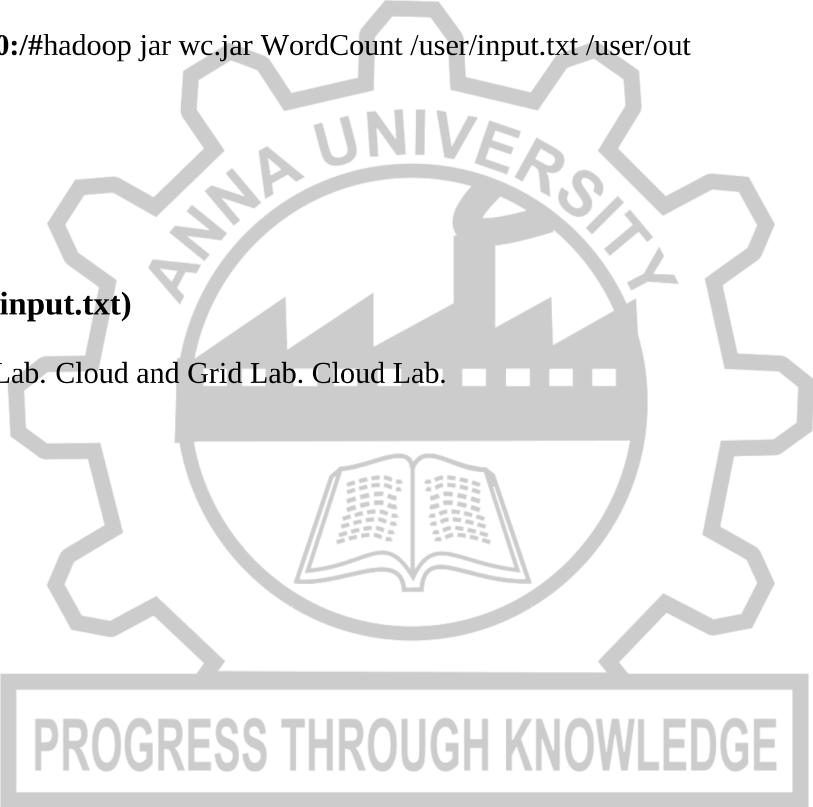
```
root@a4cseh160:/#hadoop jar wc.jar WordCount /user/input.txt /user/out
```

**Input File: (input.txt)**

Cloud and Grid Lab. Cloud and Grid Lab. Cloud Lab.

**Output:**

18	
3	Cloud
3	Lab.
2	Grid
2	and



ANNA UNIVERSITY

PROGRESS THROUGH KNOWLEDGE

**Step 6:** Check the output in the Web UI at <http://localhost:50070>.

In the Utilities tab select browse file system and select the correct user.

The output is available inside the output folder named **user**.

Browsing HDFS - Mozilla Firefox  
KCET Captive Portal | mapreduce-msamk... | Browsing HDFS | +  
localhost:50070/explorer.html#/

Hadoop Overview Datanodes Snapshot Startup Progress Utilities →

Browse the file system  
Logs

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	24/9/2016, 3:43:28 PM	0	0 B	hadoop1
drwxr-xr-x	root	supergroup	0 B	24/9/2016, 3:41:52 PM	0	0 B	hadoopuser
drwx-----	root	supergroup	0 B	8/9/2016, 11:22:41 AM	0	0 B	tmp
drwxr-xr-x	root	supergroup	0 B	24/9/2016, 3:46:23 PM	0	0 B	user

Hadoop, 2014.

localhost:50070/explorer.html#

**Step 7:** To Delete an output folder

**Syntax:**

hdfs dfs -rm -R outputpath

**Output:**

```
root@a4cseh160:/#hdfs dfs -rm -R /user/out.txt
```

**Result:**

Thus the numbers of words were counted successfully by the use of Map and Reduce tasks.