SRM Valliammai Engineering College

Department of Computer Science & Engineering

CS8711 CLOUD COMPUTING LABORATORY

VII SEMESTER CSE I & II

Academic Year 2021-22

Prepared by

Ms.K.Devi, AP/CSE

Ms.S.Suma,AP/CSE

CS8711 CLOUD COMPUTING LABORATORY L T P C 0 0 3 2

OBJECTIVES:

The student should be made to:

- Be exposed to tool kits for grid and cloud environment.
- Be familiar with developing web services/Applications in grid framework
- Learn to run virtual machines of different configuration.
- Learn to use Hadoop

LIST OF EXPERIMENTS:

Course Objective:

- To develop web applications in cloud
- To learn the design and development process involved in creating a cloud based application
- To learn to implement and use parallel programming using Hadoop

Exercises:

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.

Course Outcome:

On completion of this course, the students will be able to:

- Configure various virtualization tools such as Virtual Box, VMware workstation.
- Design and deploy a web application in a PaaS environment.
- Learn how to simulate a cloud environment to implement new schedulers.
- Install and use a generic cloud environment that can be used as a private cloud.
- Manipulate large data sets in a parallel environment.

TOTAL: 45 PERIODS

CS8711 CLOUD COMPUTING LABORATORY

EX. No:1 Install VirtualBox/VMware Workstation

Aim:

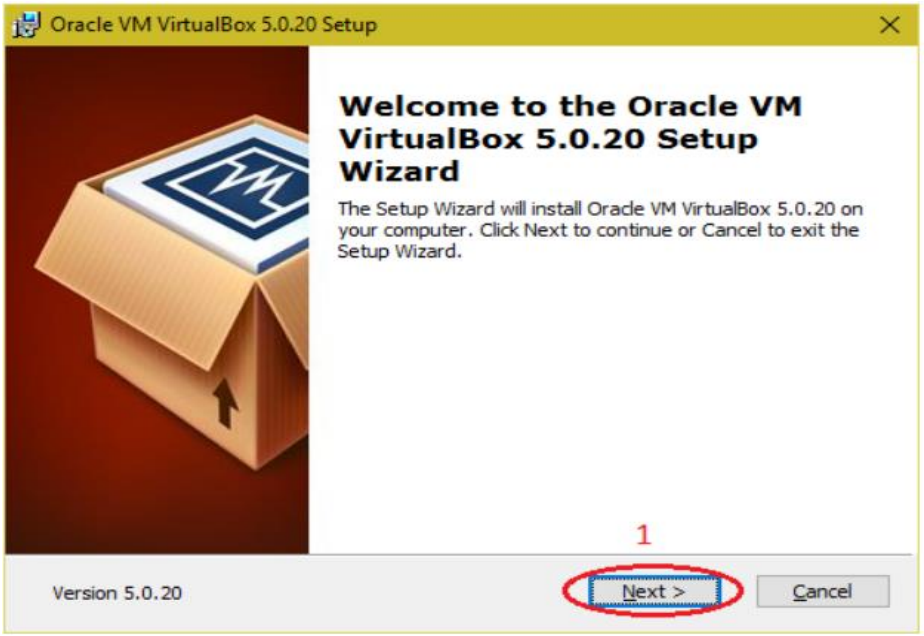
Find procedure to Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows7 or 8.

Procedure:

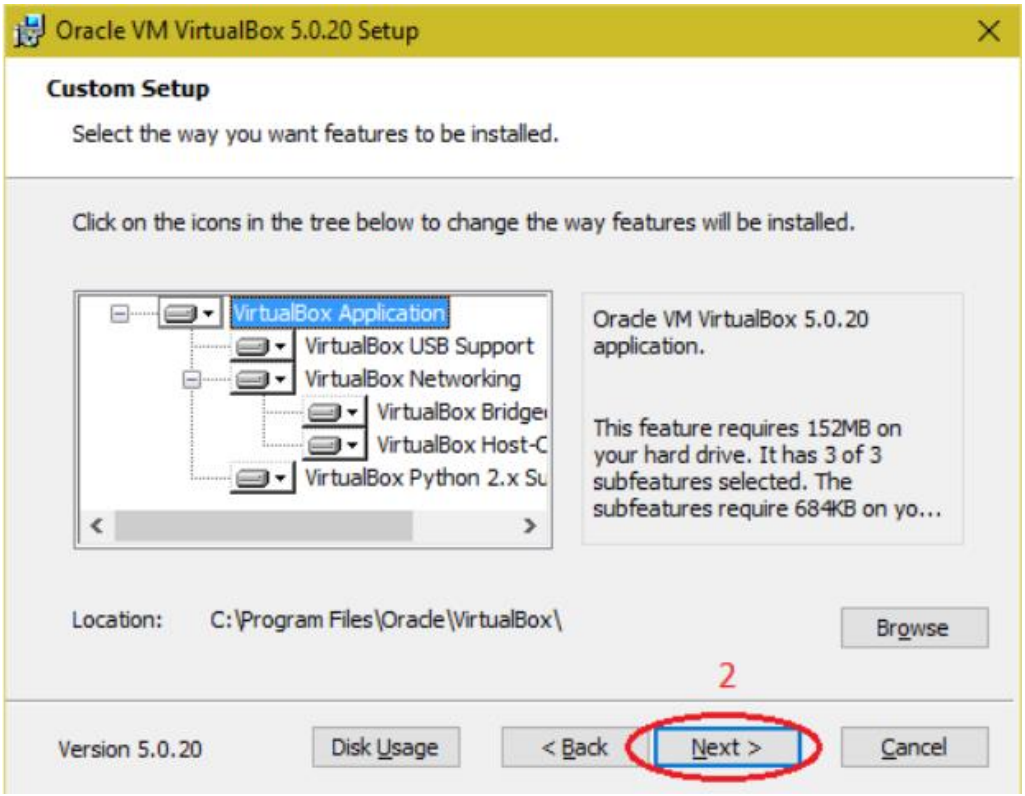
Step 1- Download Link

To download VirtualBox setup for installation click the link https://www.virtualbox.org/wiki.Downloads

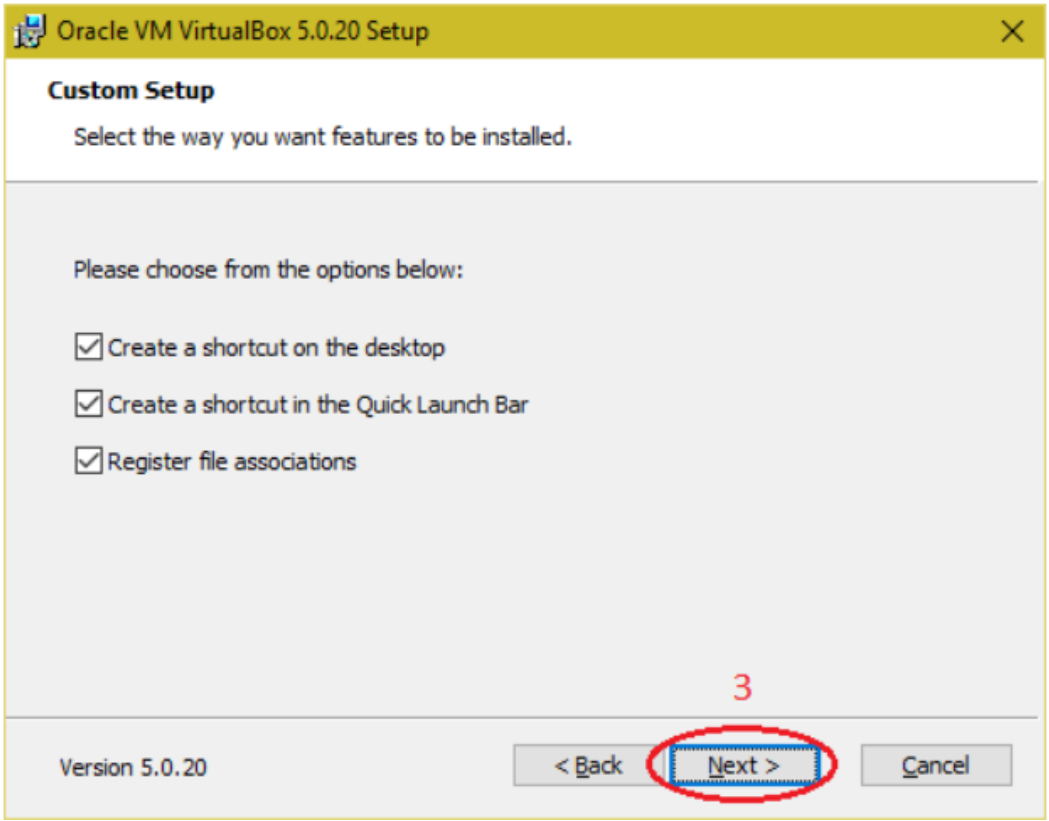
Step 2 – Run the virtual box setup and click on “Next” button.



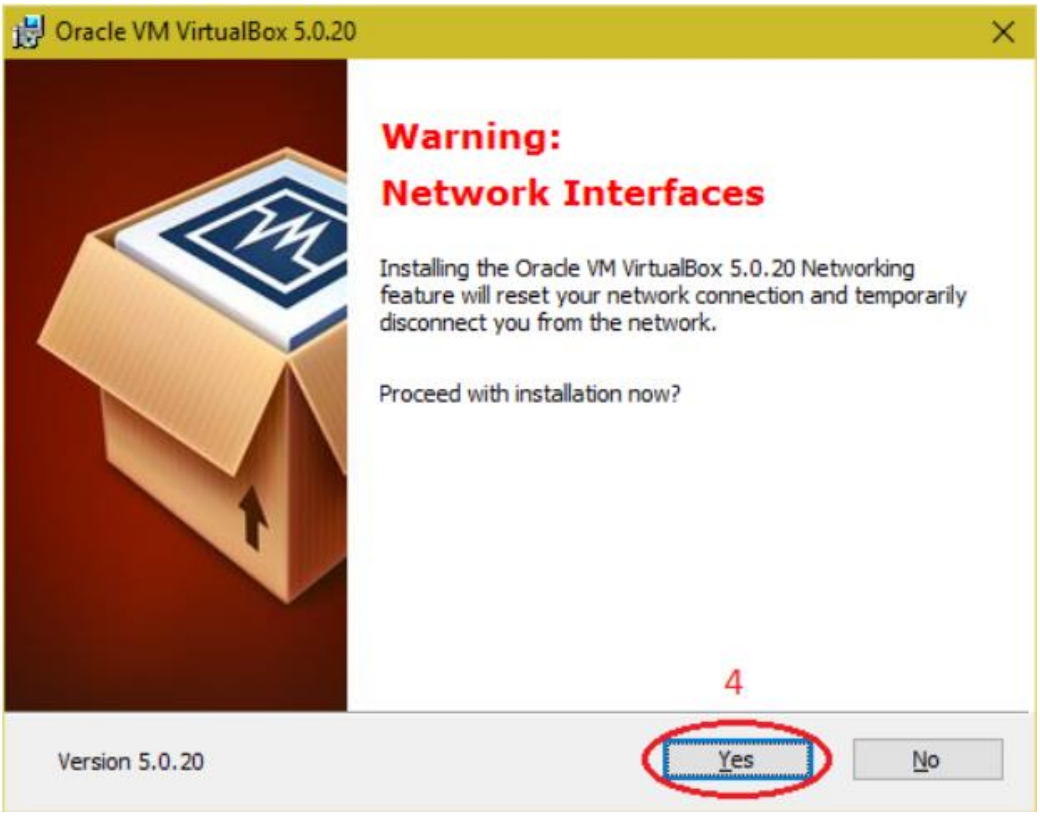
Step 3 – Click on "Next" button.



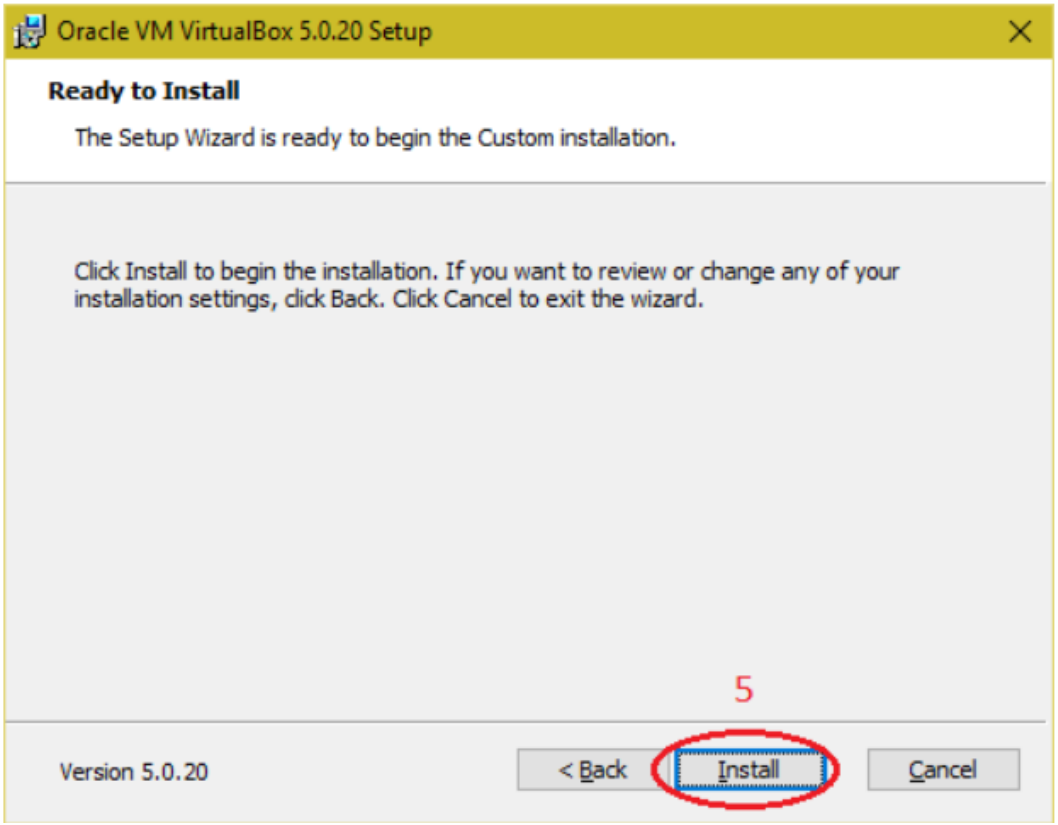
Step 4 – Click on "Next" Button



Step 5 – Click on "Yes" Button

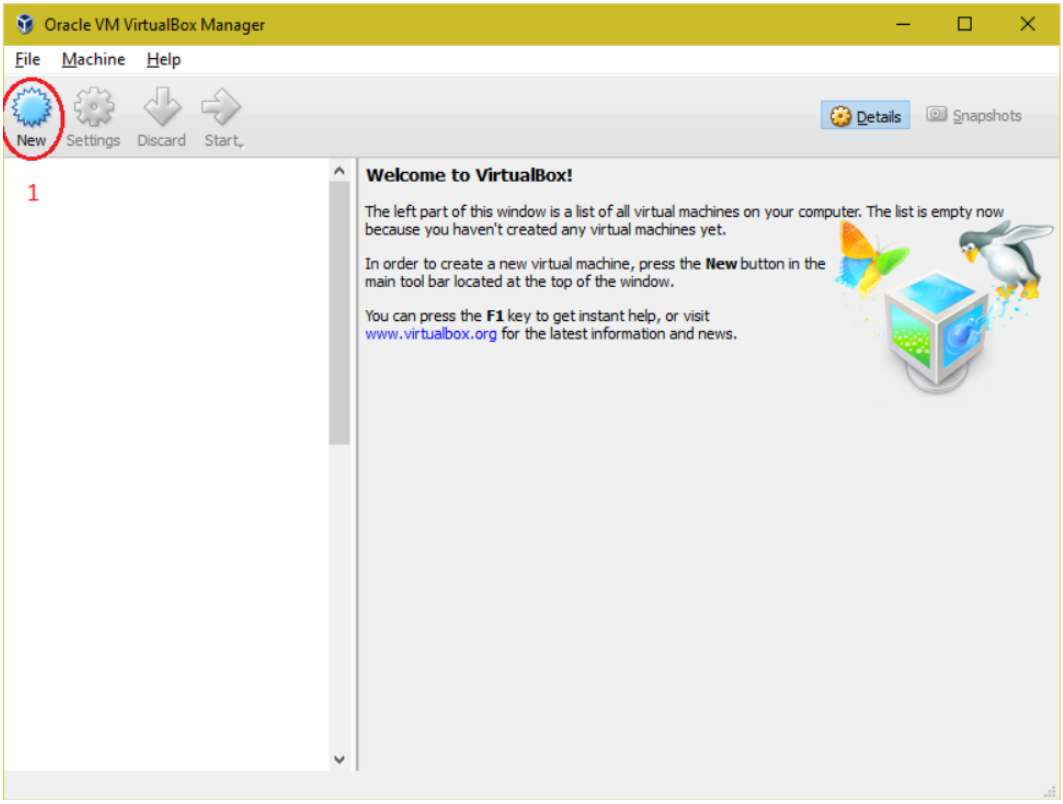


Step 6 – Click on "Install" Button

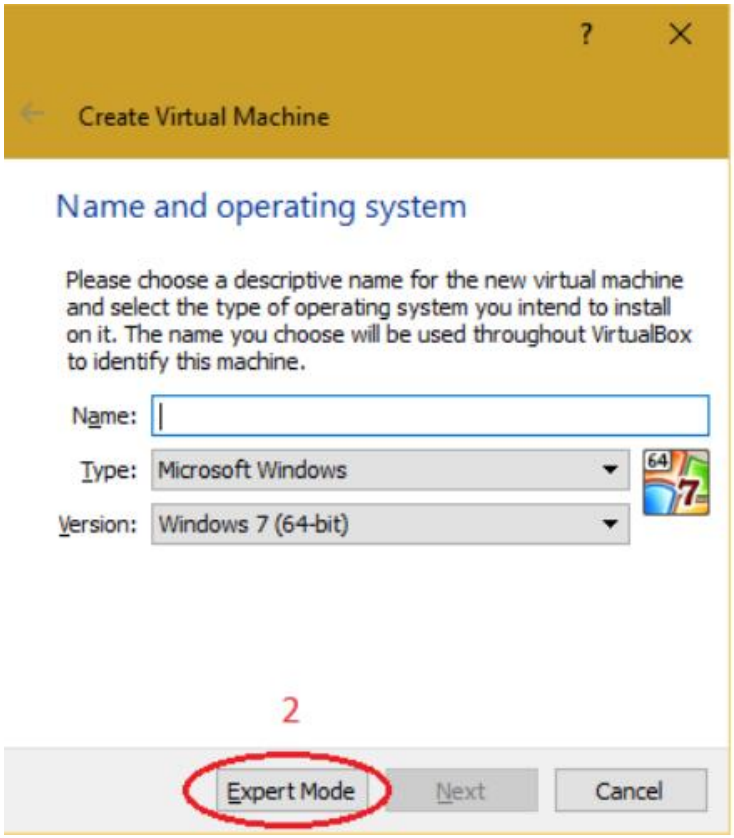


INSTALLING "UBUNTU" AS VIRTUAL MACHINE IN "ORACLE VM VIRTUALBOX".

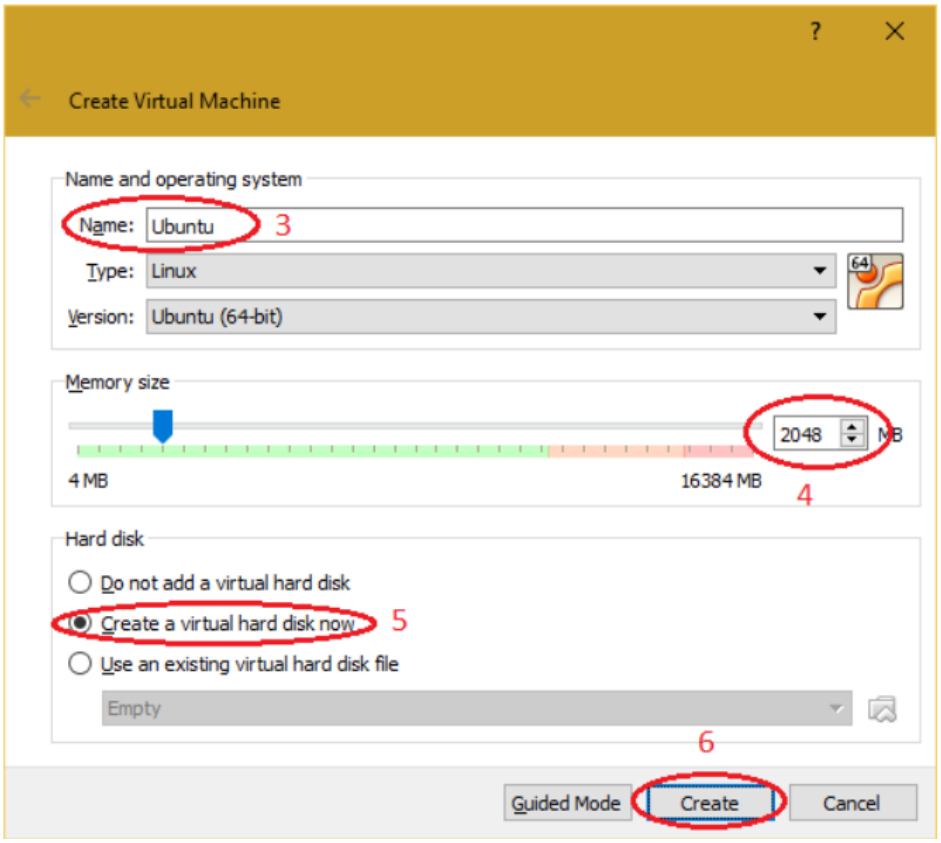
Step 7 – Open "Oracle Virtual Box Manager"



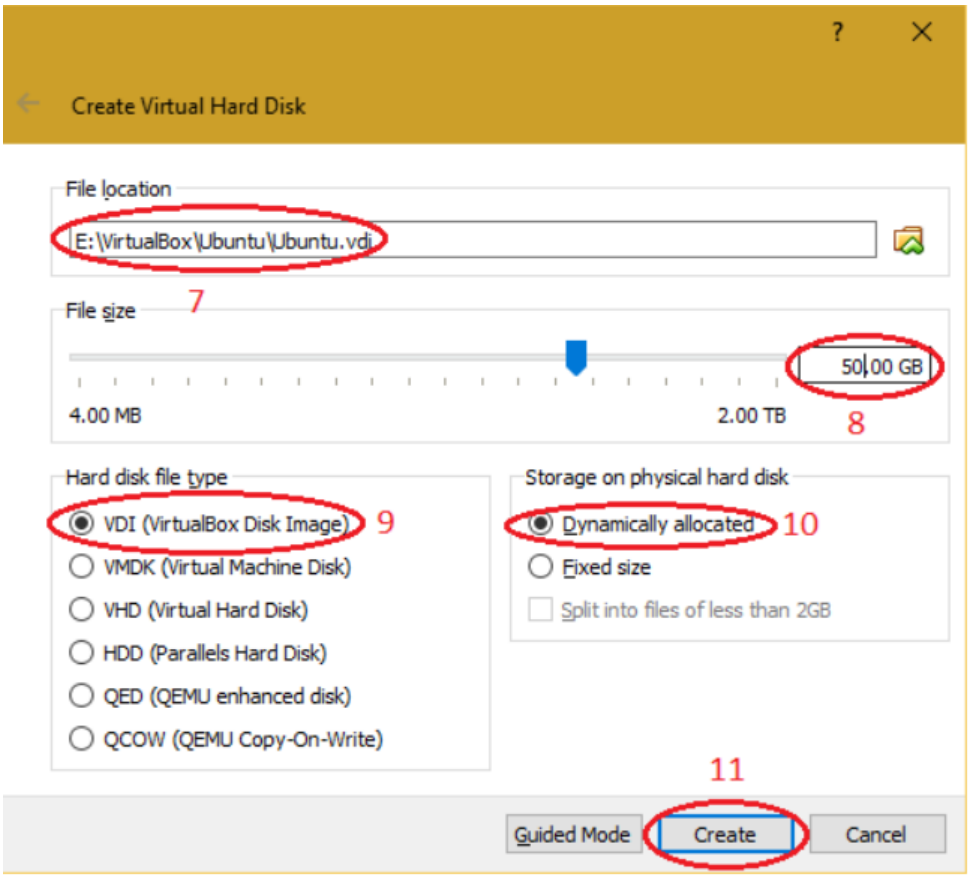
Step 8 – Click on "New" Button and select "Expert Mode".



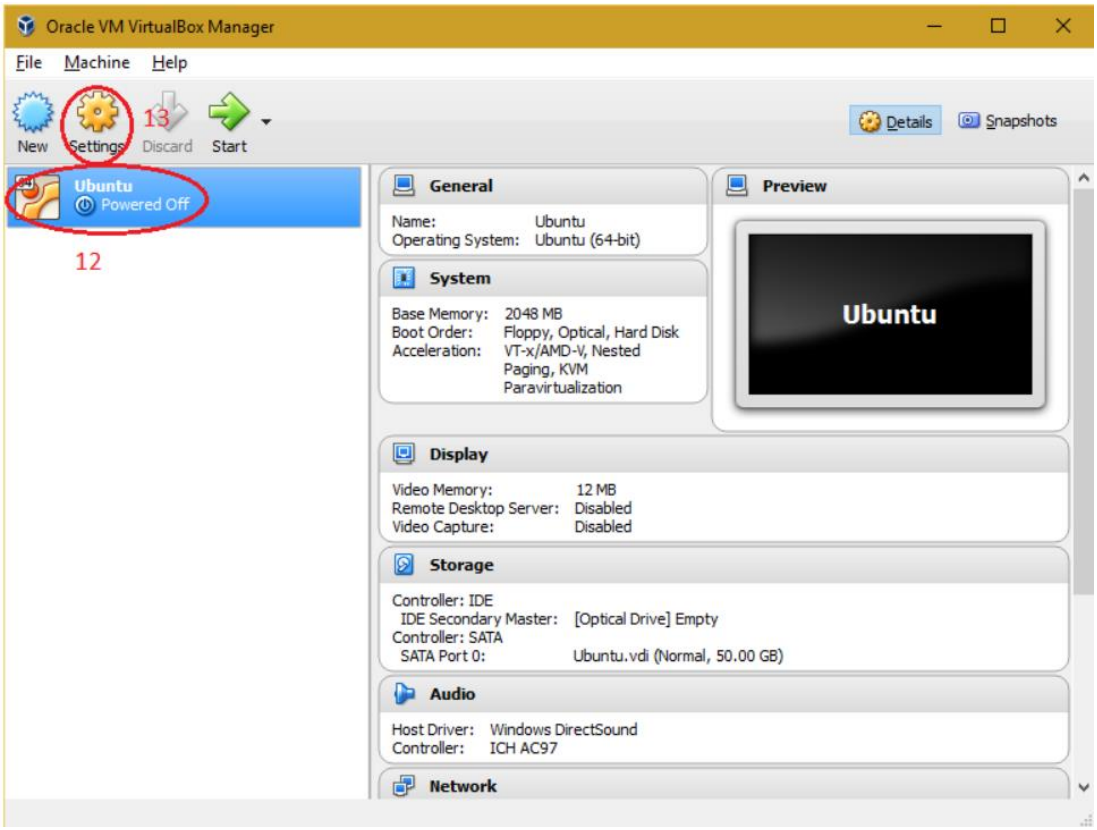
Step 9 – Provide the name and operating system information for virtual machine.



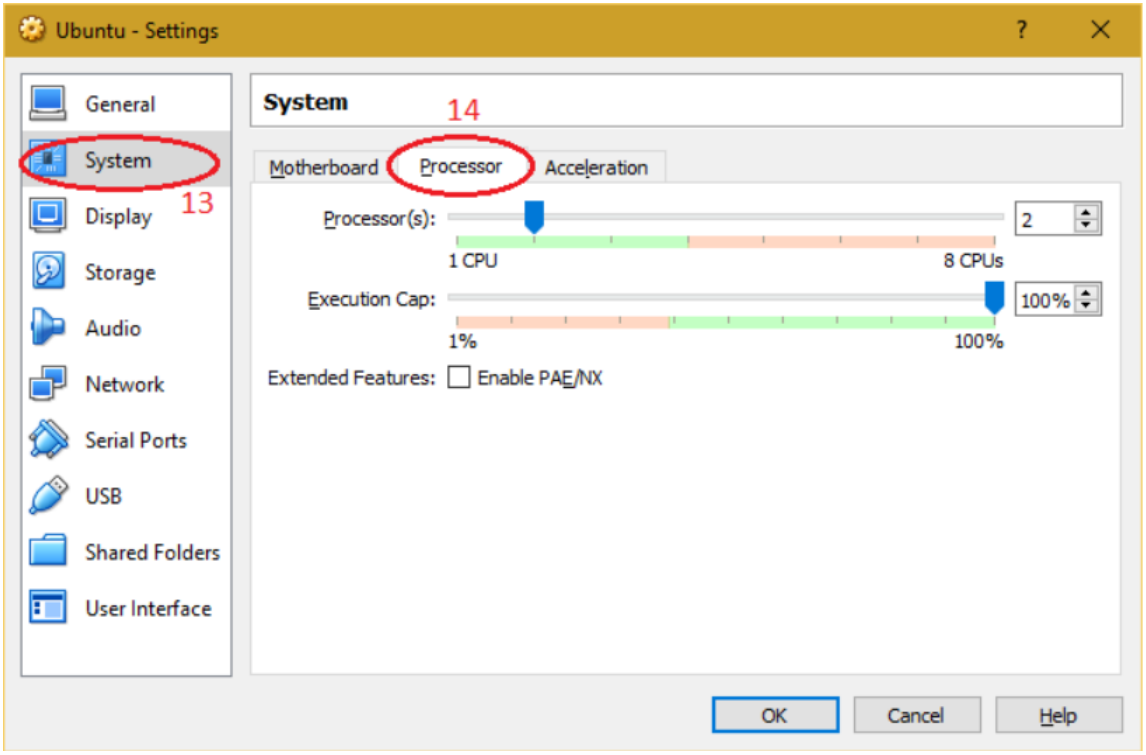
Step 10 – Select the path for virtual hard disk and click on "Create" button.



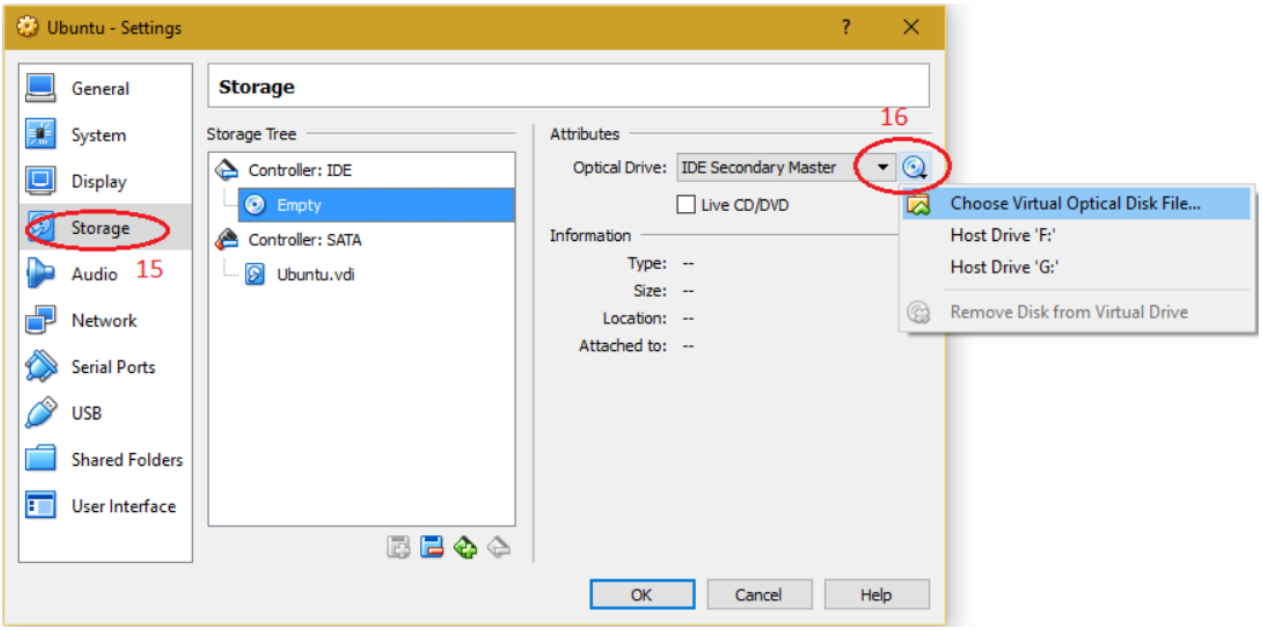
Step 11 – Select the virtual machine from the Virtual Box Manager and click on "Settings" Button.

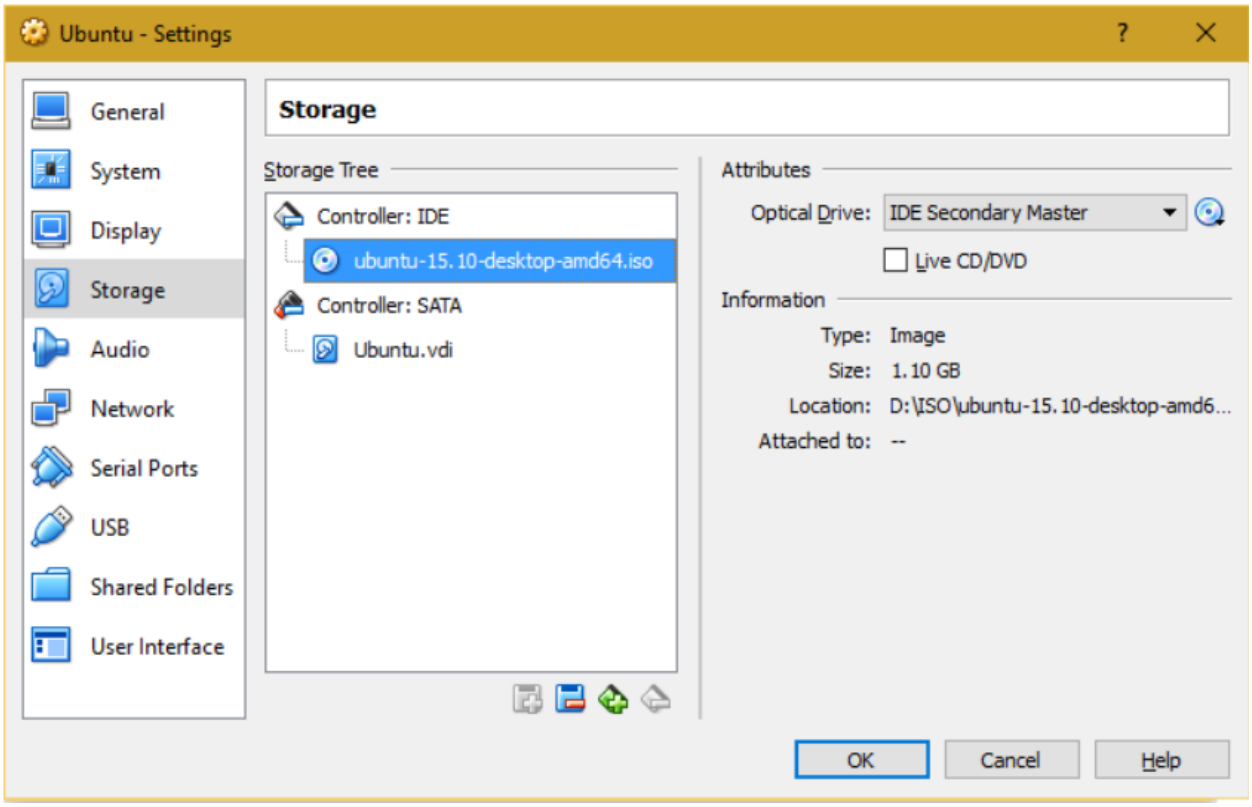


Step 12 – Select "System" and navigate to "Processor" tab to adjust number of processor of virtual machine for better performance.

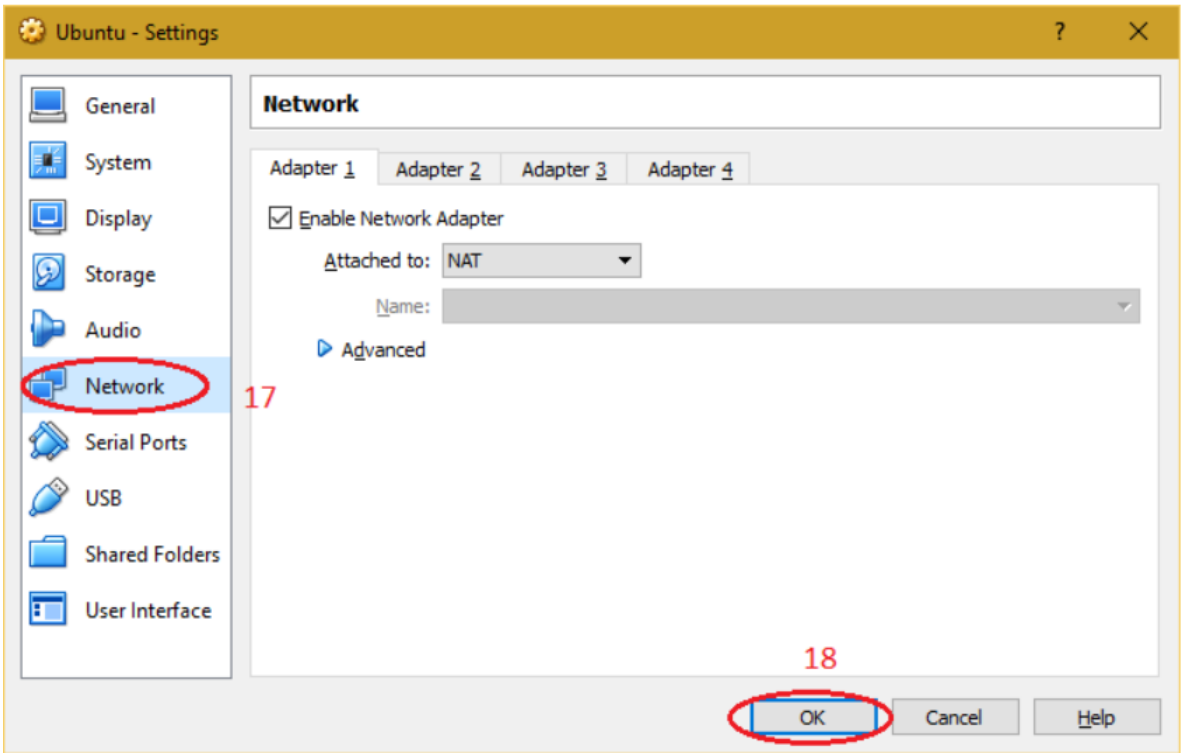


Step 13 – Select "Storage" and choose the installation media of Operating System (ISO/CD/DVD/USB). Preferred Linux ".iso" can be downloaded from https://ubuntu.com/download/desktop. Also many flavours of Linux are available on the internet – Fedora, CentOS, Ubuntu, Debian, Mageia, openSUSE, ArchLinux, Slackware Linux, etc.

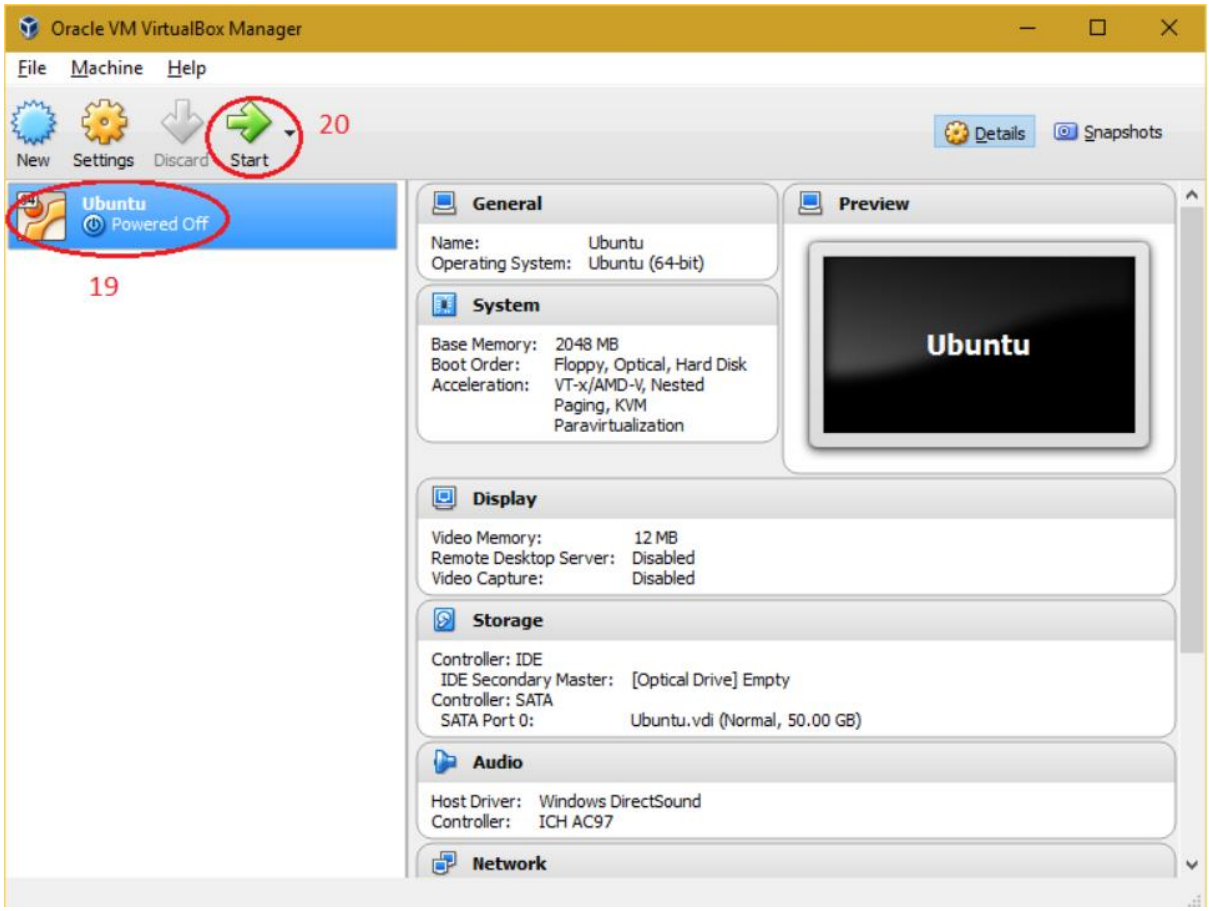




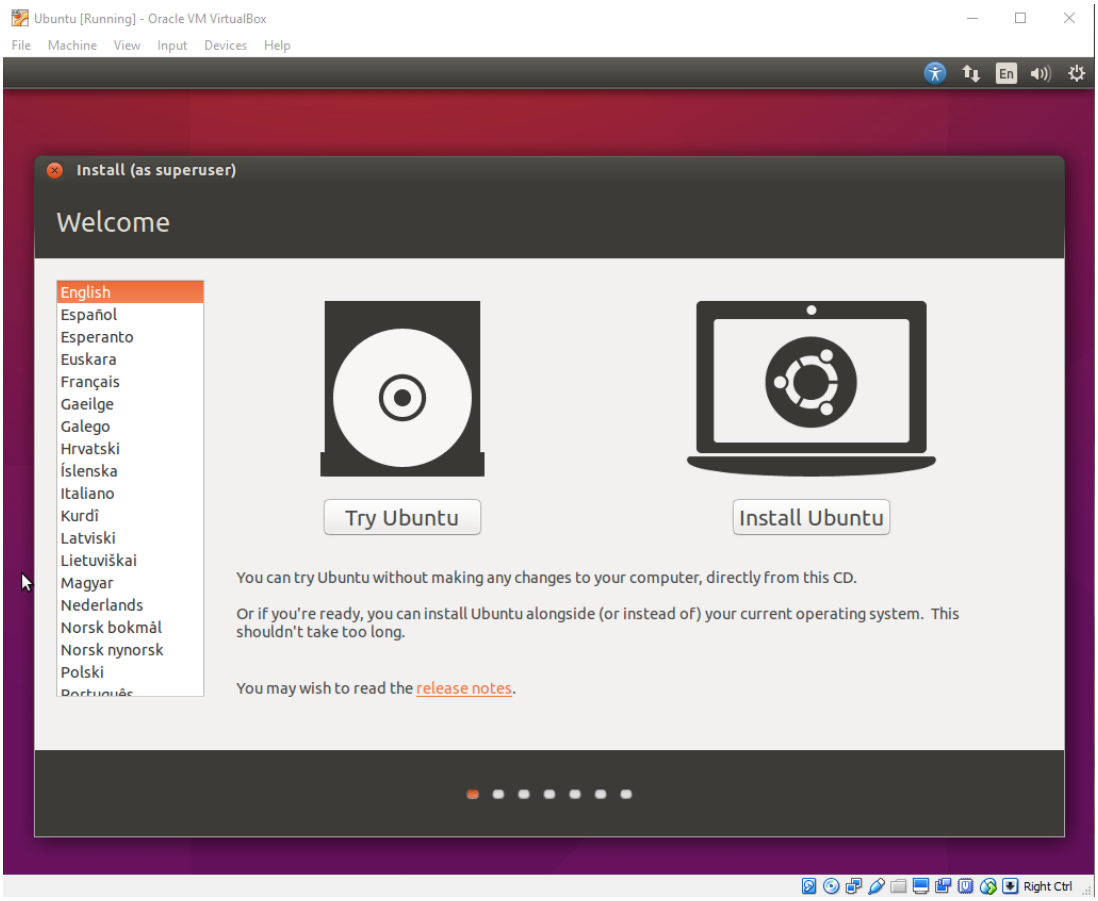
Step 14 – Select "Network" to make changes required for network setting of virtual machine and click on "OK".



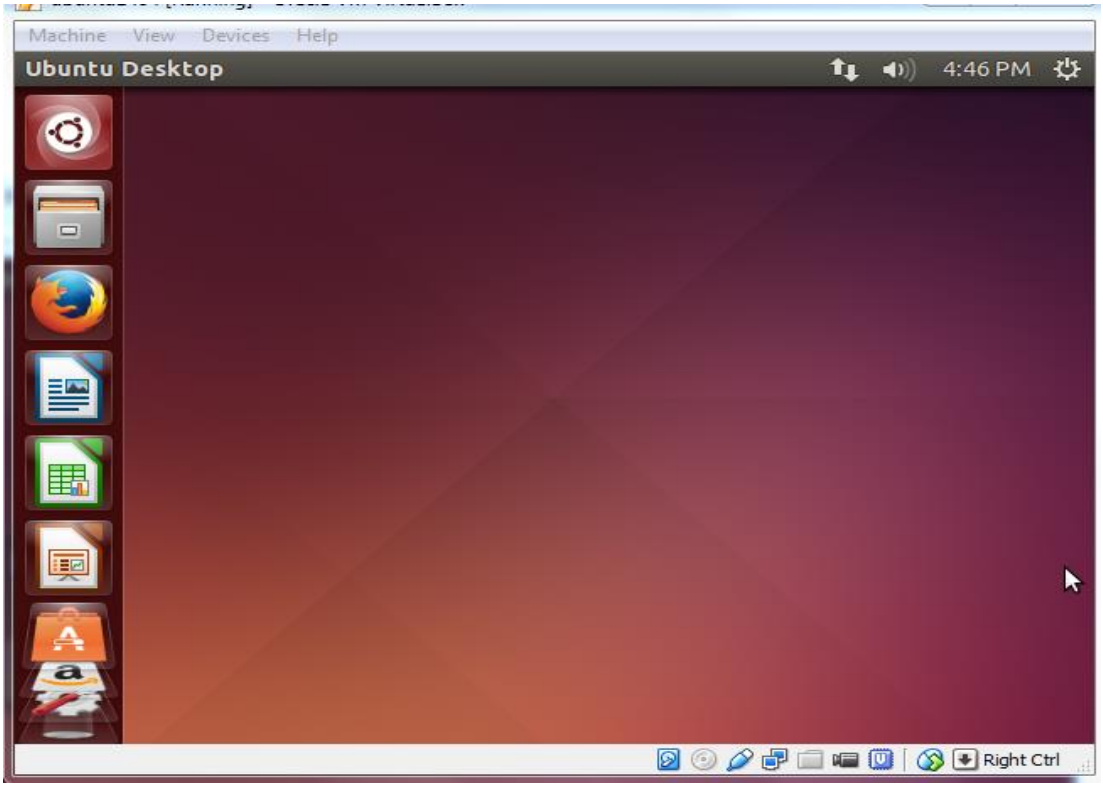
Step 15 – Select the created virtual machine and click on "Start" button.



Step 16 – Proceed with the installation of operating system in virtual machine.



Step 17 – The Final Linux Desktop Virtual OS will be seen as below:



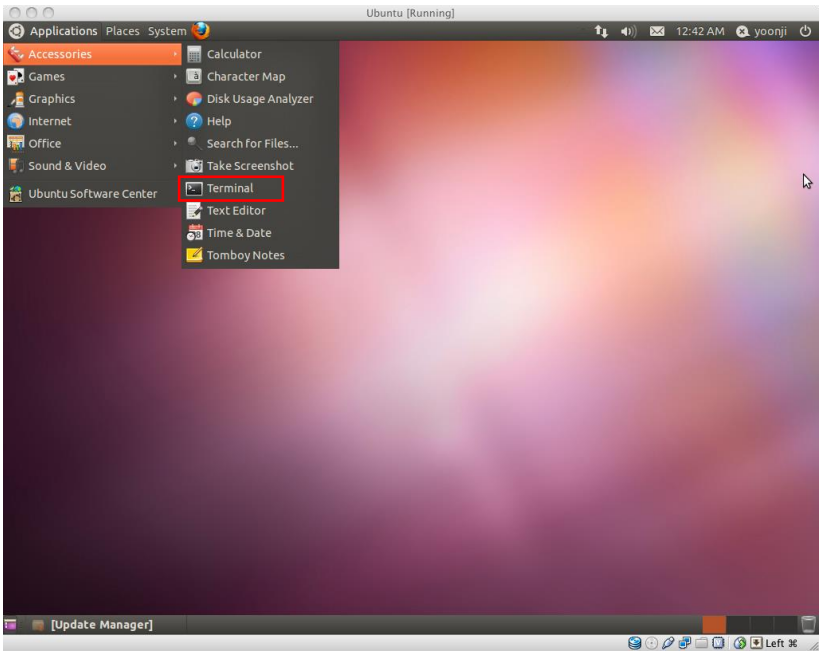
EX. No:2 **C Programming on Linux in VirtualBox**

Aim:

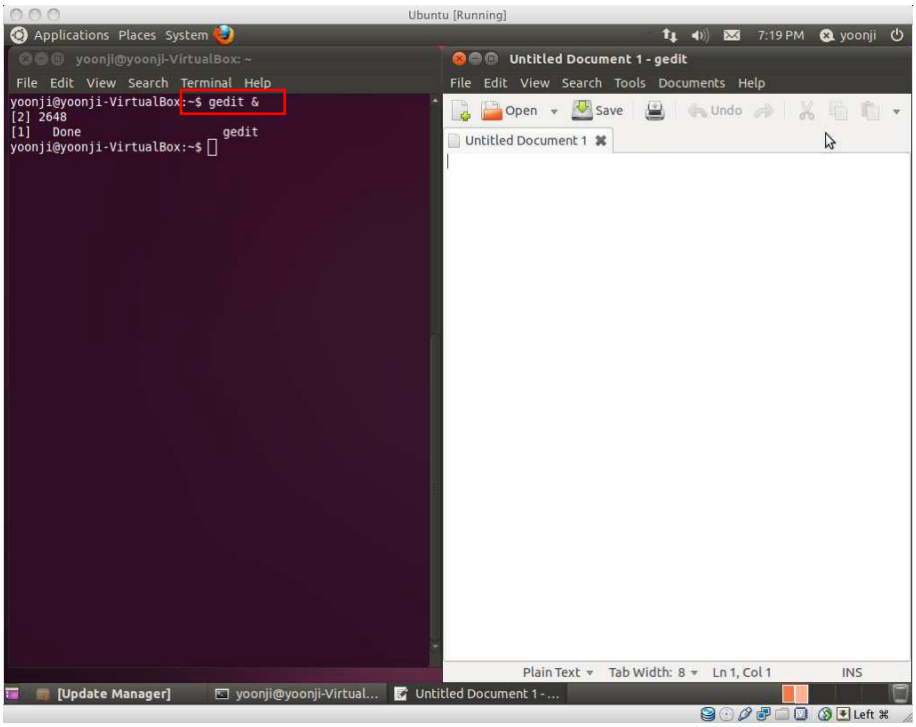
To find a procedure to install a C compiler in the virtual machine created using virtual box and execute Simple Programs

Procedure:

1. Open Terminal from "Applications -> Accessories -> Terminal".



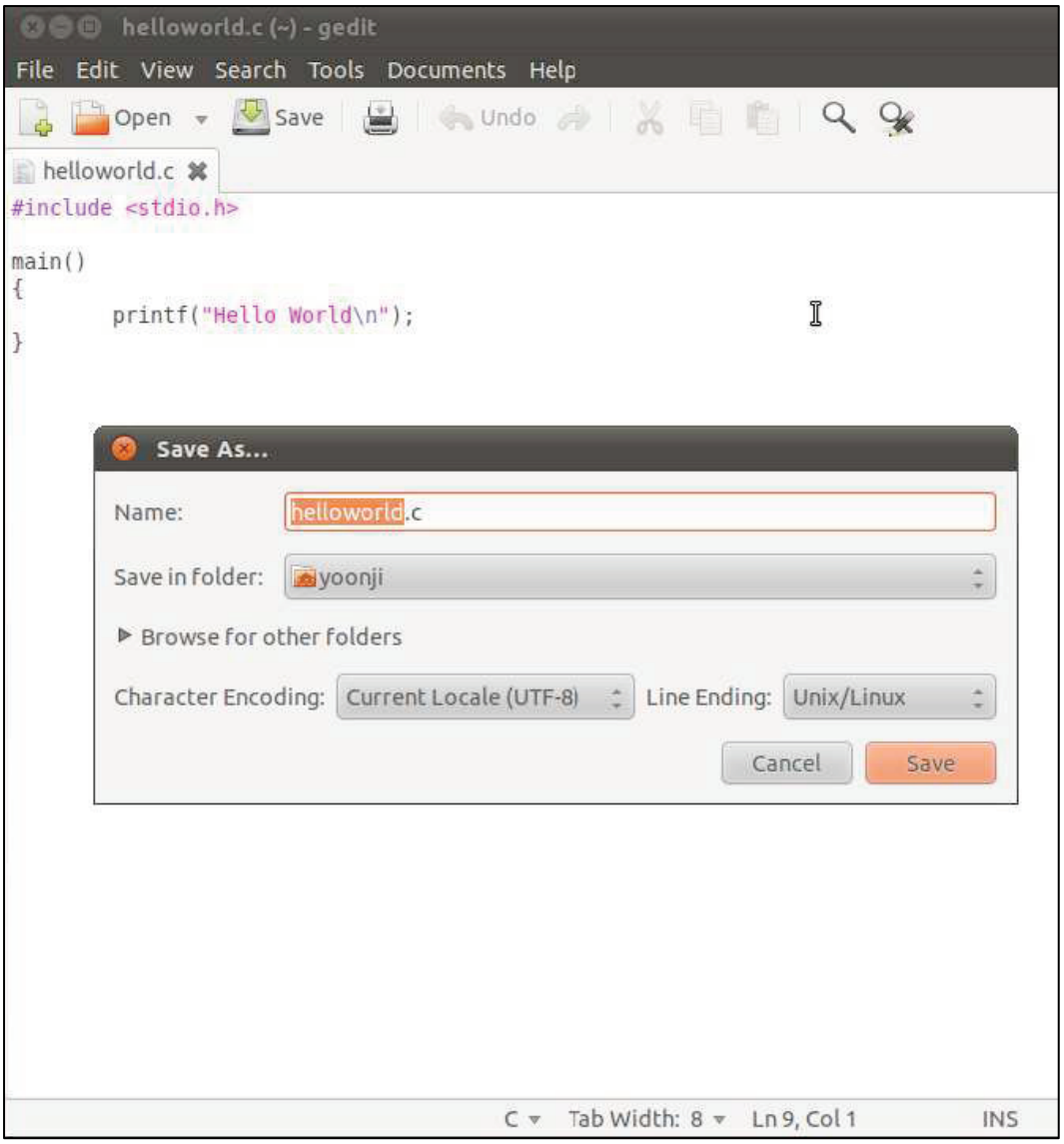
2. Open gedit by "gedit &" on terminal or we can use any other Text Editor Application.



3. Type the following code on gedit (or using any other Text Editor)

```
#include<stdio.h>
main(){
    printf("Hello World\n");
}
```

4. Save this file as "helloworld.c".

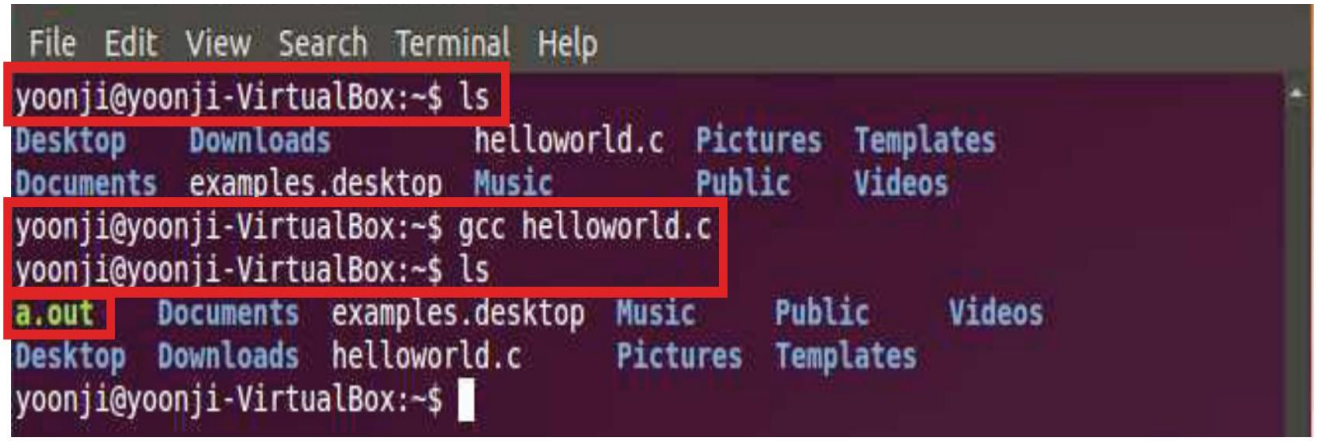


5. Then go to terminal and type “ls” on terminal to see all the files under current folder.

6. Make sure that the file we created “helloworld.c” is in the current directory or not. If not, type cd <directory_path> to go the directory which contains “helloworld.c”

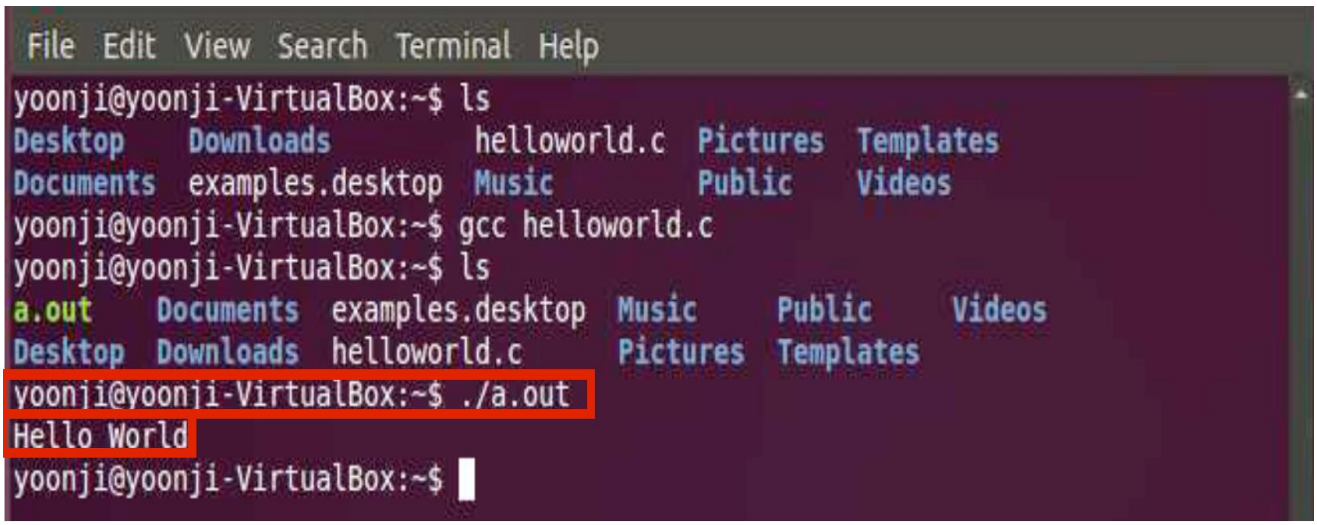
7. Now type “gcc helloworld.c” to compile and now this will create a new executable file called as “a.out”.

8. Then type “ls” to confirm that the executable file “a.out” is created.



9. Now type “./a.out” on terminal to run the program.

10. Here we have to see “Hello World” on the next line.



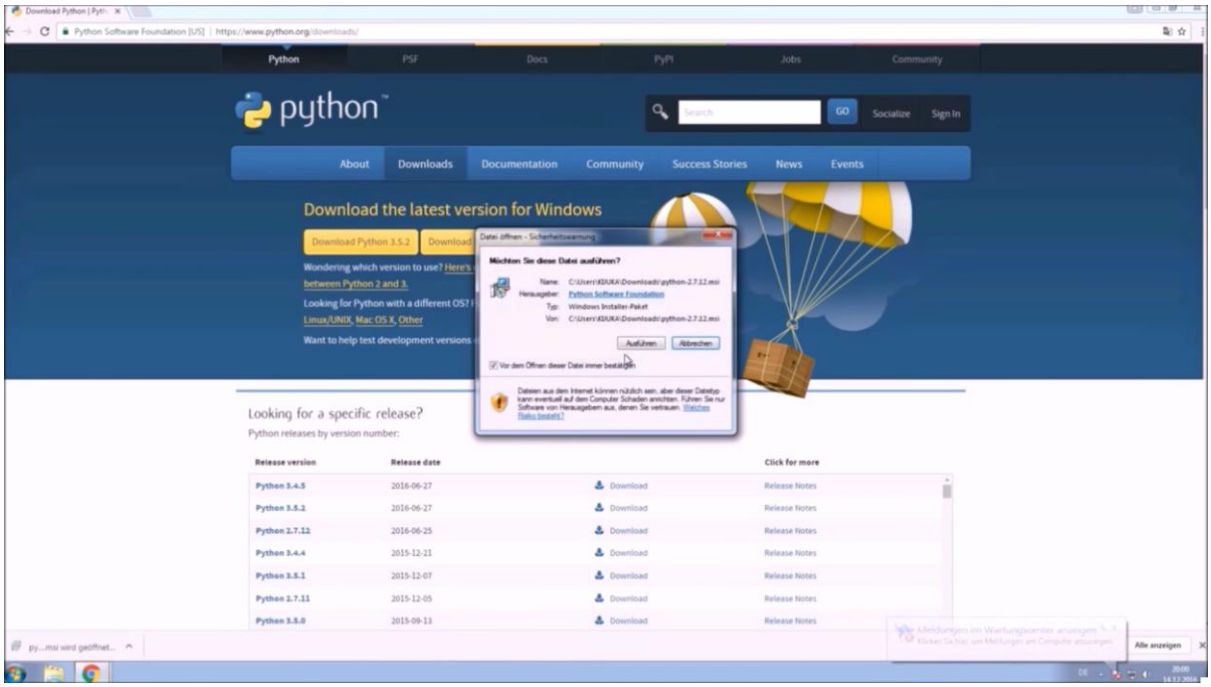
EX. No:3 Installing and Running the Google App Engine On Windows

Aim:

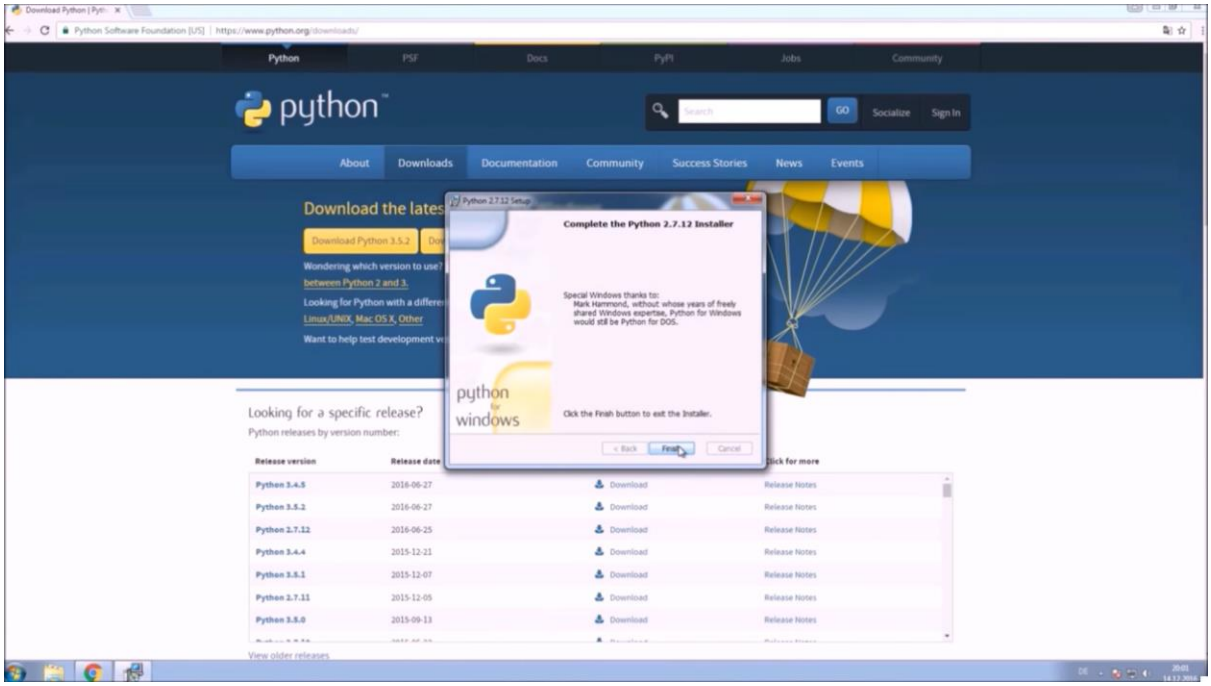
To find a procedure to Install Google App Engine. Create hello world app and other simple web applications using python/java.

Procedure:

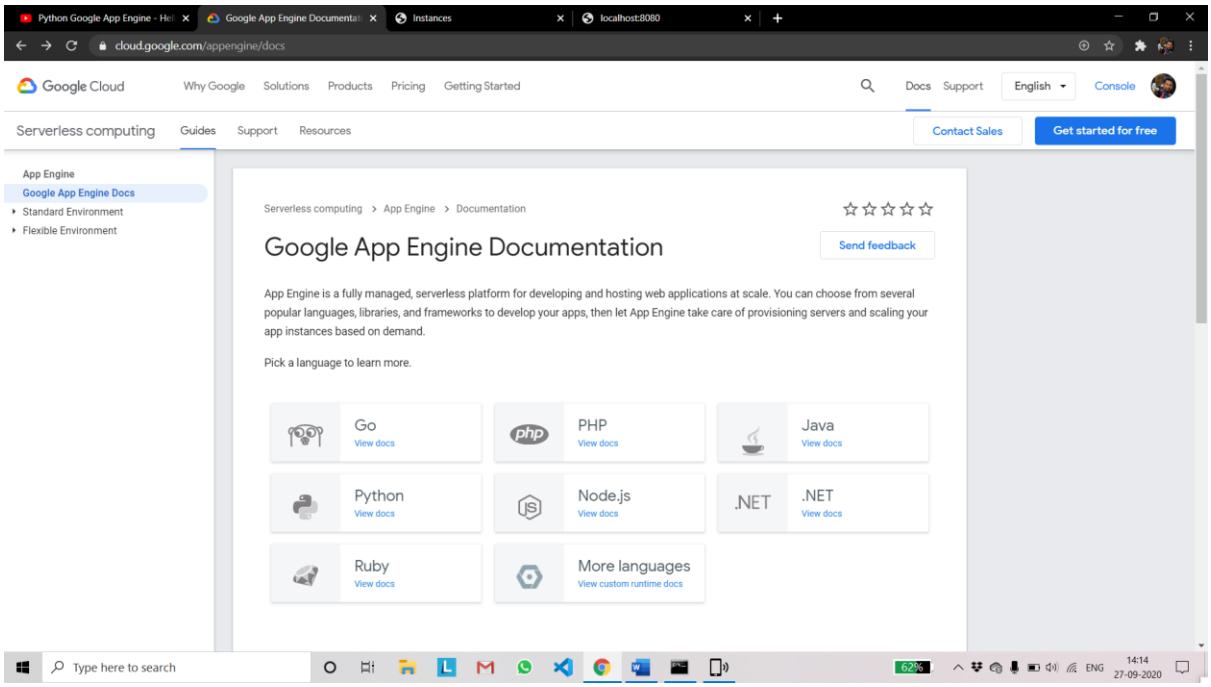
1. Start the project by installing Python in the below website: https://www.python.org/downloads/



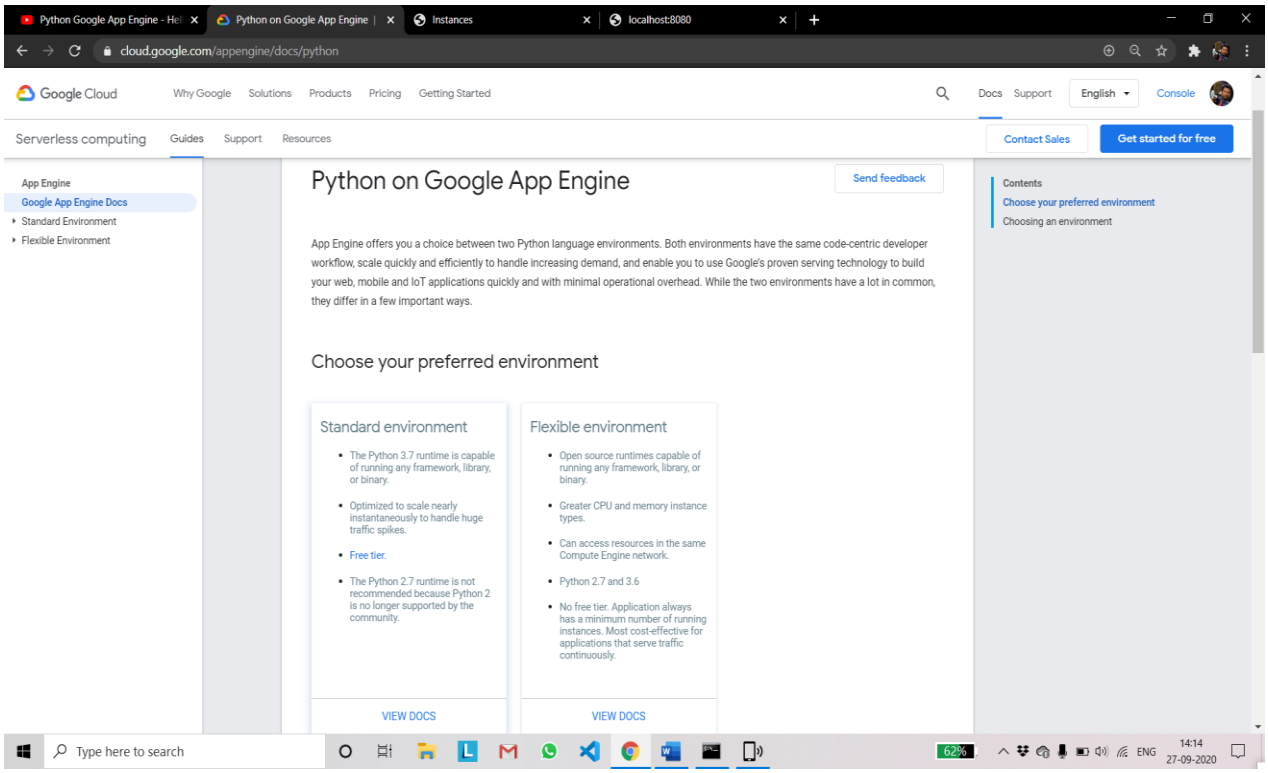
2. Install Python and click “Finish” after installation.



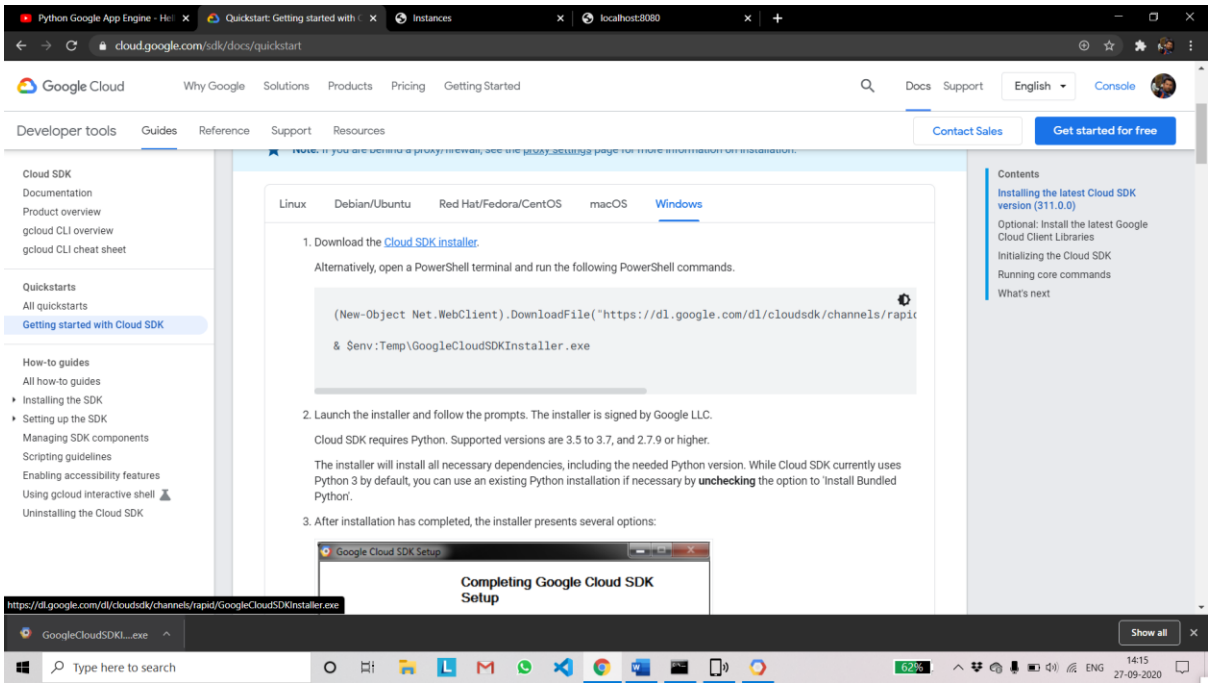
3. Go into “https:www.cloud.google.com/appengine/docs” and click for “Python”



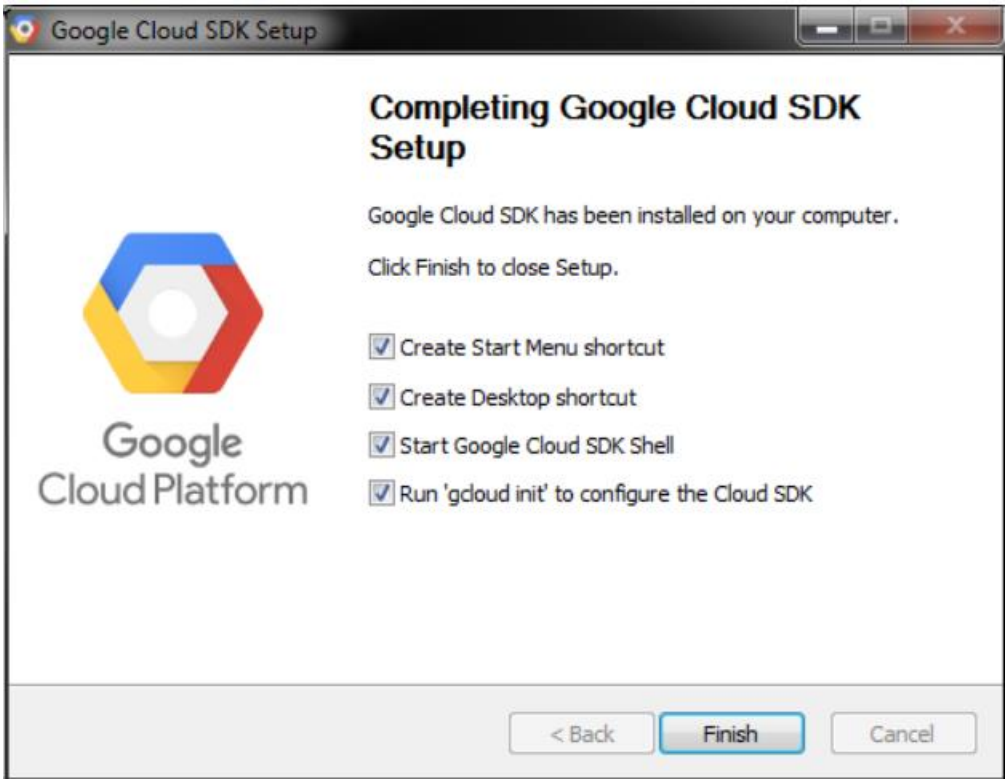
4. On the Python page, Click for “Standard Environment” which gives Python27 Environment.



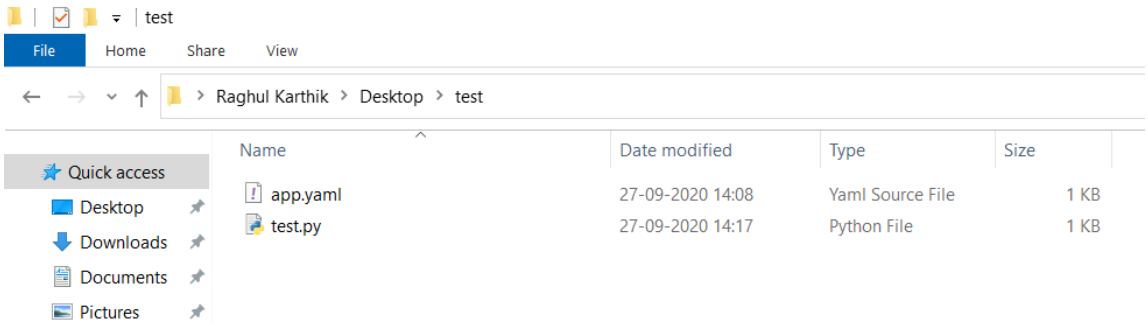
5. Now download the “Google Cloud SDK Installer” and proceed to install it.



6. After installing it, click on “Finish”

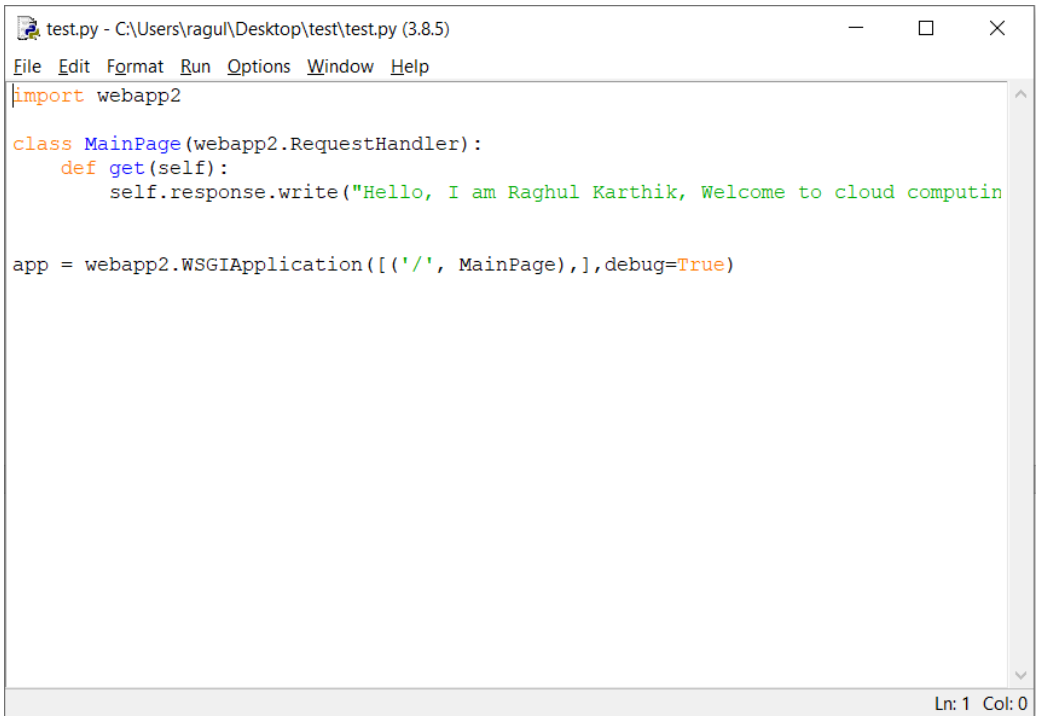


7. On the desktop, create a new folder called “Test”. Inside it create new files called as “test.py” and “app.yaml”



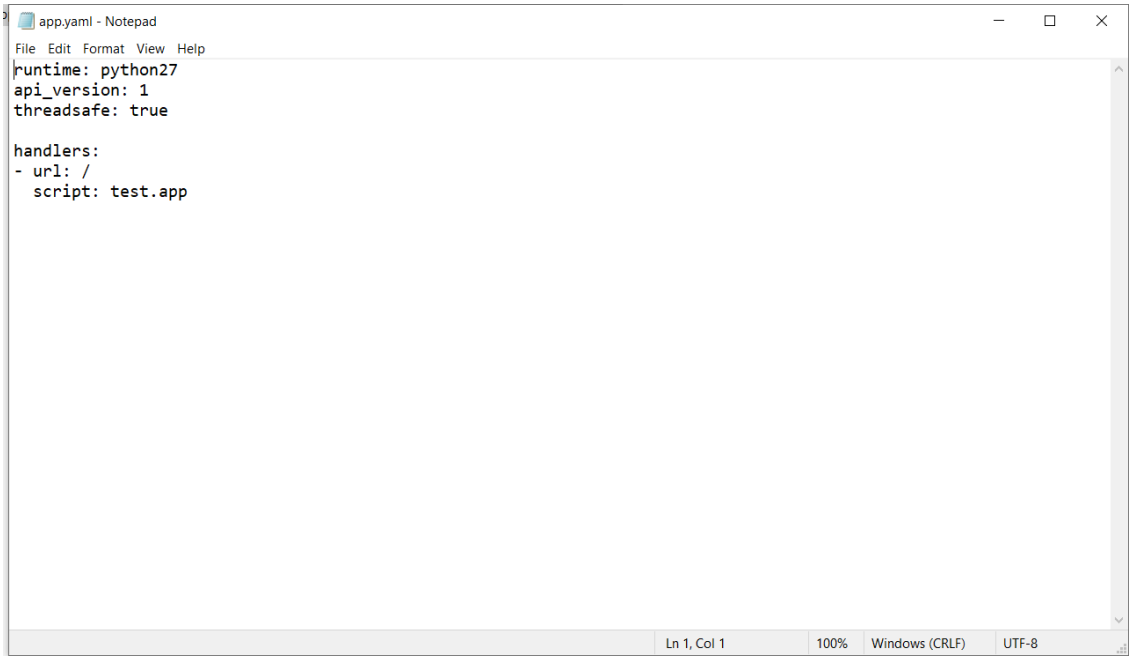
8. Then in “test.py”, write the code as follows:

```
import webapp2

class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.write("Hello, I am Raghul Karthik, Welcome to cloud computing laboratory")


app = webapp2.WSGIApplication([('/', MainPage),],debug=True)
```



9. Then in “app.yaml” file, write the code as follow:

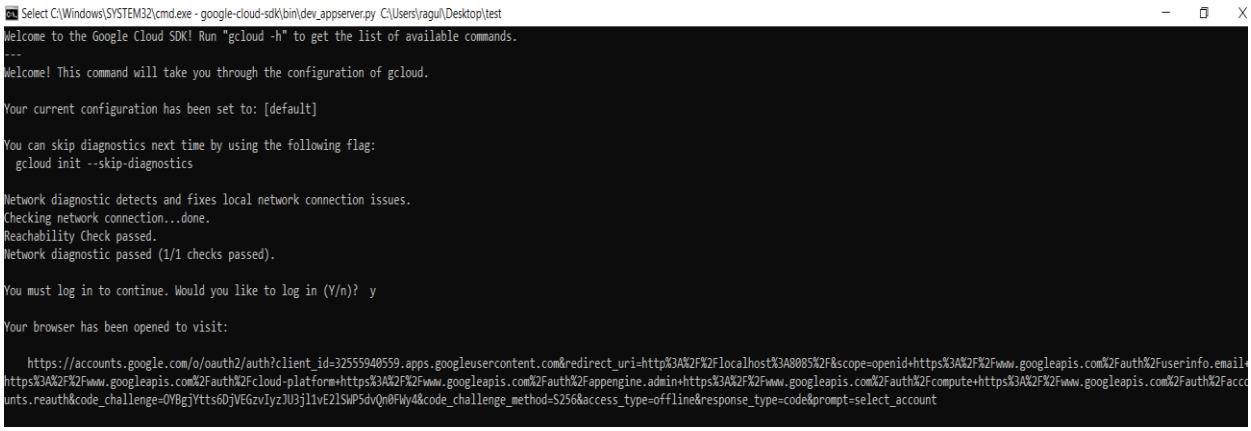
```
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /
  script: test.app
```



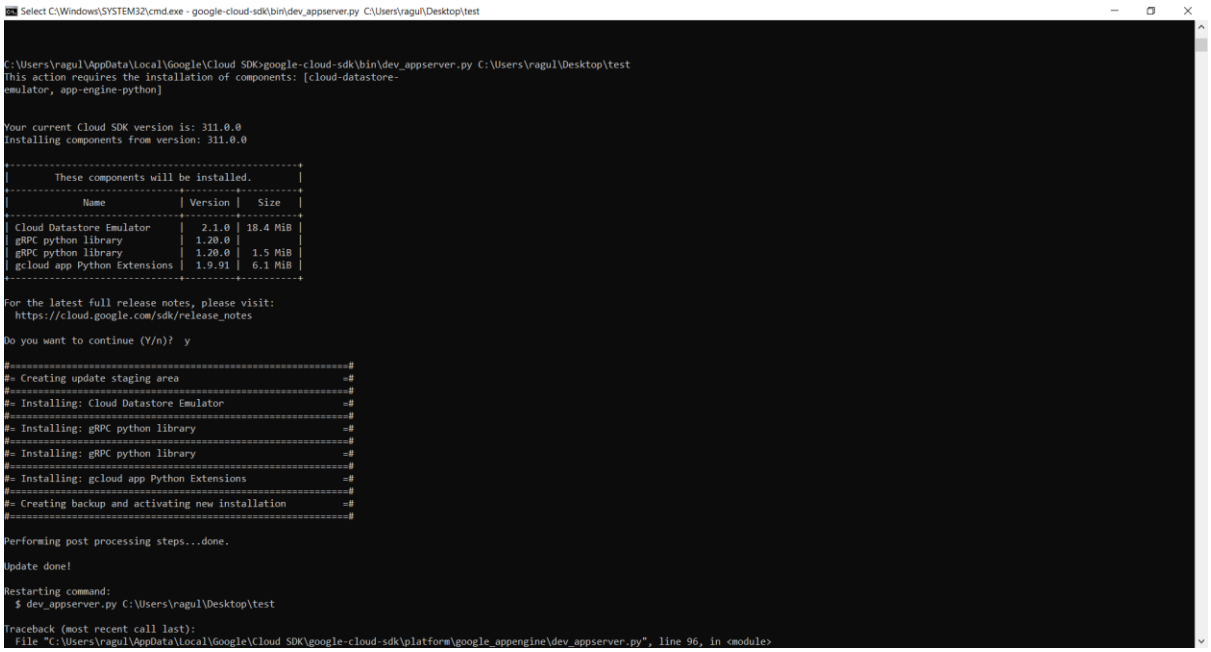
10. Into the console of Google Cloud SDK Console, type as follows:

gcloud init

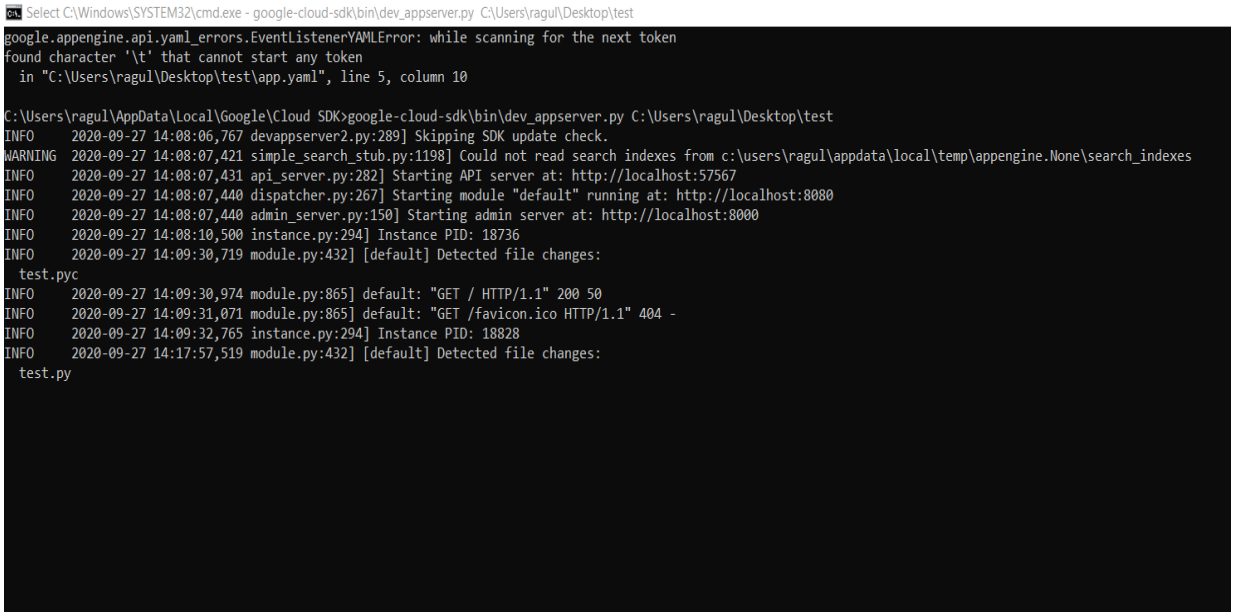
It will prompt us to sign in and sign in with your google account.



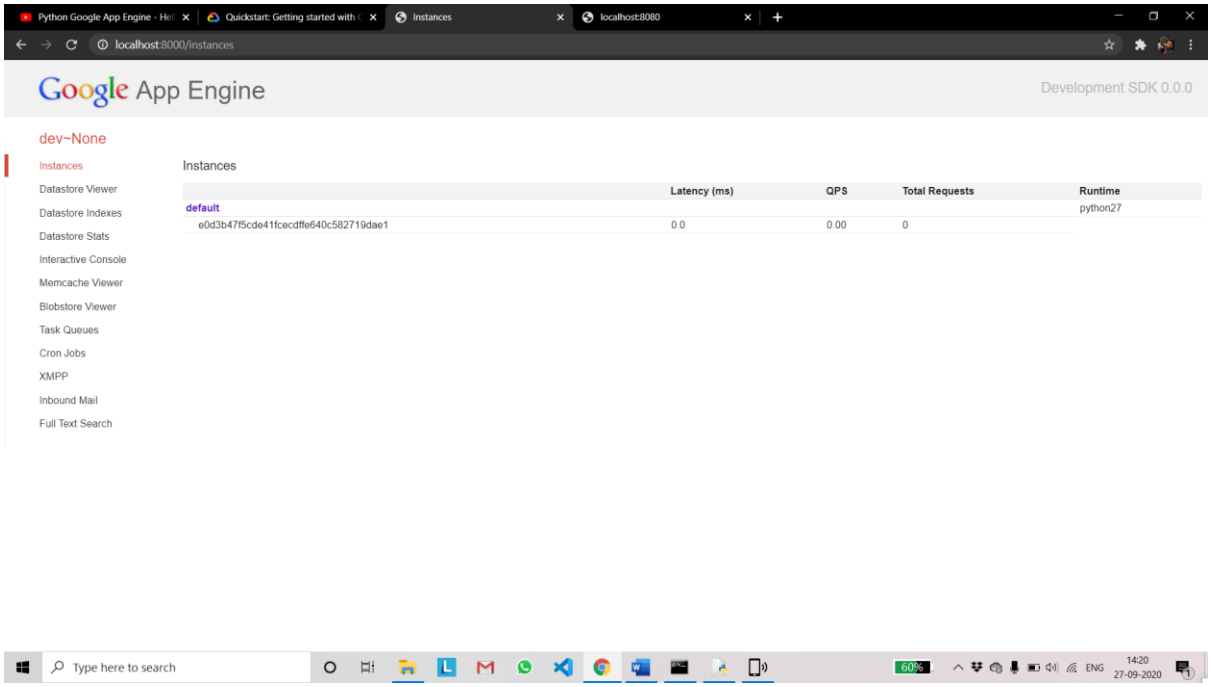
11. Write the following command to console:
    google-cloud-sdk\bin\dev_appserver.py C:\Users\ragul\Desktop\test
    and answer “Do You want to continue?” as “Yes”.



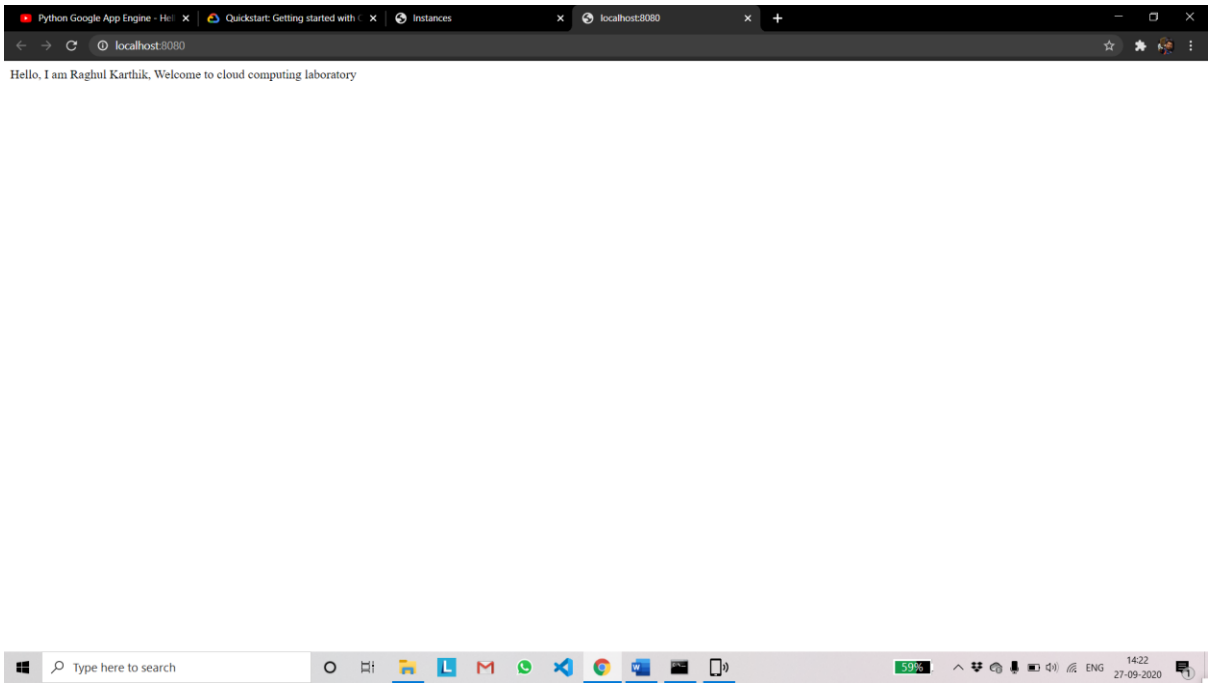
It will start a local server at the port 8000. We can view it by http://localhost:8000.



12. Go to the “localhost:8000” and this will shown as follows:



13. When clicking “default” option, it will redirect us to the file where we can see the output.



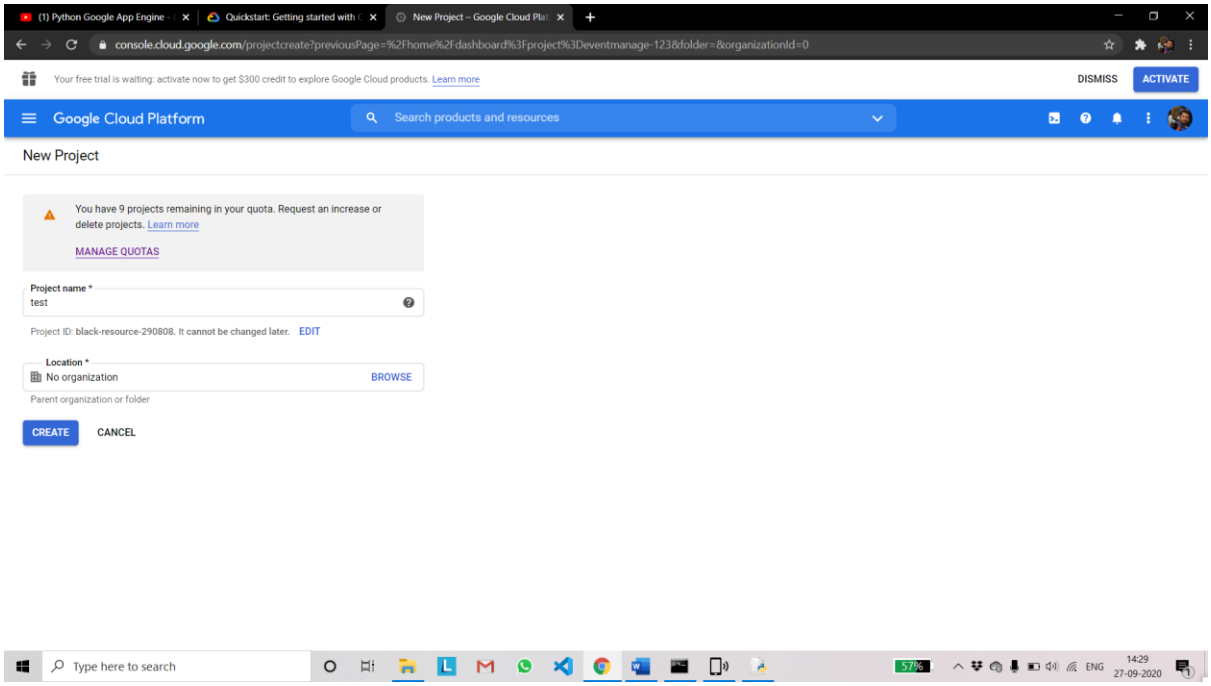
EX. No:4 Installing and Running the Google App Engine On Windows

Aim:

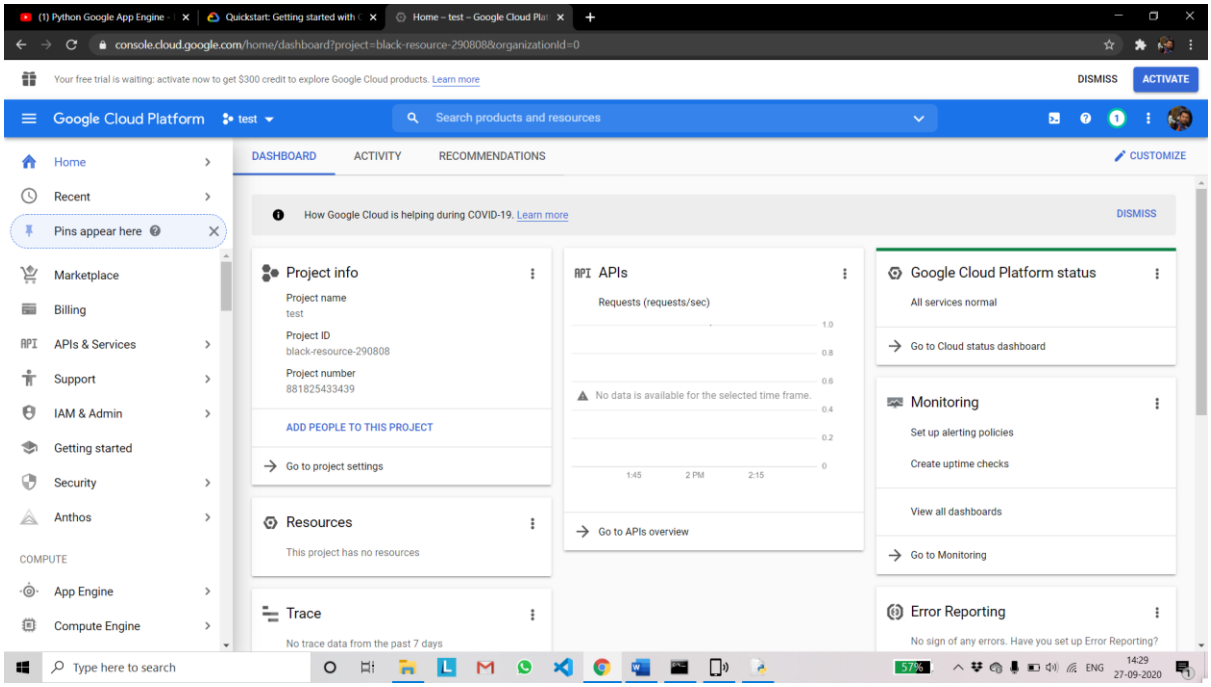
To find a procedure to Install Google App Engine. Create hello world app and other simple web applications using python/java and Use GAE launcher to launch the prepared web application.

Procedure:

1. Open “Google Cloud Platform” by typing “console.cloud.google.com” and create a new prooject.



2. After creating new project, you are redirected to project dashboard and copy the project ID.



3. Enter the following command in the console:

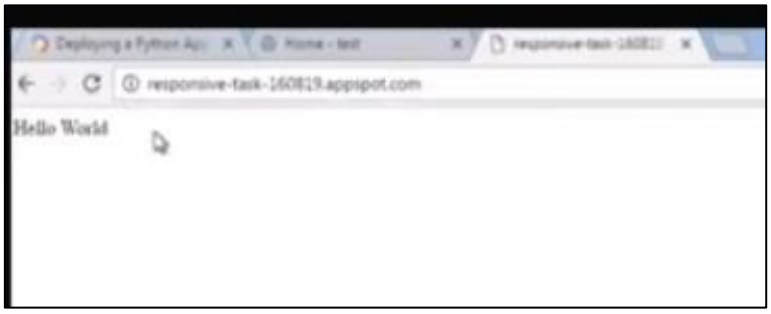
   gcloud app deploy --project <PROJECT_ID>

   and then select the server region and answer “Yes”.

```
C:\Users\ragul\Desktop\test>gcloud app deploy --project black-resource-290808
You are creating an app for project [black-resource-290808].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

Please choose the region where you want your App Engine application
located:

 [1] asia-east2
 [2] asia-northeast1
 [3] asia-northeast2
 [4] asia-northeast3
 [5] asia-south1
 [6] asia-southeast2
 [7] australia-southeast1
 [8] europe-west
 [9] europe-west2
 [10] europe-west3
 [11] europe-west6
 [12] northamerica-northeast1
 [13] southamerica-east1
 [14] us-central
 [15] us-east1
 [16] us-east4
 [17] us-west2
 [18] us-west3
 [19] us-west4
 [20] cancel
Please enter your numeric choice:  2

Creating App Engine application in project [black-resource-290808] and region [asia-northeast1]....done.
Services to deploy:

descriptor:      [C:\Users\ragul\Desktop\test\app.yaml]
source:          [C:\Users\ragul\Desktop\test]
target project:  [black-resource-290808]
target service:  [default]
target version:  [20200927t143251]
target url:      [https://black-resource-290808.an.r.appspot.com]


Do you want to continue (Y/n)?  y
```

```
Updating service [default]...done.
Deployed service [default] to [https://responsive-task-160819.appspot.com]

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

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

4. Now the application has been deployed to the server and you can view it by entering the url that was generated during the deployment.
5. The output was shown as below:



EX. No:5 Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Aim:

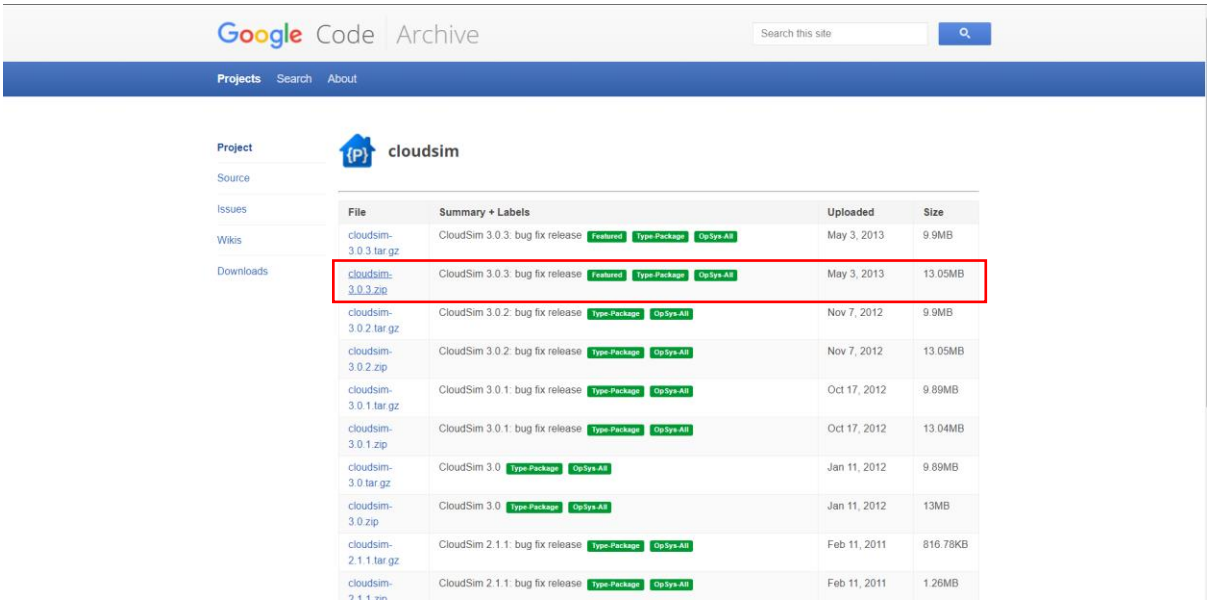
To Simulate a cloud scenario using CloudSim and run a scheduling algorithm (Shortest Job First – SJF) that is not present in CloudSim.

Procedure:

1. CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful.
2. It is a library and, hence, CloudSim does not have to be installed.
3. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used.
4. Please verify whether Java is available on your system.To use CloudSim in Eclipse:
5. Download CloudSim installable files from

   https://code.google.com/p/cloudsim/downloads/list

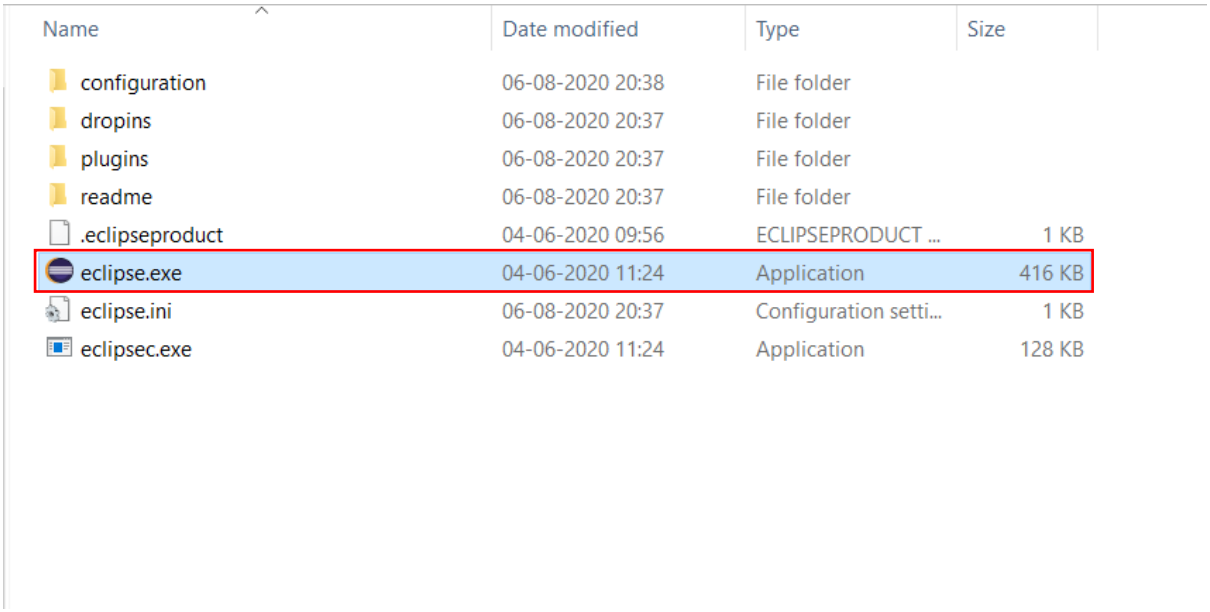
   and unzip



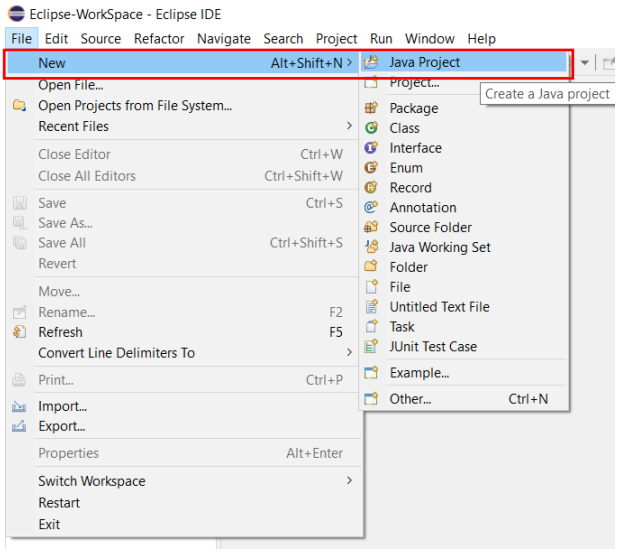
6. Cloudsim simulation toolkit setup is easy. Before you start to setup CloudSim, following resources must be Installed/downloaded on the local system
7. **Java Development Kit(JDK):** As the Cloudsim simulation toolkit is a class library written in the Java programming language, therefore, the latest version of Java(JDK) should be installed on your machine, which can be downloaded from Oracles Java portal. For assistance in the installation process, detailed documentation is provided by Oracle itself and you may follow the installation instructions
8. **Eclipse IDE for Java developers:** As per your current installed operating system(Linux/Windows). Before you download to make sure to check if 32-bit or 64-bit version is applicable to your Computer machine. Link for Eclipse Kepler version is available at the following link
9. **Download CloudSim source code:** To date, various versions of CloudSim are released the latest version is 5.0, which is based on a container-based engine. Whereas to keep the setup simple for beginners we will be setting up the most used

version i.e. 3.0.3, which can be directly downloaded by clicking on any of the following: Click for Windows or click for Linux.

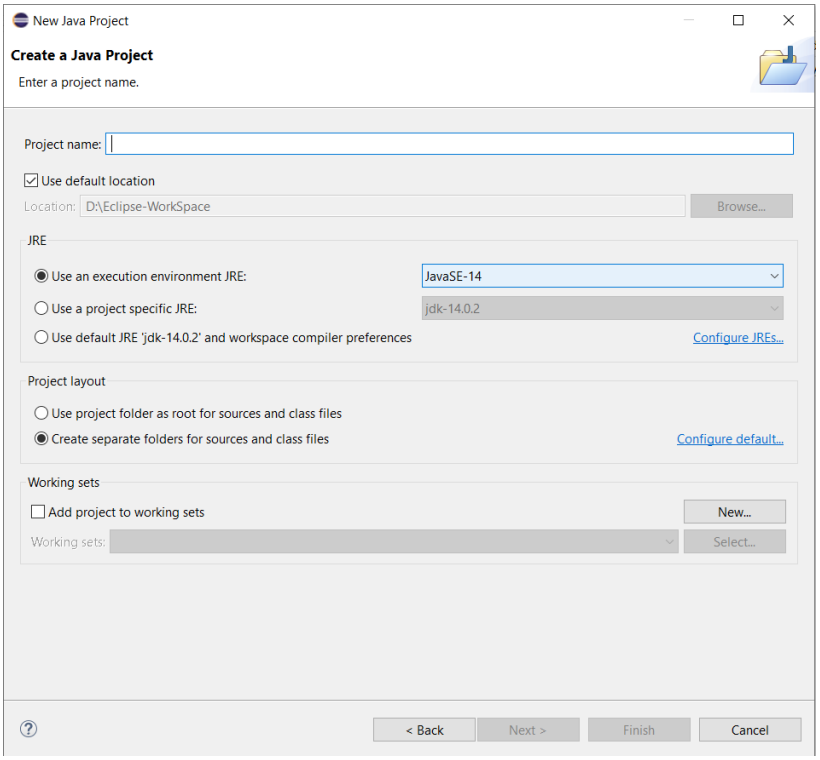
10. **One external requirement of Cloudsim** i.e. common jar package of math-related functions is to be downloaded from the Apache website or you may directly download by clicking here.
11. Unzip Eclipse, Cloudsim and Common Math libraries to some common folder.
12. First of all, navigate to the folder where you have unzipped the eclipse folder and open Eclipse.exe



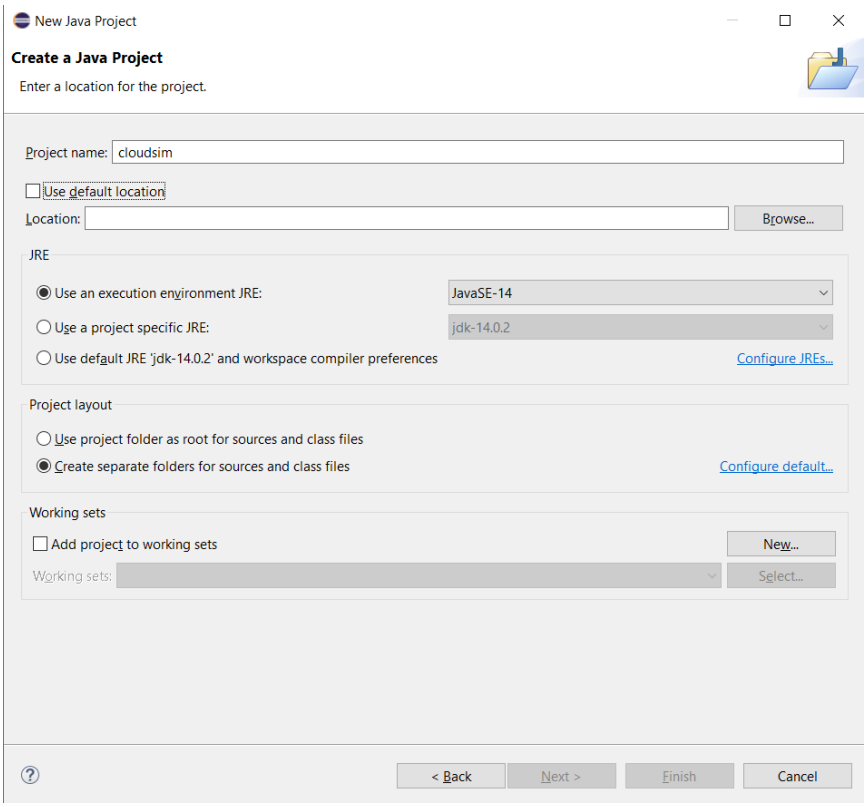
13. Now in Eclipse navigate the menu: File -> New -> Project, to open the new project wizard



14. 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'.

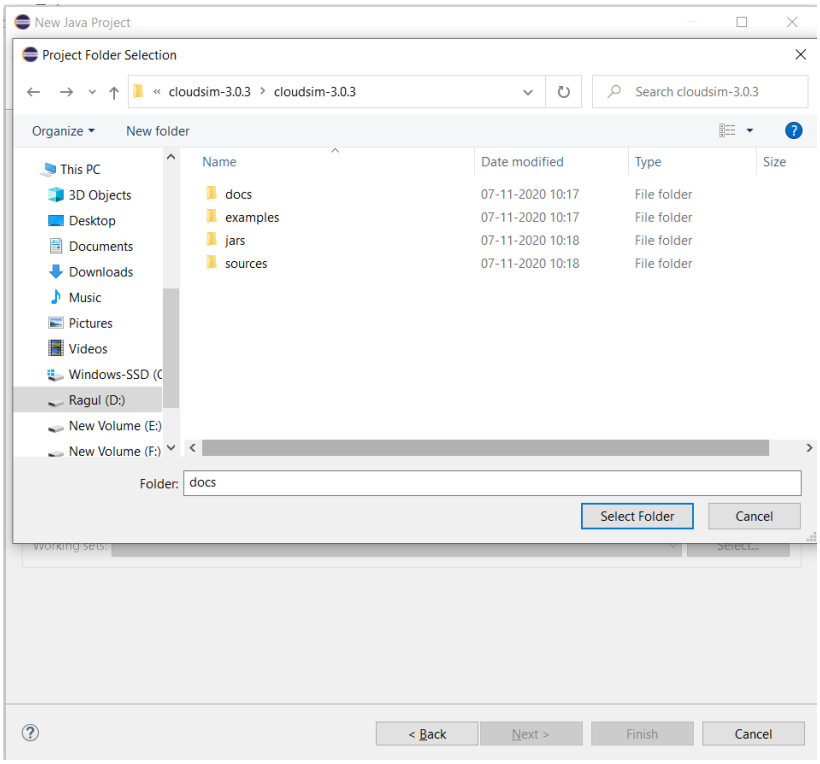


15. Now a detailed new project window will open, here you will provide the project name and the path of CloudSim project source code, which will be done as follows:
    - Project Name: CloudSim.

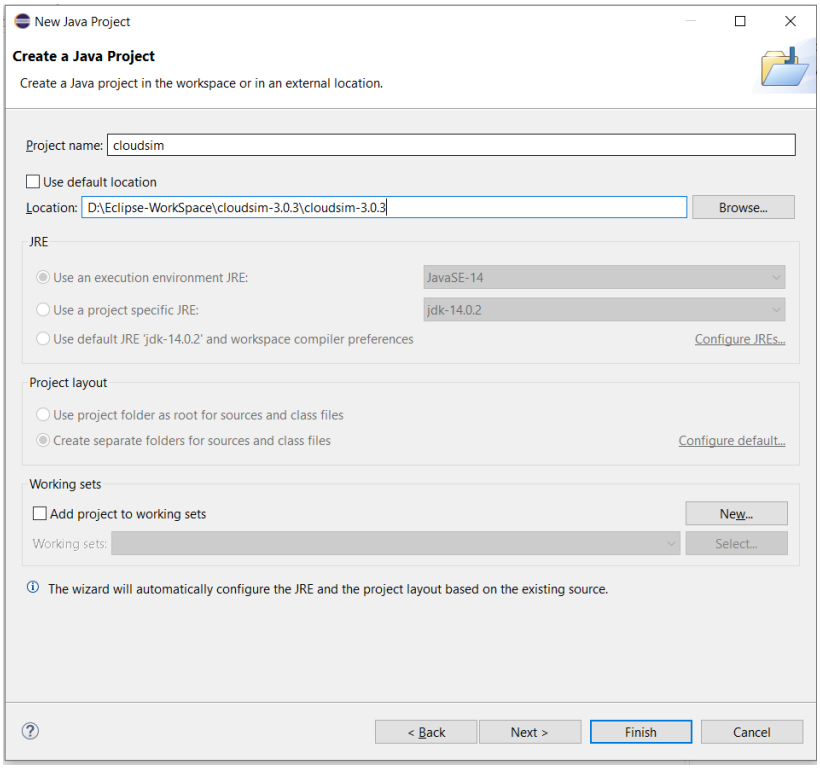


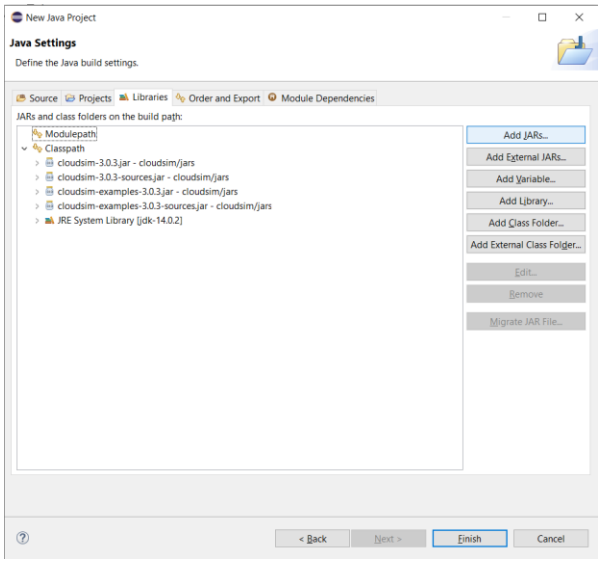
- Unselect the ‘Use default location’ option and then click on ‘Browse’ to open the path where you have unzipped the Cloudsim project and finally click Next to set project settings.

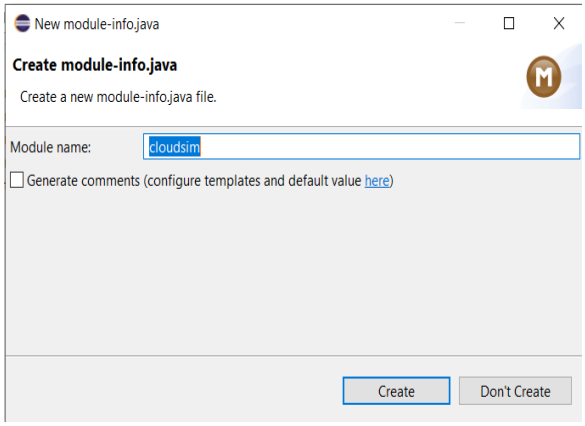
16. Make sure you navigate the path till you can see the bin, docs, examples etc folder in the navigation plane.



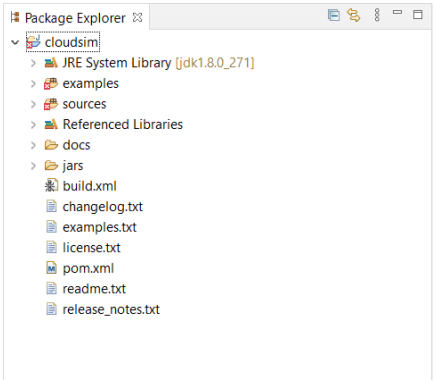
17. Once done, finally, click ‘Next’ to go to the next step i.e. setting up of project settings



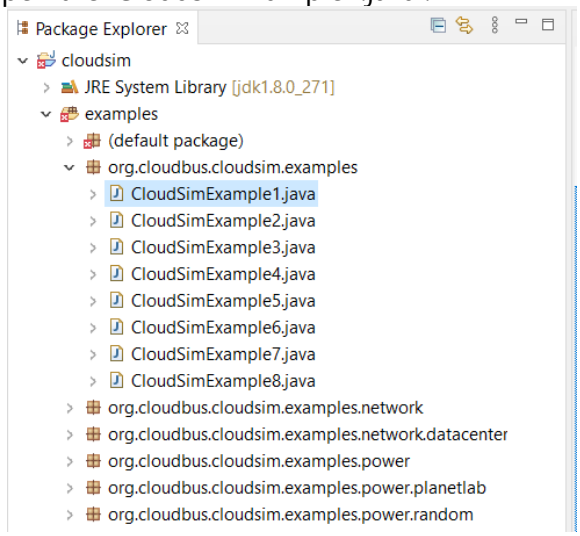




18. Once the project is configured you can open the 'Project Explorer' and start exploring the Cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured Cloudsim project, which may take some time depending on the configuration of the computer system.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



20. 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'.



PROJECT INITIALIZATION:

1. The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows:

   CloudSim.init(num_user, calendar, trace_flag)

2. Data centres are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or spaceshared, the time zone and its price:

   Datacenter datacenter = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList), storageList, 0);

3. The third step is to create a broker:

   DatacenterBroker broker = createBroker();

4. The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM's owner, mips, number Of Pes amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:

   Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())

5. Submit the VM list to the broker:

   broker.submitVmList(vmlist)

6. Create a cloudlet with length, file size, output size, and utilisation model:

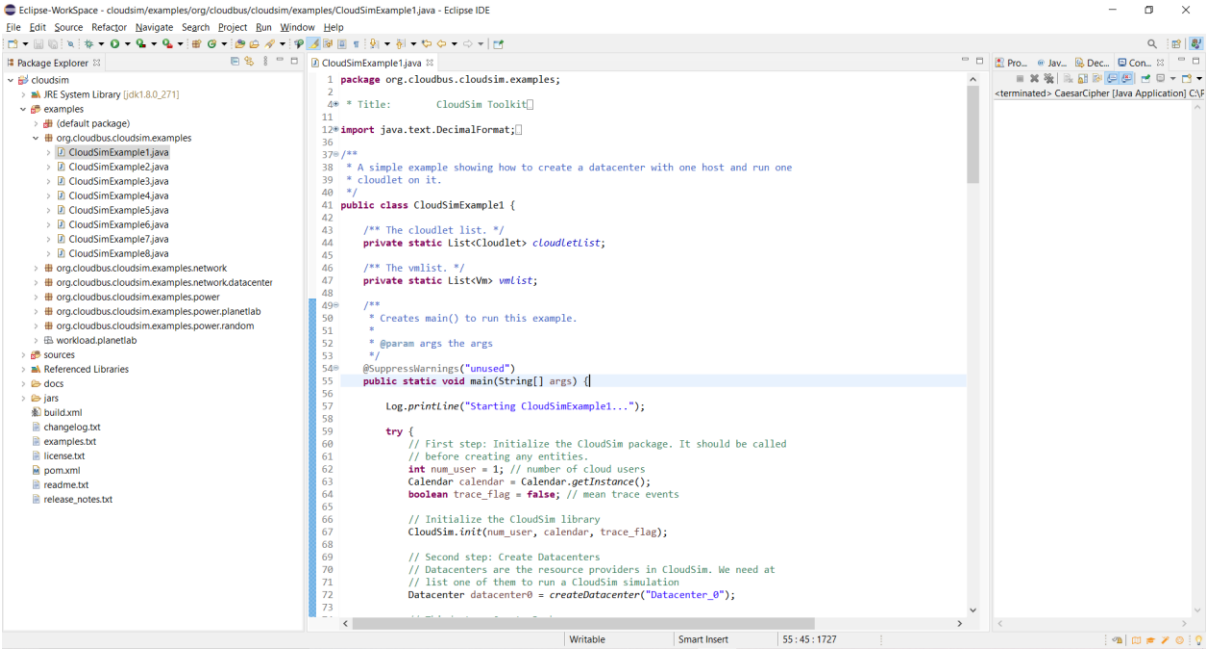
   Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);

7. Submit the cloudlet list to the broker:

   broker.submitCloudletList(cloudletList)

8. Start the simulation:

   CloudSim.startSimulation()



9. 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.

```
Problems  Javadoc  Declaration  Console
<terminated> CloudSimExample1 [Java Application] C:\Program Files\Java\jdk1.8.0_271\bin\javaw.exe  (7 Nov, 2020 10:56:22 AM – 10:56:26 AM)
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

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

10. Now we can run a Shortest Job First(SJF) scheduling algorithm in the cloudsim project.
11. On initializing the scheduling algorithm the program will fetch the output as follows:

```
Starting CloudSimExample6...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
0.0: Broker: Trying to Create VM #9 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #6 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #6 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #7 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #8 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #9 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #9 to Host #1 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #3 has been created in Datacenter #2, Host #1
0.1: Broker: VM #4 has been created in Datacenter #2, Host #0
0.1: Broker: VM #5 has been created in Datacenter #2, Host #1
0.1: Broker: Creation of VM #6 failed in Datacenter #2
0.1: Broker: Creation of VM #7 failed in Datacenter #2
0.1: Broker: Creation of VM #8 failed in Datacenter #2
0.1: Broker: Creation of VM #9 failed in Datacenter #2
0.1: Broker: Trying to Create VM #6 in Datacenter_1
0.1: Broker: Trying to Create VM #7 in Datacenter_1
0.1: Broker: Trying to Create VM #8 in Datacenter_1
0.1: Broker: Trying to Create VM #9 in Datacenter_1
0.2: Broker: VM #6 has been created in Datacenter #3, Host #0
0.2: Broker: VM #7 has been created in Datacenter #3, Host #0
```

```
0.1: Broker: Trying to Create VM #9 in Datacenter_1
0.2: Broker: VM #6 has been created in Datacenter #3, Host #0
0.2: Broker: VM #7 has been created in Datacenter #3, Host #0
0.2: Broker: VM #8 has been created in Datacenter #3, Host #0
0.2: Broker: VM #9 has been created in Datacenter #3, Host #1
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
0.2: Broker: Sending cloudlet 2 to VM #2
0.2: Broker: Sending cloudlet 3 to VM #3
0.2: Broker: Sending cloudlet 4 to VM #4
0.2: Broker: Sending cloudlet 5 to VM #5
0.2: Broker: Sending cloudlet 6 to VM #6
0.2: Broker: Sending cloudlet 7 to VM #7
0.2: Broker: Sending cloudlet 8 to VM #8
0.2: Broker: Sending cloudlet 9 to VM #9
0.2: Broker: Sending cloudlet 10 to VM #0
0.2: Broker: Sending cloudlet 11 to VM #1
0.2: Broker: Sending cloudlet 12 to VM #2
0.2: Broker: Sending cloudlet 13 to VM #3
0.2: Broker: Sending cloudlet 14 to VM #4
0.2: Broker: Sending cloudlet 15 to VM #5
0.2: Broker: Sending cloudlet 16 to VM #6
0.2: Broker: Sending cloudlet 17 to VM #7
0.2: Broker: Sending cloudlet 18 to VM #8
0.2: Broker: Sending cloudlet 19 to VM #9
0.2: Broker: Sending cloudlet 20 to VM #0
0.2: Broker: Sending cloudlet 21 to VM #1
0.2: Broker: Sending cloudlet 22 to VM #2
0.2: Broker: Sending cloudlet 23 to VM #3
0.2: Broker: Sending cloudlet 24 to VM #4
0.2: Broker: Sending cloudlet 25 to VM #5
0.2: Broker: Sending cloudlet 26 to VM #6
0.2: Broker: Sending cloudlet 27 to VM #7
0.2: Broker: Sending cloudlet 28 to VM #8
0.2: Broker: Sending cloudlet 29 to VM #9
0.2: Broker: Sending cloudlet 30 to VM #0
0.2: Broker: Sending cloudlet 31 to VM #1
0.2: Broker: Sending cloudlet 32 to VM #2
0.2: Broker: Sending cloudlet 33 to VM #3
0.2: Broker: Sending cloudlet 34 to VM #4
0.2: Broker: Sending cloudlet 35 to VM #5
0.2: Broker: Sending cloudlet 36 to VM #6
0.2: Broker: Sending cloudlet 37 to VM #7
0.2: Broker: Sending cloudlet 38 to VM #8
0.2: Broker: Sending cloudlet 39 to VM #9
1.22: Broker: Cloudlet 2 received
1.65: Broker: Cloudlet 4 received
1.762: Broker: Cloudlet 5 received
1.788: Broker: Cloudlet 7 received
1.901: Broker: Cloudlet 8 received
2.134: Broker: Cloudlet 3 received
2.256: Broker: Cloudlet 1 received
2.423: Broker: Cloudlet 6 received
2.782: Broker: Cloudlet 9 received
3.1289999999999996: Broker: Cloudlet 0 received
3.197: Broker: Cloudlet 17 received
3.2389999999999994: Broker: Cloudlet 14 received
3.616: Broker: Cloudlet 16 received
3.726: Broker: Cloudlet 18 received
3.8229999999999995: Broker: Cloudlet 15 received
4.045999999999999: Broker: Cloudlet 11 received
4.155999999999999: Broker: Cloudlet 12 received
4.405999999999999: Broker: Cloudlet 24 received
4.749: Broker: Cloudlet 28 received
4.879999999999999: Broker: Cloudlet 10 received
5.036: Broker: Cloudlet 27 received
5.082999999999999: Broker: Cloudlet 13 received
5.689: Broker: Cloudlet 19 received
5.728999999999999: Broker: Cloudlet 25 received
5.882: Broker: Cloudlet 26 received
6.110999999999999: Broker: Cloudlet 22 received
6.699999999999999: Broker: Cloudlet 21 received
6.827: Broker: Cloudlet 38 received
7.153: Broker: Cloudlet 36 received
7.185: Broker: Cloudlet 20 received
7.294999999999999: Broker: Cloudlet 34 received
7.482999999999999: Broker: Cloudlet 32 received
7.592999999999998: Broker: Cloudlet 23 received
7.797: Broker: Cloudlet 37 received
8.322: Broker: Cloudlet 29 received
8.567999999999998: Broker: Cloudlet 35 received
8.677999999999997: Broker: Cloudlet 31 received
8.952999999999998: Broker: Cloudlet 33 received
9.713999999999999: Broker: Cloudlet 39 received
8.567999999999998: Broker: Cloudlet 35 received
8.677999999999997: Broker: Cloudlet 31 received
8.952999999999998: Broker: Cloudlet 33 received
9.713999999999999: Broker: Cloudlet 39 received
9.826999999999998: Broker: Cloudlet 30 received
9.826999999999998: Broker: All Cloudlets executed. Finishing...
9.826999999999998: Broker: Destroying VM #0
9.826999999999998: Broker: Destroying VM #1
9.826999999999998: Broker: Destroying VM #2
9.826999999999998: Broker: Destroying VM #3
9.826999999999998: Broker: Destroying VM #4
9.826999999999998: Broker: Destroying VM #5
9.826999999999998: Broker: Destroying VM #6
9.826999999999998: Broker: Destroying VM #7
9.826999999999998: Broker: Destroying VM #8
9.826999999999998: Broker: Destroying VM #9
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
```

12. The output of the scheduling algorithm is as follows:

```
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time    user id
    2        SUCCESS        2            2           1.02        0.2           1.22    4
    4        SUCCESS        2            4           1.45        0.2           1.65    4
    5        SUCCESS        2            5           1.56        0.2           1.76    4
    7        SUCCESS        3            7           1.59        0.2           1.79    4
    8        SUCCESS        3            8           1.7        0.2           1.9    4
    3        SUCCESS        2            3           1.93        0.2           2.13    4
    1        SUCCESS        2            1           2.06        0.2           2.26    4
    6        SUCCESS        3            6           2.22        0.2           2.42    4
    9        SUCCESS        3            9           2.58        0.2           2.78    4
    0        SUCCESS        2            0           2.93        0.2           3.13    4
    17        SUCCESS        3            7           1.41        1.79           3.2    4
    14        SUCCESS        2            4           1.59        1.65           3.24    4
    16        SUCCESS        3            6           1.19        2.42           3.62    4
    18        SUCCESS        3            8           1.82        1.9           3.73    4
    15        SUCCESS        2            5           2.06        1.76           3.82    4
    11        SUCCESS        2            1           1.79        2.26           4.05    4
    12        SUCCESS        2            2           2.94        1.22           4.16    4
    24        SUCCESS        2            4           1.17        3.24           4.41    4
    28        SUCCESS        3            8           1.02        3.73           4.75    4
    10        SUCCESS        2            0           1.75        3.13           4.88    4
    27        SUCCESS        3            7           1.84        3.2           5.04    4
    13        SUCCESS        2            3           2.95        2.13           5.08    4
    19        SUCCESS        3            9           2.91        2.78           5.69    4
    25        SUCCESS        2            5           1.91        3.82           5.73    4
    26        SUCCESS        3            6           2.27        3.62           5.88    4
    22        SUCCESS        2            2           1.96        4.16           6.11    4
    21        SUCCESS        2            1           2.65        4.05           6.7    4
    38        SUCCESS        3            8           2.08        4.75           6.83    4
    36        SUCCESS        3            6           1.27        5.88           7.15    4
    20        SUCCESS        2            0           2.31        4.88           7.18    4
    34        SUCCESS        2            4           2.89        4.41           7.29    4
    32        SUCCESS        2            2           1.37        6.11           7.48    4
    23        SUCCESS        2            3           2.51        5.08           7.59    4
    37        SUCCESS        3            7           2.76        5.04           7.8    4
    29        SUCCESS        3            9           2.63        5.69           8.32    4
    35        SUCCESS        2            5           2.84        5.73           8.57    4
    31        SUCCESS        2            1           1.98        6.7           8.68    4
    33        SUCCESS        2            3           1.36        7.59           8.95    4
    39        SUCCESS        3            9           1.39        8.32           9.71    4
    30        SUCCESS        2            0           2.64        7.18           9.83    4
CloudSimExample6 finished!
```

RESULT:

Thus, the Simulation of a cloud scenario using CloudSim was successfully executed and running a scheduling algorithm (Shortest Job First) that is not present in CloudSim was done successfully and output was obtained.

EX. No:6 Find a procedure to transfer the files from one virtual machine to another virtual machine.

Aim:

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

Procedure:

I. A virtual machine is a software environment that emulates the hardware required to install an operating system (OS). In very general terms, this lets you install an operating system on an existing OS much like an app.

II. The different methods available for us to transfer the files from one virtual machine to another virtual machine are as follows:

- Copy and paste / Drag and Drop
- USB drive
- Network share

III. Clearly, each option is best for a specific type of data. For example, copy and paste is best for sharing text and small files, such as copying code from a browser on your host PC into a terminal session in the guest OS.

IV. Each virtual machine has its own operative system running on and acts as a physical machine.

V. Each virtual machine is an instance of a program owned by an user in the hosting operative system and should undergo the restrictions of the user in the hosting OS.

VI. E.g Let we say that Hastur and Meow are users of the hosting machine, but they did not allow each other to see their directories (no read/write/execute authorization).

VII. When each of them run a virtual machine, for the hosting OS those virtual machine are two normal programs owned by Hastur and Meow and cannot see the private directory of the other user.

VIII. This is a restriction due to the hosting OS. It's easy to overcame it: it's enough to give authorization to read/write/execute to a directory or to chose a different directory in which both users can read/write/execute.

1. Copy and Paste Data / Drag and Drop:

- The simplest option is to copy the data from your host PC and paste it into the window of the guest VM. Or copy from the guest VM and into an open file browser on the host.

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

- For VMware users, you'll need to first install the VMware Tools package, which brings additional features. You can commence this via **VM > Install VMware Tools**. If you haven't already downloaded VMware Tools, instructions for doing so will be given.

- You can then enable copy and paste in **VM > Settings > Options**. Select **Guest Isolation**, then **Enable copy and paste** and confirm with **OK**.

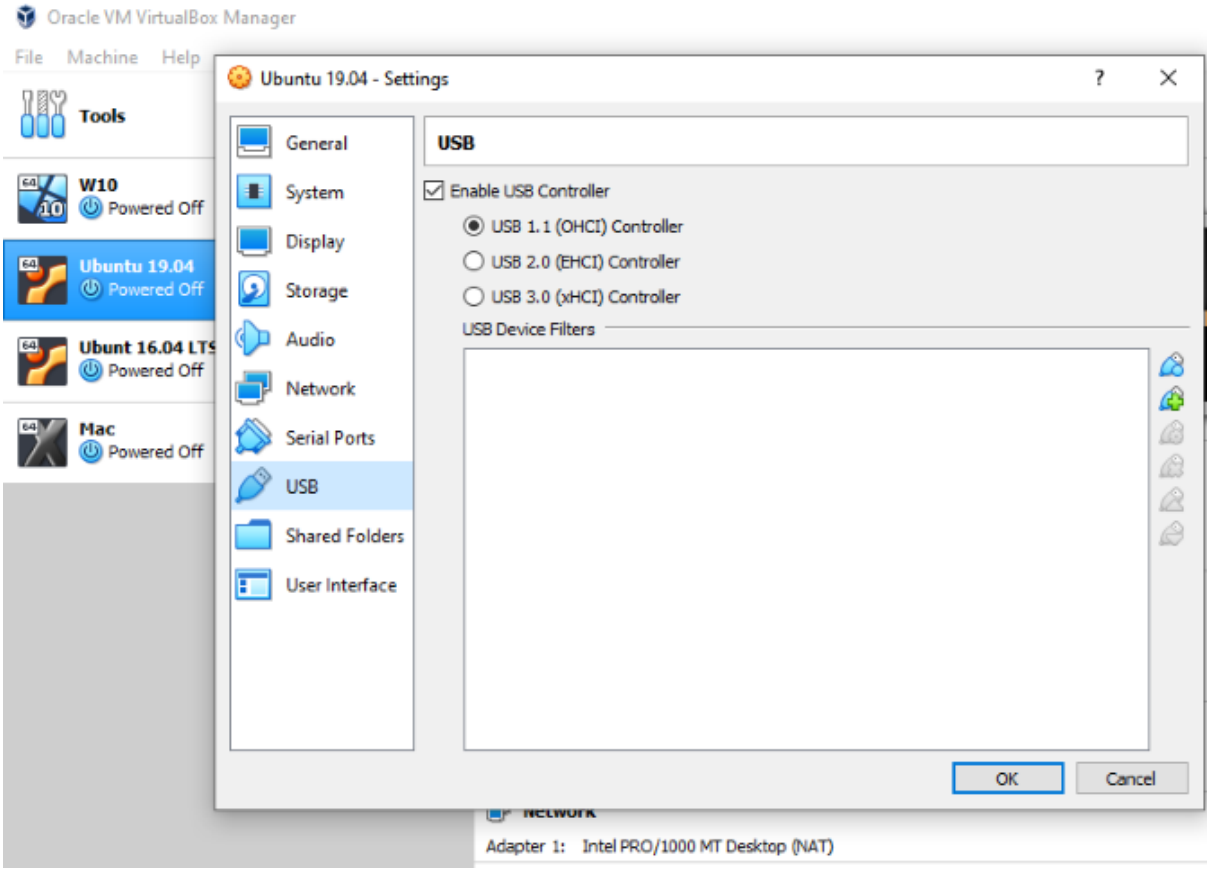
- Sharing data between the guest and host operating systems in this way is best suited for smaller files. You might also share text strings, URLs, that sort of thing. Steer clear of large files though---you have other options for those.

2.Transfer files via USB Drive:

- Using a USB stick to transfer data between two physical machines is a time-honored tradition.

- To be able to access USB devices from within VirtualBox, you'll need to enable USB access. For this, the VirtualBox Extension Pack is required from www.virtualbox.org/wiki/Downloads.

- Once you've done that, insert the USB device you wish to use. Next, open VirtualBox and click **File > Preferences**, then **Extensions** and click **+**. Browse to the downloaded Extension Pack, click **Open**, then when prompted, **Install**. Follow the prompts to complete the process. You can then check to confirm USB is enabled in **Settings > USB.**



- With the USB support added, you'll need to enable it. In the main VirtualBox window, right click the VM you want to use and select **Settings > USB**. Click **+** then browse for the USB device. It will be available when you launch the VM. Additional drives can be added in the same way.

- With VMware, when a USB device is connected and the VM is the active window, the device is detected. However, it will not be detected by the host PC in this scenario. For this to happen, remove the drive, minimize the VM, then reconnect.

- It's simple but can get messy if you forget which operating system the USB stick is connected to.
- This option is best for large files. Of course, you're limited by the capacity of the USB device, so keep that in mind. Whatever VM software you use, safe ejection of USB devices is recommended on both host and guest virtual machines.

3. Transfer Files via a common shared network drive:

- Your third option is to set up a network share on your host PC that the guest VM can access. This means designating a portion of your PC's hard disk drive as accessible over the local network. With this set up, the VM can then connect to the network and access the drive.

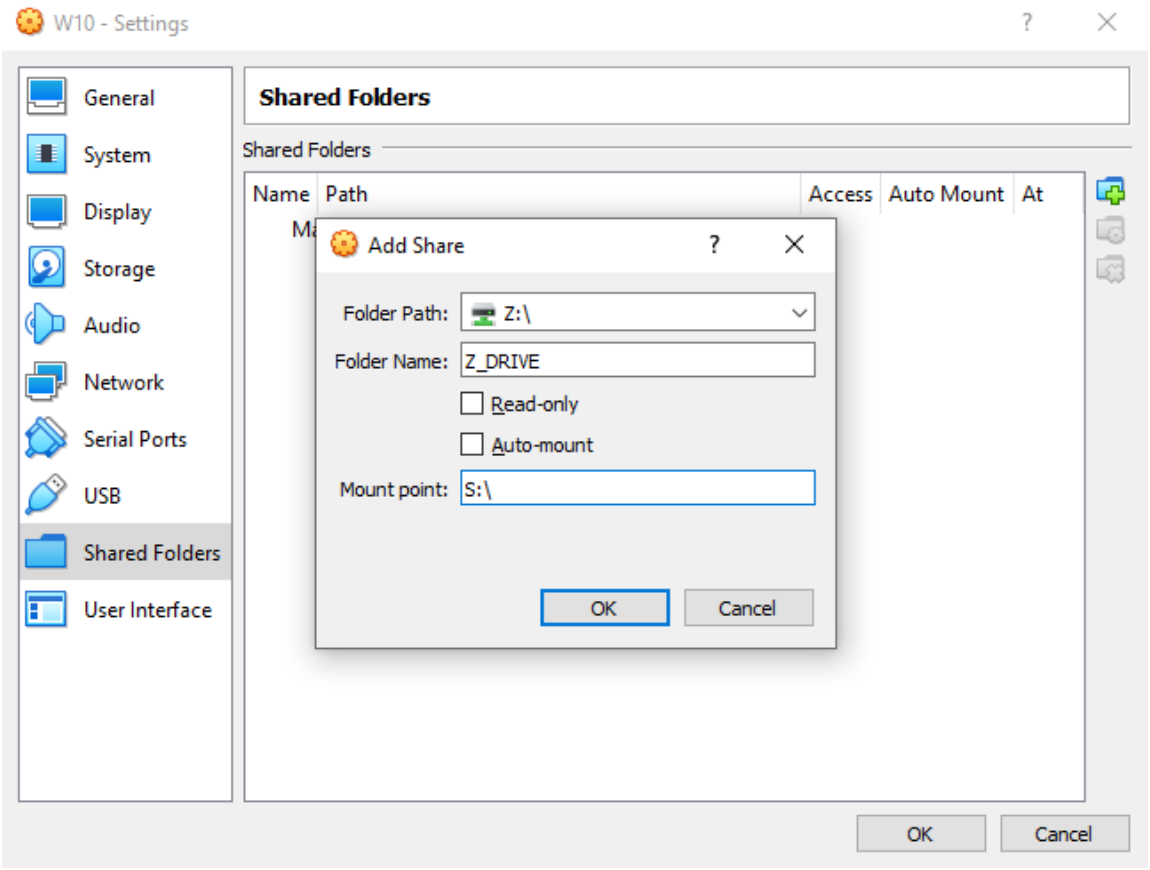
- Although physically all on the same computer, this adds higher capacity to your virtual machine data sharing.

- You should have already downloaded VirtualBox Guest Additions. This should be installed via **Devices > Install Guest Additions**, where you should browse for the appropriate EXE file. Follow the steps to the end, choosing the default options, then **Finish**.

- Launch VirtualBox and open **Devices > Shared Folders > Shared Folders Settings**. Click +, then in **Folder Path** click the arrow and select **Other**. Browse (the host OS) for the folder you're using as a share, highlight it, then **Select Folder**.

- In the Add Share window, give the share a name (keeping the same name in the guest OS as in the host OS is wise). Check **Auto-mount** and **Make permanent**, then **OK**.

- From the guest OS you'll find the share set up in the usual location for network shares. For example in Windows 10, this will be under Network Locations in Windows Explorer.



- This is how we can share via the common shared network drive.

4. Another Method to Transfer Files:

- You can use **usual method to copy files between 2 different computer** with client-server application. (e.g. scp with sshd active for linux, winscp... you can get some info about SSH servers e.g. here)

- You need an active server (sshd) on the receiving machine and a client on the sending machine. Of course you need to have the authorization setted (via password or, better, via an automatic authentication method).

- **Note:** many Linux/Ubuntu distribution install sshd by default: you can see if it is running with pgrep sshd from a shell. You can install with sudo apt-get install openssh-server.

- When you will need to be fast with Linux you will feel the need of ssh-keygen and to Generate once SSH Keys to copy files on/from a remote machine without writing password anymore. In this way it functions bash auto-completion remotely too!

RESULT:

Thus, the process of finding a procedure to transfer the files from one virtual machine to another virtual machine was successfully explained.

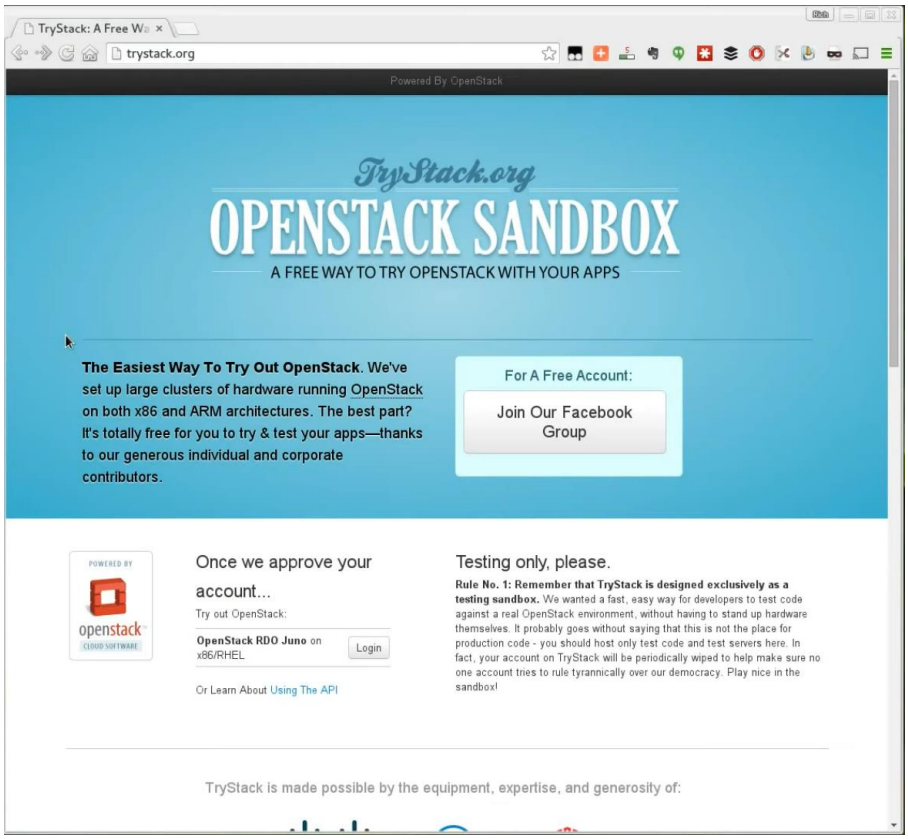
EX. No:7 Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

Aim:

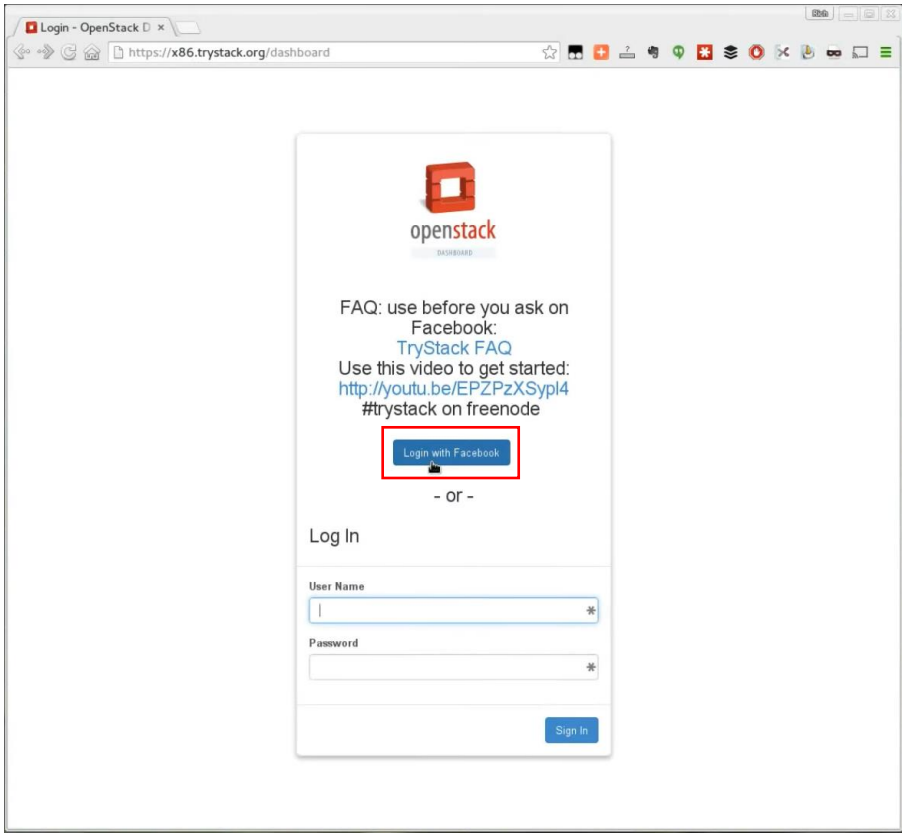
To find a procedure to launch virtual machine using trystack (Online Openstack Demo Version).

Procedure:

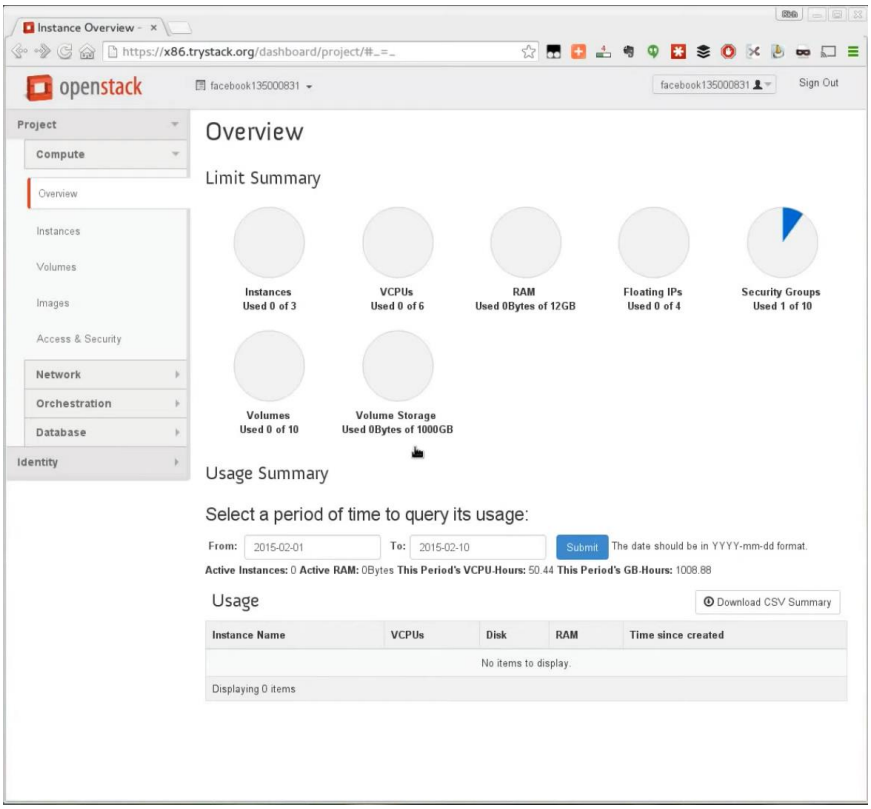
1. **OpenStack** is an open-source software cloud computing platform. OpenStack is primarily used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS).
2. In other words, you can *make your own AWS* by using OpenStack. If you want to try out OpenStack, **TryStack** is the easiest and free way to do it.
3. In order to try OpenStack in TryStack, you must register yourself by joining TryStack Facebook Group. The acceptance of group needs a couple days because it's approved manually. After you have been accepted in the TryStack Group, you can log in TryStack.



4. Login using Facebook in the login page.

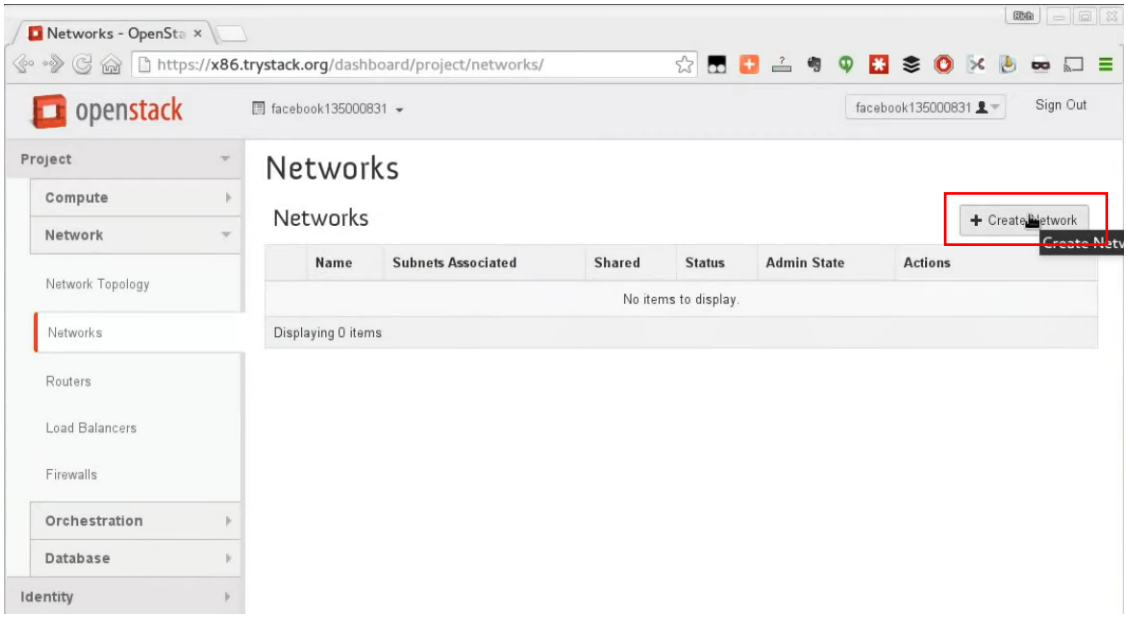


This is the overview of OpenStack Compute Dashboard.

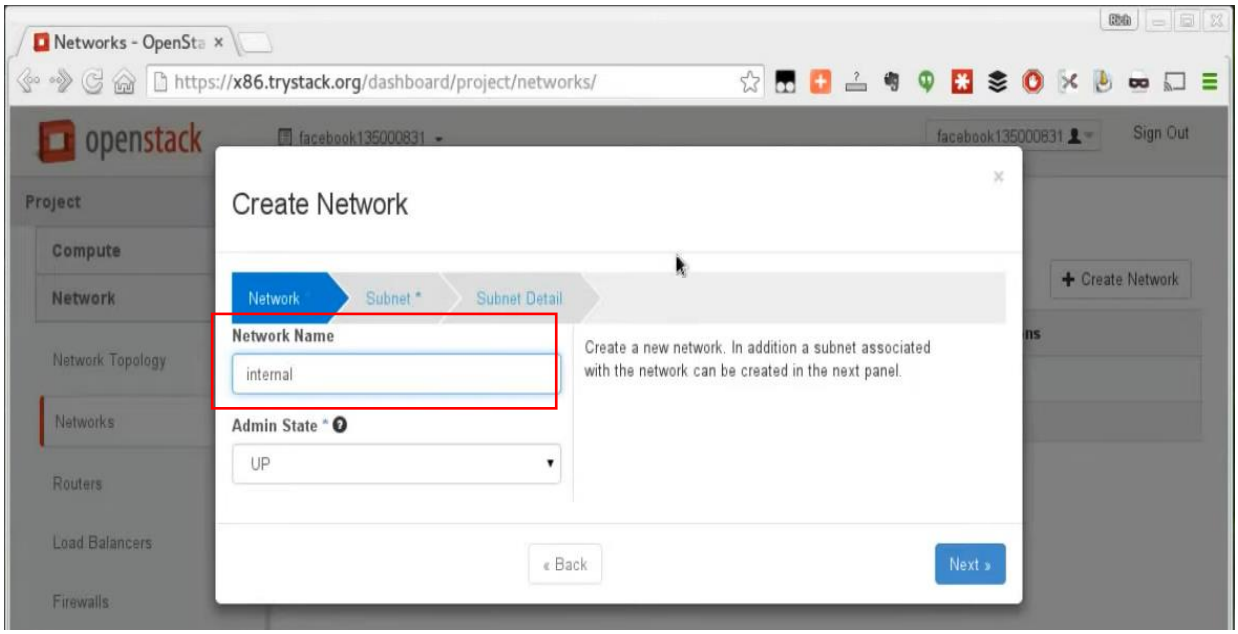


Step 1: Create Network:

1. Go to **Network > Networks** and then click **Create Network**.

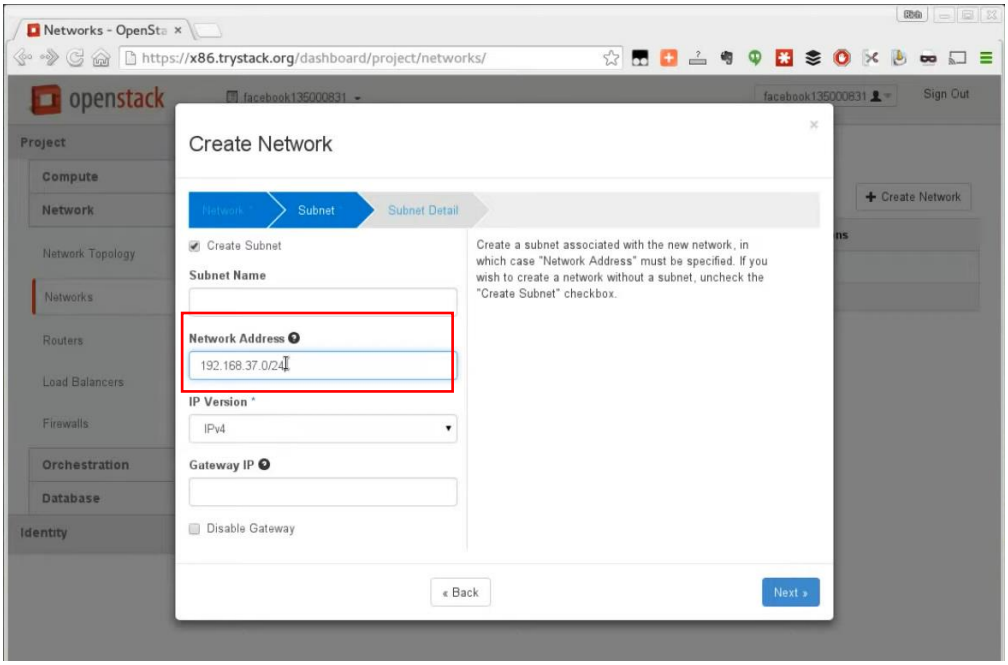


2. In **Network** tab, fill **Network Name** for example internal and then click **Next**.

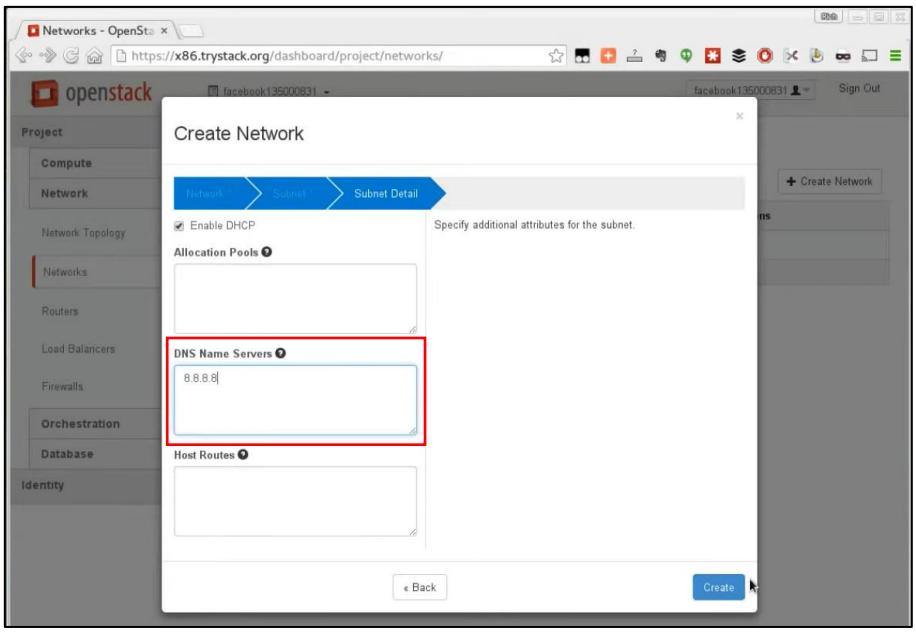


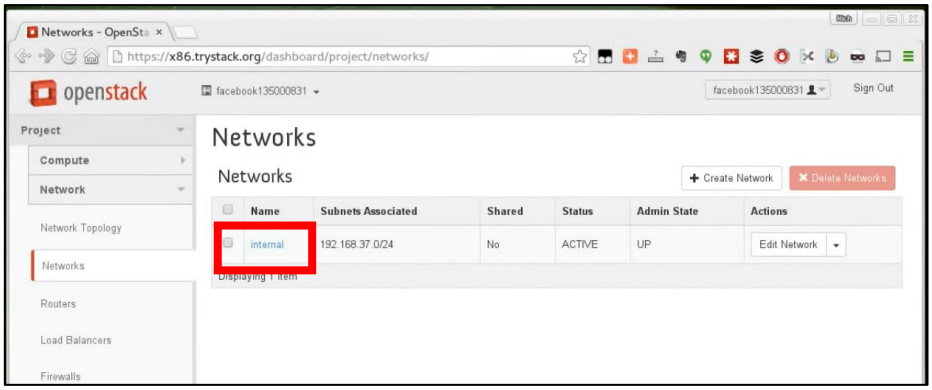
3. In Subnet tab,

- Fill **Network Address** with appropriate CIDR, for example 192.168.1.0/24. Use private network CIDR block as the best practice.
- Select **IP Version** with appropriate IP version, in this case IPv4.
- Click **Next**.



4. In **Subnet Details** tab, fill **DNS Name Servers** with 8.8.8.8 (Google DNS) and then click **Create**.

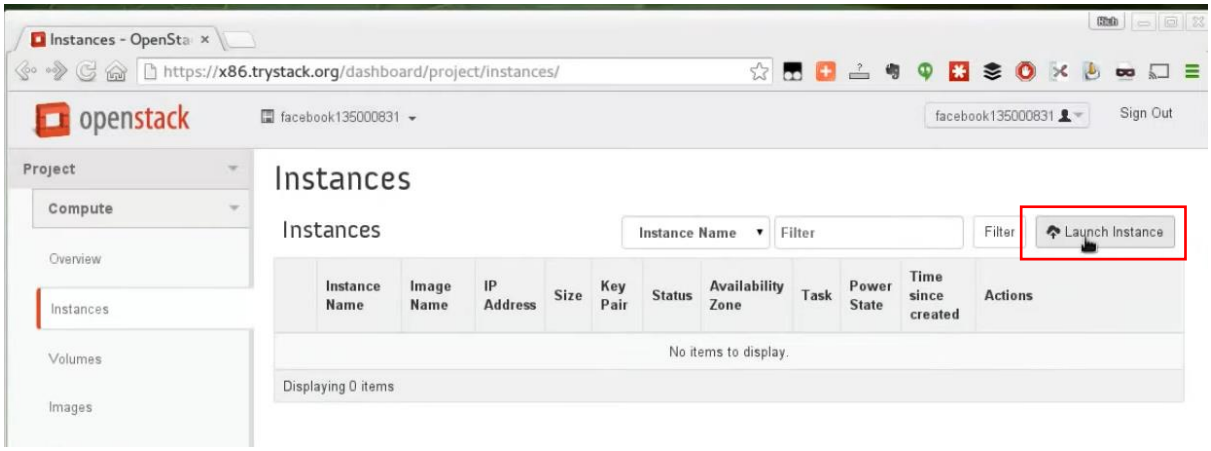




Step 2: Create Instance

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2. You need the instance to connect to the network that we just created in the previous step.

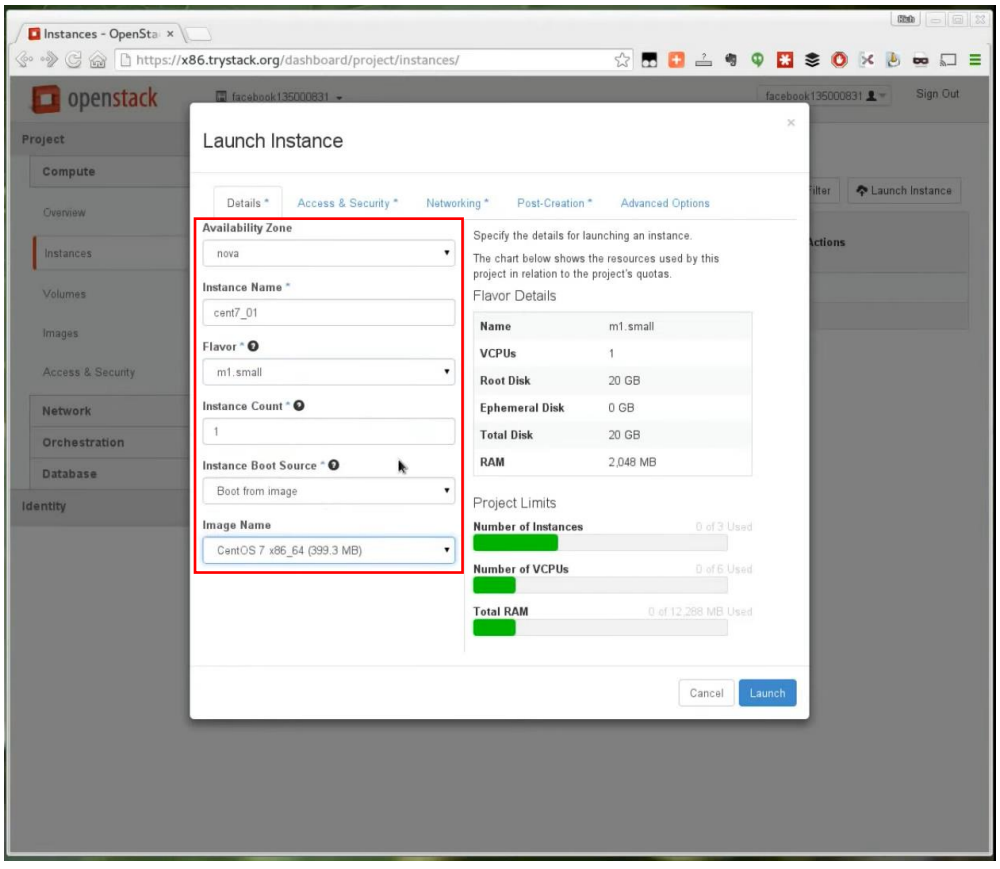
1. Go to **Compute > Instances** and then click **Launch Instance**.



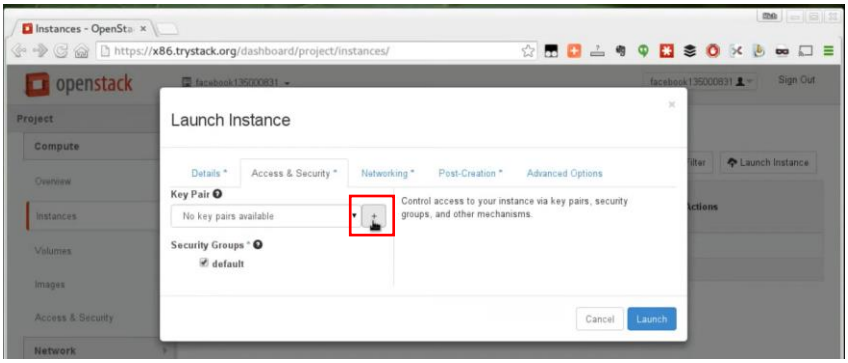
2. In **Details** tab,

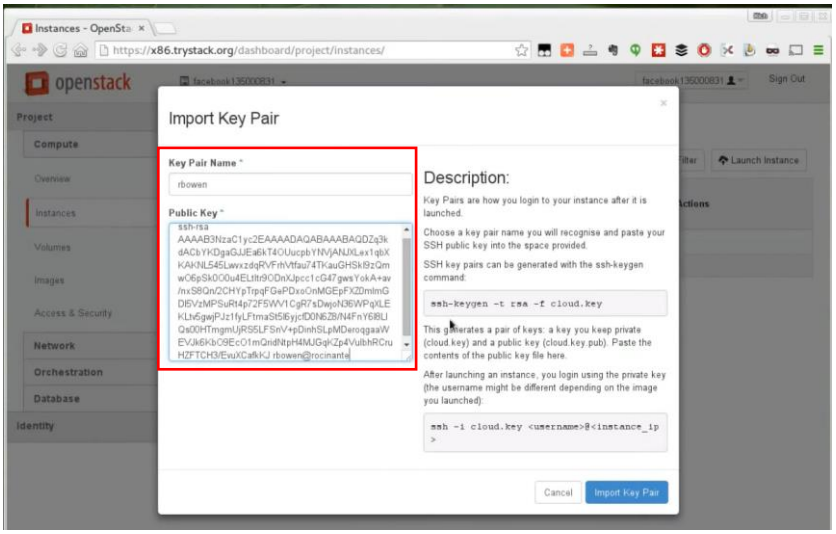
- Fill **Instance Name**, for example Ubuntu 1.
- Select **Flavor**, for example m1.medium.
- Fill **Instance Count** with **1**.
- Select **Instance Boot Source** with **Boot from Image**.

- Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want install Ubuntu 14.04 in your virtual machine.

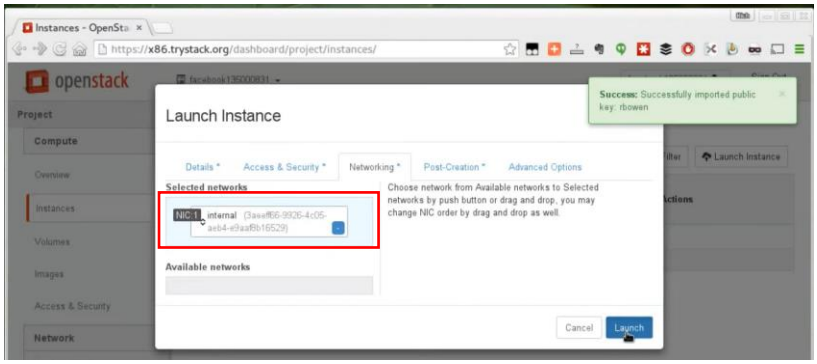


3. In **Access & Security** tab,
   - Click [+] button of **Key Pair** to import key pair. This key pair is a public and private key that we will use to connect to the instance from our machine.
   - In **Import Key Pair** dialog,
     - Fill **Key Pair Name** with your machine name (for example Edward-Key).
     - Fill **Public Key** with your **SSH public key** (usually is in ~/.ssh/id_rsa.pub). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate key pair.
     - Click **Import key pair**.
   - In **Security Groups**, mark/check **default**.

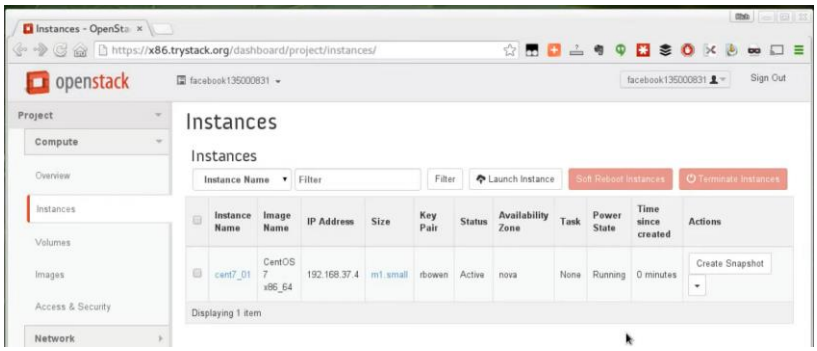




4. In **Networking** tab,
   - In **Selected Networks**, select network that have been created in Step 1, for example internal.



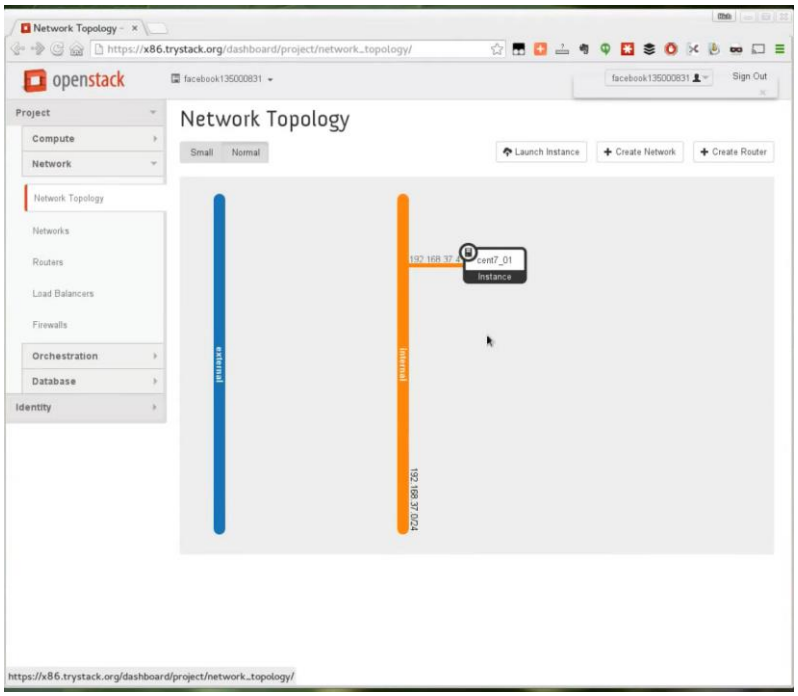
5. Click **Launch**.



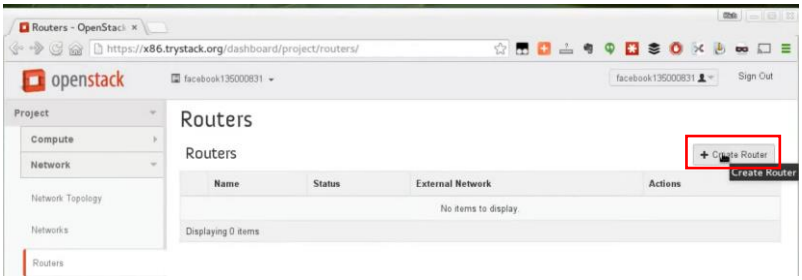
6. If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu 2.

Step 3: Create Router

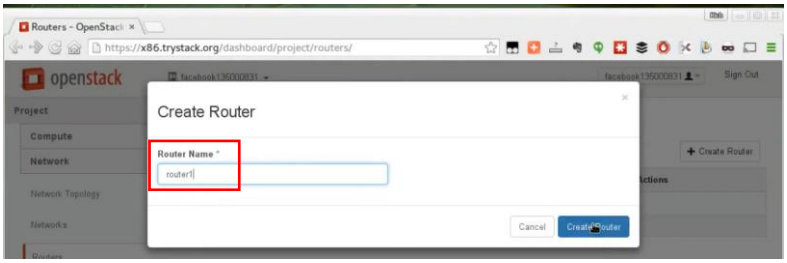
In the step 1, we created our network, but it is isolated. It doesn't connect to the internet. To make our network has an internet connection, we need a router that running as the gateway to the internet.



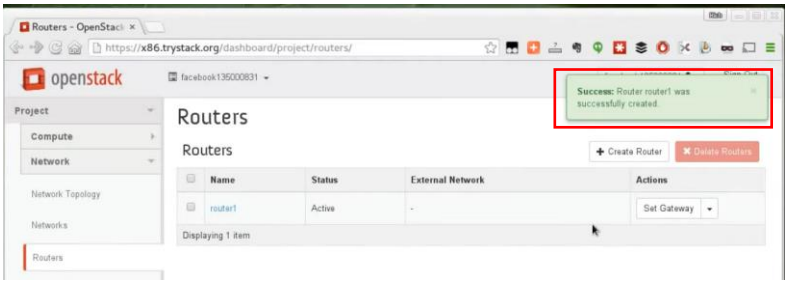
1. Go to **Network > Routers** and then click **Create Router**.



2. Fill **Router Name** for example router1 and then click **Create router**.

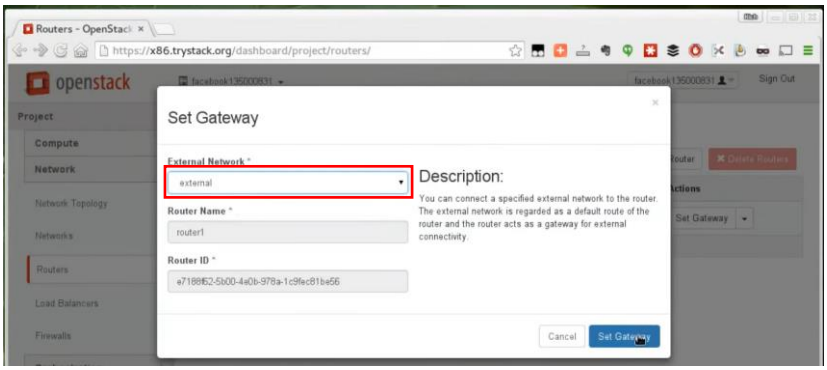


3. Click on your **router name link**, for example router1, **Router Details** page.

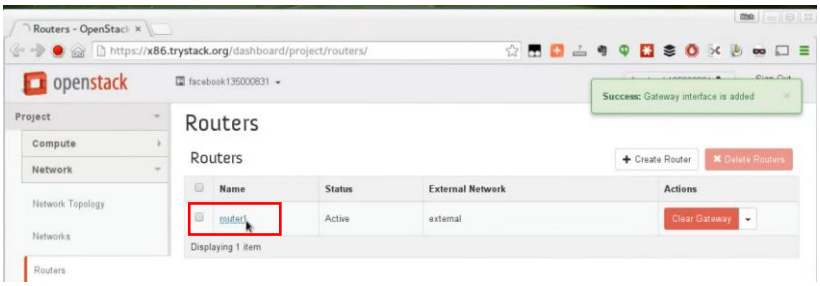


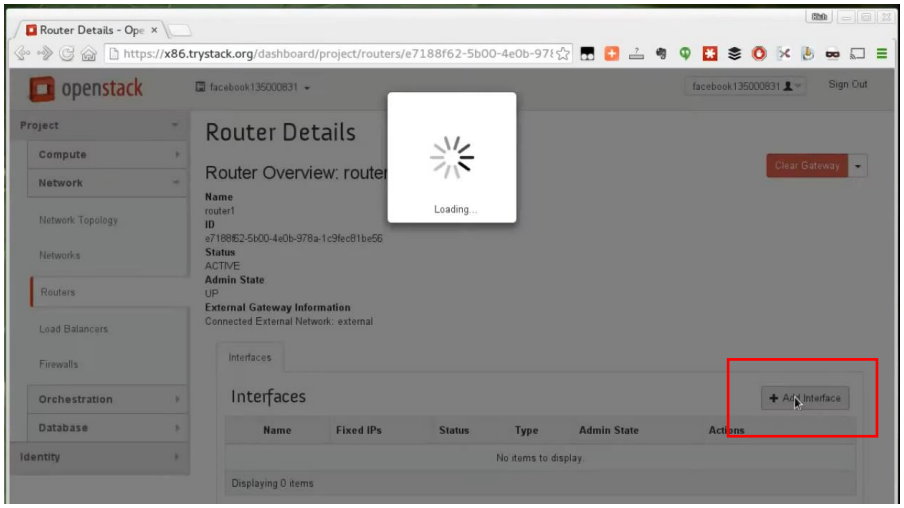
4. Click **Set** Gateway button in upper right:

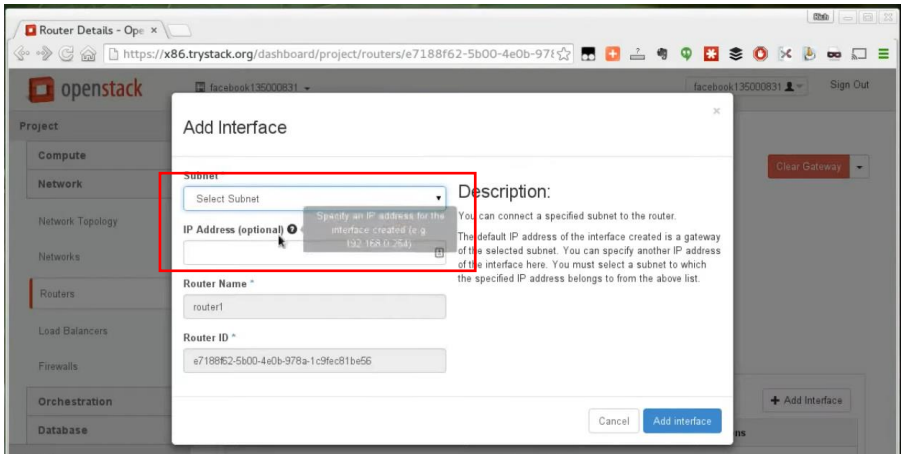
- Select **External networks** with **external**.
- Then **OK**.

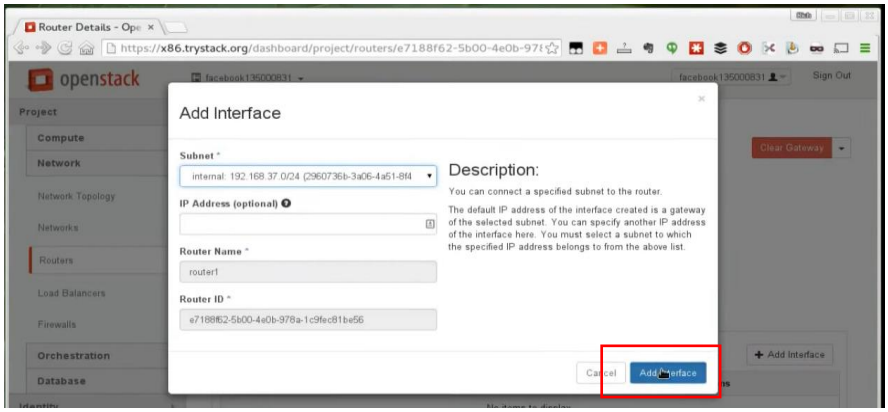


5. Click **Add Interface** button.
    - Select **Subnet** with the network that you have been created in Step 1.
    - Click **Add interface**.

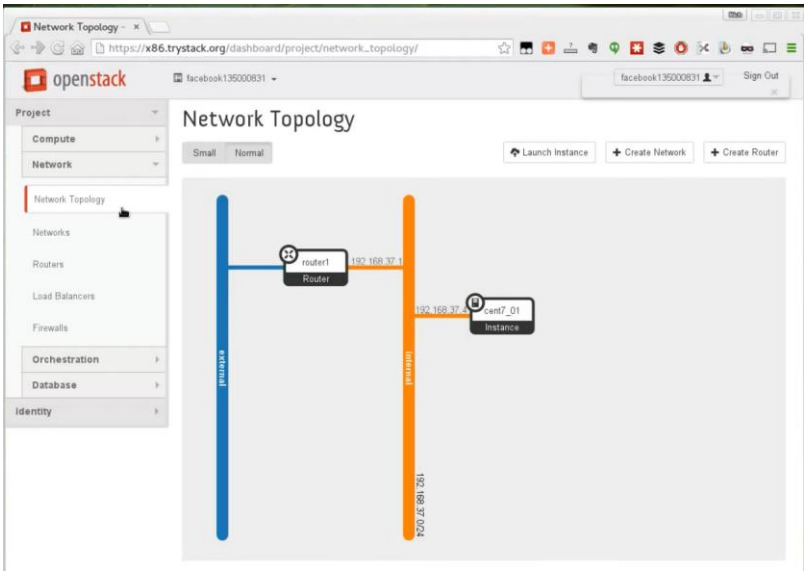








6. Go to **Network > Network Topology**. You will see the network topology. In the example, there are two network, i.e. external and internal, those are bridged by a router. There are instances those are joined to internal network.



Step 4: Configure Floating IP Address

Floating IP address is public IP address. It makes your instance is accessible from the internet. When you launch your instance, the instance will have a private network IP, but no public IP. In OpenStack, the public IPs is collected in a pool and managed by admin (in our case is TryStack). You need to request a public (floating) IP address to be assigned to your instance.

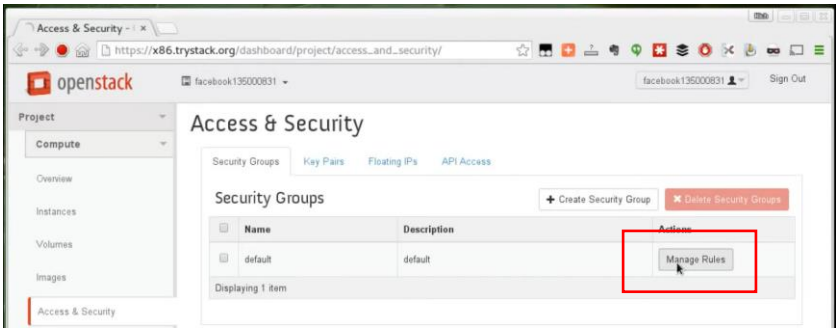
1. Go to **Compute > Instance**.
2. In one of your instances, click **More > Associate Floating IP**.
3. In **IP Address**, click Plus [**+**].
4. Select **Pool** to **external** and then click **Allocate IP**.
5. Click **Associate**.
6. Now you will get a public IP, e.g. 8.21.28.120, for your instance.

Step 5: Configure Access & Security

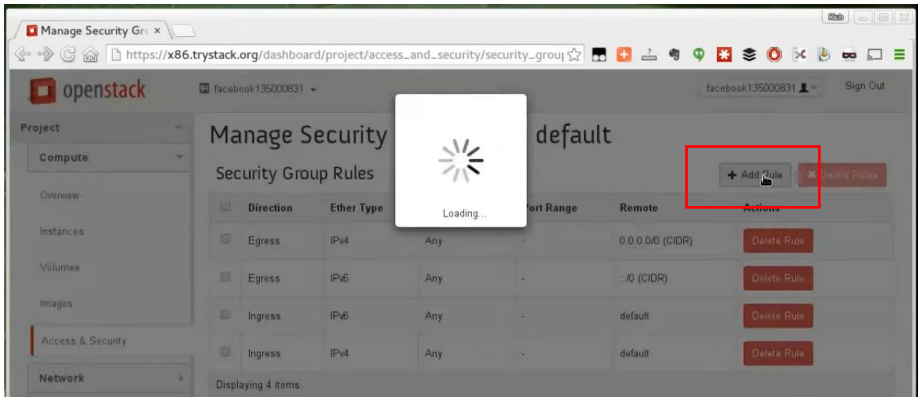
OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called *Security Group*.

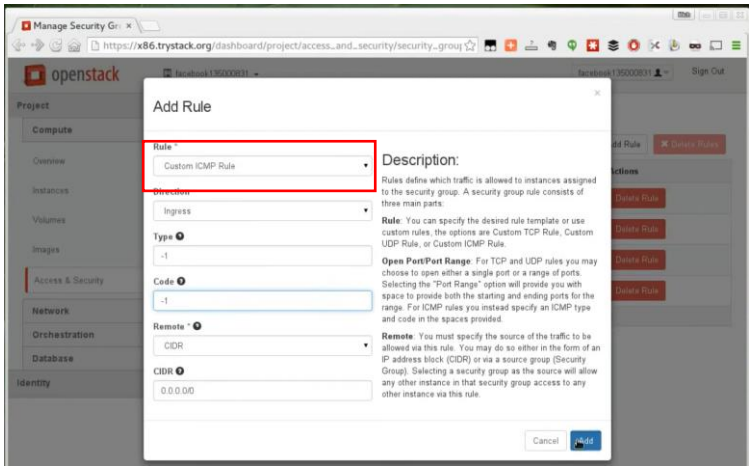
1. Go to **Compute > Access & Security** and then open **Security Groups** tab.

2. In **default** row, click **Manage Rules**.

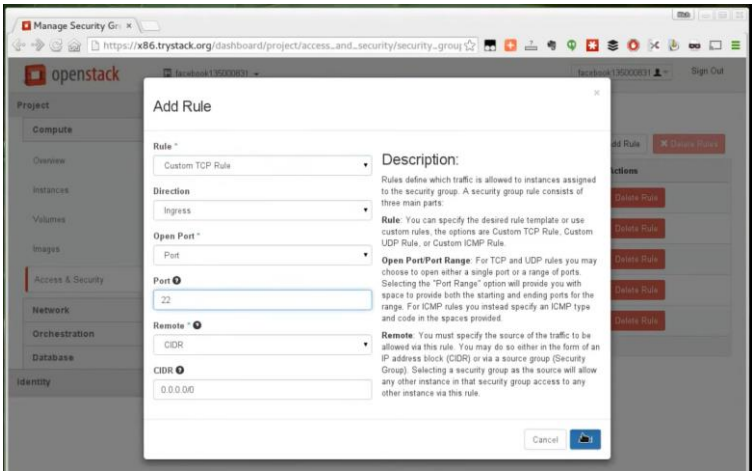


3. Click **Add Rule**, choose **ALL ICMP** rule to enable ping into your instance, and then click **Add**.





4. Click **Add Rule**, choose **HTTP** rule to open HTTP port (port 80), and then click **Add**.



5. Click **Add Rule**, choose **SSH** rule to open SSH port (port 22), and then click **Add**.

6. You can open other ports by creating new rules.

Step 6: SSH to Your Instance

- Now, you can SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be ubuntu.

At last after all the process if we ping the new instance created, it would work as expected as below:

RESULT:

Thus, to find a procedure to launch virtual machine using trystack (Online Openstack Demo Version) was successfully completed and executed.

EX. No:8 Install Hadoop single node cluster and run simple applications like wordcount.

Aim:

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

Procedure:

1. The following are the procedures to install Hadoop single node cluster in windows OS:

2. **Choose target OS version:**

The Hadoop developers have used Windows Server 2008 and Windows Server 2008 R2 during development and testing. Windows Vista and Windows 7 are also likely to work because of the Win32 API similarities with the respective server SKUs.

3. **Getting Hadoop sources:**

The current stable release as of August 2014 is 2.5. The source distribution can be retrieved from the ASF download server or using subversion or git.

- From the ASF Hadoop download page or a mirror.
- Subversion URL: _https://svn.apache.org/repos/asf/hadoop/common/branches/branch-2.5_
- Git repository URL: *git://git.apache.org/hadoop-common.git*. After downloading the sources via git, switch to the stable 2.5 using ***git checkout branch-2.5***, or use the appropriate branch name if you are targeting a newer version.

4. **Installing Dependencies and Setting up Environment for Building:**

- The BUILDING.txt file in the root of the source tree has detailed information on the list of requirements and how to install them.
- It also includes information on setting up the environment and a few quirks that are specific to Windows.
- It is strongly recommended that you read and understand it before proceeding.

5. **A few words on Native IO support**

- Hadoop on Linux includes optional Native IO support. However Native IO is mandatory on Windows and without it you will not be able to get your installation working.
- You must follow all the instructions from BUILDING.txt to ensure that Native IO support is built correctly.

6. **Build and Copy the Package files**

- To build a binary distribution run the following command from the root of the source tree.

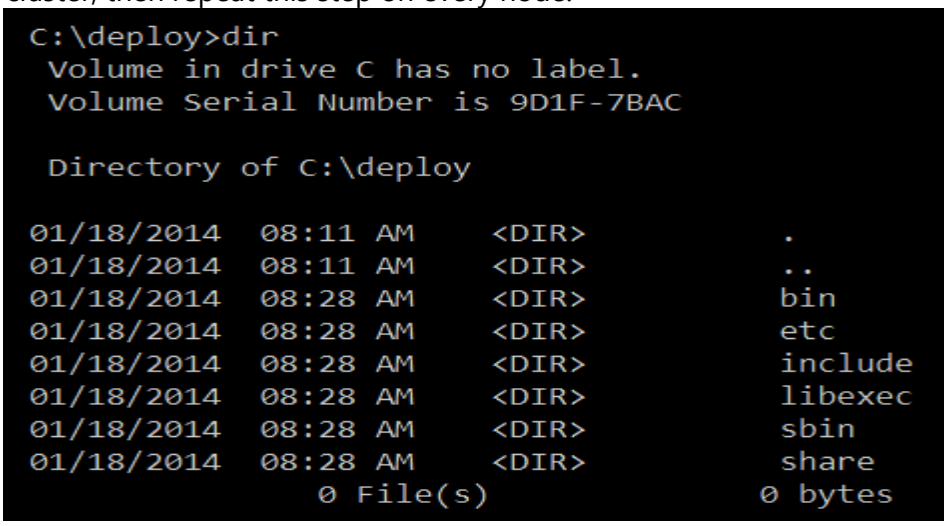
mvn package -Pdist,native-win -DskipTests -Dtar

- Note that this command must be run from a Windows SDK command prompt as documented in BUILDING.txt. A successful build generates a binary hadoop .tar.gz package in _hadoop-dist\target_.

- The Hadoop version is present in the package file name. If you are targeting a different version then the package name will be different.

7. Installation

- Pick a target directory for installing the package. We use c:\deploy as an example.
- Extract the tar.gz file (e.g.hadoop-2.5.0.tar.gz) under c:\deploy.
- This will yield a directory structure like the following. If installing a multi-node cluster, then repeat this step on every node.



8. Starting a Single Node (pseudo-distributed) Cluster:

This section describes the absolute minimum configuration required to start a Single Node (pseudo-distributed) cluster and also run an example MapReduce job.

9. Example HDFS Configuration

- Before you can start the Hadoop Daemons you will need to make a few edits to configuration files.
- The configuration file templates will all be found in c:\deploy\etc\hadoop, assuming your installation directory is c:\deploy.

- First edit the file hadoop-env.cmd to add the following lines near the end of the file.

```
set HADOOP_PREFIX=c:\deploy
set HADOOP_CONF_DIR=%HADOOP_PREFIX%\etc\hadoop
set YARN_CONF_DIR=%HADOOP_CONF_DIR%
set PATH=%PATH%;%HADOOP_PREFIX%\bin
```

- Edit or create the file core-site.xml and make sure it has the following configuration key:

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://0.0.0.0:19000</value>
  </property>
</configuration>
```

- Edit or create the file hdfs-site.xml and add the following configuration key:

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

- Finally, edit or create the file slaves and make sure it has the following entry:

localhost

- The default configuration puts the HDFS metadata and data files under \tmp on the current drive. In the above example this would be c:\tmp. For your first test setup you can just leave it at the default.

10. Example YARN Configuration:

- Edit or create mapred-site.xml under %HADOOP_PREFIX%\etc\hadoop and add the following entries, replacing %USERNAME% with your Windows user name.

```
<configuration>

  <property>
    <name>mapreduce.job.user.name</name>
    <value>%USERNAME%</value>
  </property>

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

 <property>
   <name>yarn.apps.stagingDir</name>
   <value>/user/%USERNAME%/staging</value>
 </property>

 <property>
   <name>mapreduce.jobtracker.address</name>
   <value>local</value>
 </property>

</configuration>
```

- Finally, edit or create yarn-site.xml and add the following entries:

```
<configuration>
  <property>
    <name>yarn.server.resourcemanager.address</name>
    <value>0.0.0.0:8020</value>
  </property>

  <property>
    <name>yarn.server.resourcemanager.application.expiry.interval</name>
    <value>60000</value>
  </property>

  <property>
    <name>yarn.server.nodemanager.address</name>
    <value>0.0.0.0:45454</value>
  </property>

  <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>

  <property>
    <name>yarn.server.nodemanager.remote-app-log-dir</name>
    <value>/app-logs</value>
  </property>

  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/dep/logs/userlogs</value>
  </property>

  <property>
    <name>yarn.server.mapreduce-appmanager.attempt-listener.bindAddress</name>
    <value>0.0.0.0</value>
  </property>

  <property>
    <name>yarn.server.mapreduce-appmanager.client-service.bindAddress</name>
    <value>0.0.0.0</value>
  </property>

  <property>
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
  </property>

  <property>
    <name>yarn.log-aggregation.retain-seconds</name>
    <value>-1</value>
  </property>

  <property>
    <name>yarn.application.classpath</name>
    <value>%HADOOP_CONF_DIR%,%HADOOP_COMMON_HOME%/share/hadoop/common/*,%HADOOP_COMMON_HOME
%/share/hadoop/common/lib/*,%HADOOP_HDFS_HOME%/share/hadoop/hdfs/*,%HADOOP_HDFS_HOME%/share/hadoop/hdfs/lib/*,
%HADOOP_MAPRED_HOME%/share/hadoop/mapreduce/*,%HADOOP_MAPRED_HOME%/share/hadoop/mapreduce/lib/*,
%HADOOP_YARN_HOME%/share/hadoop/yarn/*,%HADOOP_YARN_HOME%/share/hadoop/yarn/lib/*</value>
  </property>
</configuration>
```

11. Initialize Environment Variables

- Run **c:\deploy\etc\hadoop\hadoop-env.cmd** to setup environment variables that will be used by the startup scripts and the daemons.

12. Format the FileSystem

- Format the filesystem with the following command:

```
%HADOOP_PREFIX%\bin\hdfs namenode -format
```

- This command will print a number of filesystem parameters. Just look for the following two strings to ensure that the format command succeeded.

```
14/01/18 08:36:23 INFO namenode.FSImage: Saving image file \tmp\hadoop-username
\dfs\name\current\fsimage.ckpt_0000000000000000000 using no compression
14/01/18 08:36:23 INFO namenode.FSImage: Image file \tmp\hadoop-username\dfs
\name\current\fsimage.ckpt_0000000000000000000 of size 200 bytes saved in 0
seconds.
```

13. Start HDFS Daemons:

- Run the following command to start the Name{{`Node and Data}}`Node on localhost.

```
%HADOOP_PREFIX%\sbin\start-dfs.cmd
```

- To verify that the HDFS daemons are running, try copying a file to HDFS.

```
C:\deploy>%HADOOP_PREFIX%\bin\hdfs dfs -put myfile.txt /

C:\deploy>%HADOOP_PREFIX%\bin\hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - username supergroup          4640 2014-01-18 08:40 /myfile.txt

C:\deploy>
```

14. Start YARN Daemons and run a YARN job

- Finally, start the YARN daemons.

```
%HADOOP_PREFIX%\sbin\start-yarn.cmd
```

- The cluster should be up and running! To verify, we can run a simple wordcount job on the text file we just copied to HDFS.

```
%HADOOP_PREFIX%\bin\yarn jar %HADOOP_PREFIX%\share\hadoop\mapreduce\hadoop-
        mapreduce-examples-2.5.0.jar wordcount /myfile.txt /out
```

15. Example: WordCount v1.0

- Before we jump into the details, lets walk through an example MapReduce application to get a flavour for how they work.

- WordCount is a simple application that counts the number of occurrences of each word in a given input set.

- This works with a local-standalone, pseudo-distributed or fully-distributed Hadoop installation

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

public static class TokenizerMapper extends Mapper<Object, Text, Text,
IntWritable>{
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();

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

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

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

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

16. USAGE:

- Assuming environment variables are set as follows:

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

- Compile WordCount.java and create a jar:

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

-Assuming that:

- /user/joe/wordcount/input - input directory in HDFS
- /user/joe/wordcount/output - output directory in HDFS

-Sample text-files as input:

```
$ bin/hadoop fs -ls /user/joe/wordcount/input/
/user/joe/wordcount/input/file01
/user/joe/wordcount/input/file02

$ bin/hadoop fs -cat /user/joe/wordcount/input/file01
Hello World Bye World

$ bin/hadoop fs -cat /user/joe/wordcount/input/file02
Hello Hadoop Goodbye Hadoop
```

- Run the application:

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

OUTPUT:

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

RESULT:

Thus, the process to install Hadoop single node cluster was done successfully and simple applications like word count was successfully executed and output was obtained.