

<b>Exp. No: 1</b>	<b>Installation of VirtualBox with Linux OS</b>
<b>Date:</b>	

**Aim:**

To Install VirtualBox with different flavours of Linux OS on top of windows7 or 8 or 10 OS.

**Procedure:**

The installation is divided into

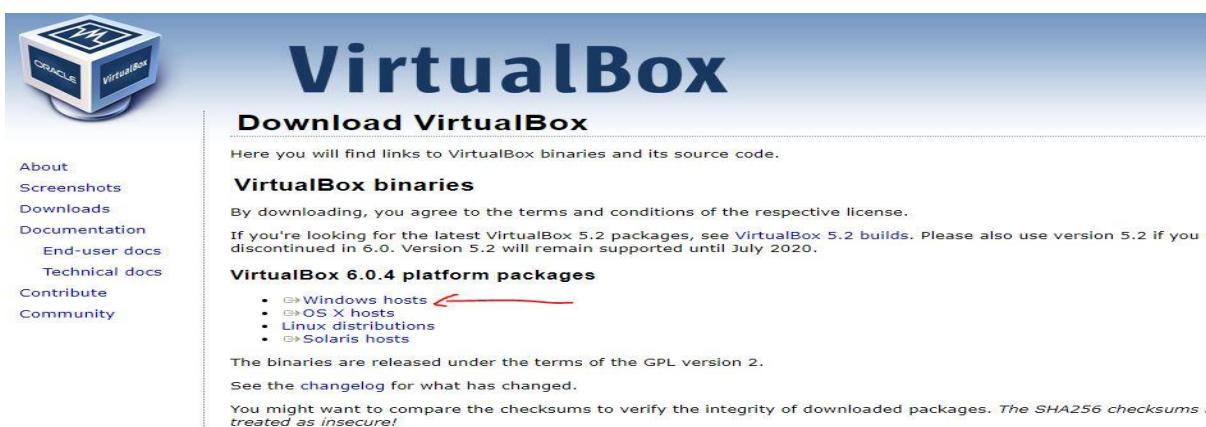
1. Installation of VirtualBox on Windows 10/8/7
2. Creation of Ubuntu (Linux) VM
3. Installation of Linux OS on VirtualBox

**Installation of VirtualBox on Windows 10/8/7**

1. **Download** VirtualBox software from **Oracle official website**.
2. **Double-click** on downloaded **VirtualBox** Win.exe file to bring up the welcome screen. Click **Next**.
3. Installation files and set the installation path. If you are not familiar, then keep the default configuration, select the **Next** button.
4. Leave the pre-selected **VirtualBox** shortcuts as it is and click on **Next** button.
5. When installing VirtualBox, it involves network functions. The wizard will automatically create a **virtual network card**, which will temporarily interrupt your network. But of course, it will return to normal immediately. So, click **Yes**.
6. Now you can go to install this virtualization software. Click **Install**. During the period, you can see that the current network was interrupted and immediately resumed.
7. Click **Finish** to launch Oracle VM VirtualBox.

**Screenshots of the above steps:****Step 1: Download VirtualBox for Windows 10/8/7**

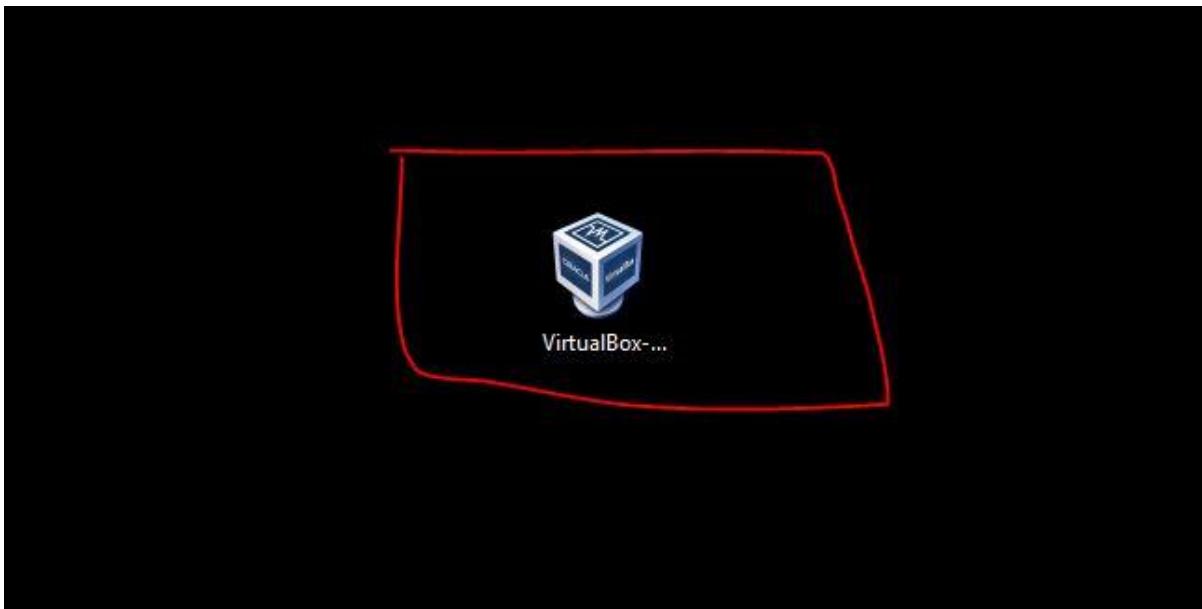
**Download** VirtualBox software from **Oracle official website**: [Download VirtualBox](https://www.virtualbox.org/wiki/Downloads)



The screenshot shows the Oracle VirtualBox download page. On the left is a sidebar with links: About, Screenshots, Downloads, Documentation, End-user docs, Technical docs, Contribute, and Community. The main content area has a large blue header "VirtualBox". Below it is a "Download VirtualBox" button. A sub-section titled "VirtualBox binaries" contains the text: "Here you will find links to VirtualBox binaries and its source code." It includes a note about agreeing to terms and conditions, a link to "VirtualBox 5.2 builds" (which is crossed out), and a note that version 5.2 will be supported until July 2020. A red arrow points to the "Windows hosts" link under "VirtualBox 6.0.4 platform packages". At the bottom, there's a note about GPL version 2, a changelog link, and a note about comparing checksums.

## Step 2: Run the VirtualBox.exe file

The downloaded VirtualBox file will be in EXE format to run that just double click on it and run it as administrator.



Click on **Next** button to start Oracle VirtualBox installation Setup Wizard.

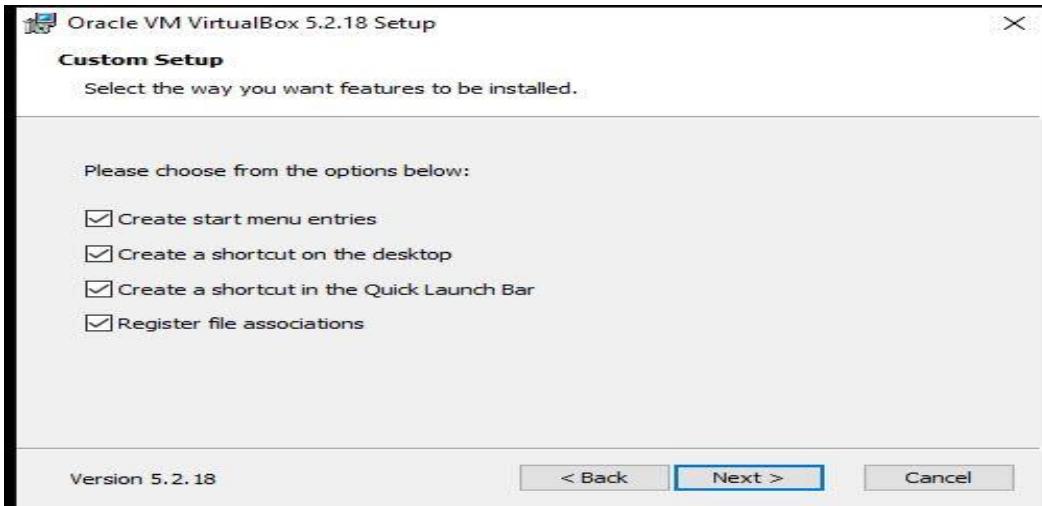


## Step 3: VirtualBox shortcuts

At this stage, you will see multiple shortcuts:

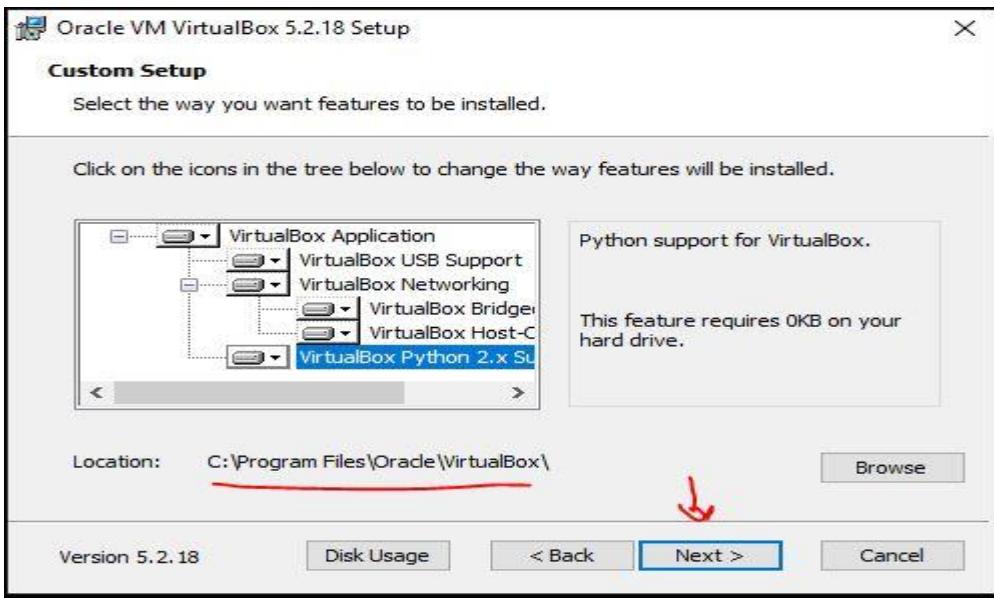
- **Create start menu entries:** To create a Virtualbox shortcut in the start menu of the Windows 10/8/7
- **Create a shortcut on the desktop:** This will create a shortcut on Desktop
- **Create a shortcut in the Quick Launch Bar:** You will get a shortcut in the Taskbar.
- **Register file associations:** Create Virtualbox file entries in Windows registries.

Leave them as it is and click on the **NEXT** button.



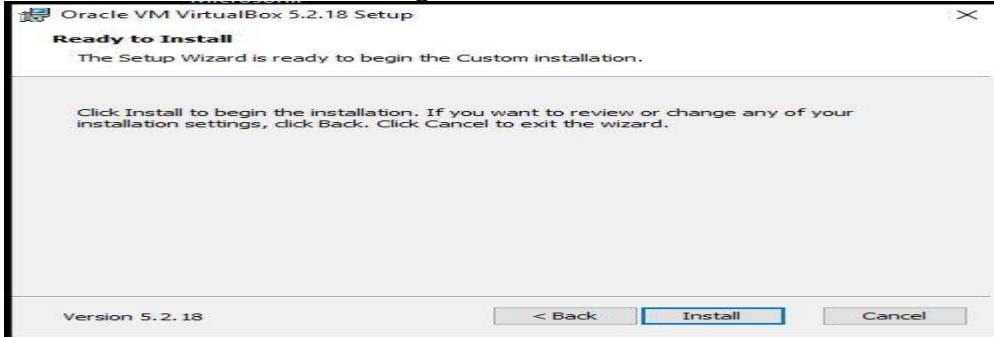
#### Step 4: File Location

By default the VirtualBox will install its core files in the C: Drive. In case you have low space on the C: Drive, then just click on the Browse button and select the location where you want to install it. However, if you are not acquainted with this option then simply leave it as default and click on **NEXT** button.



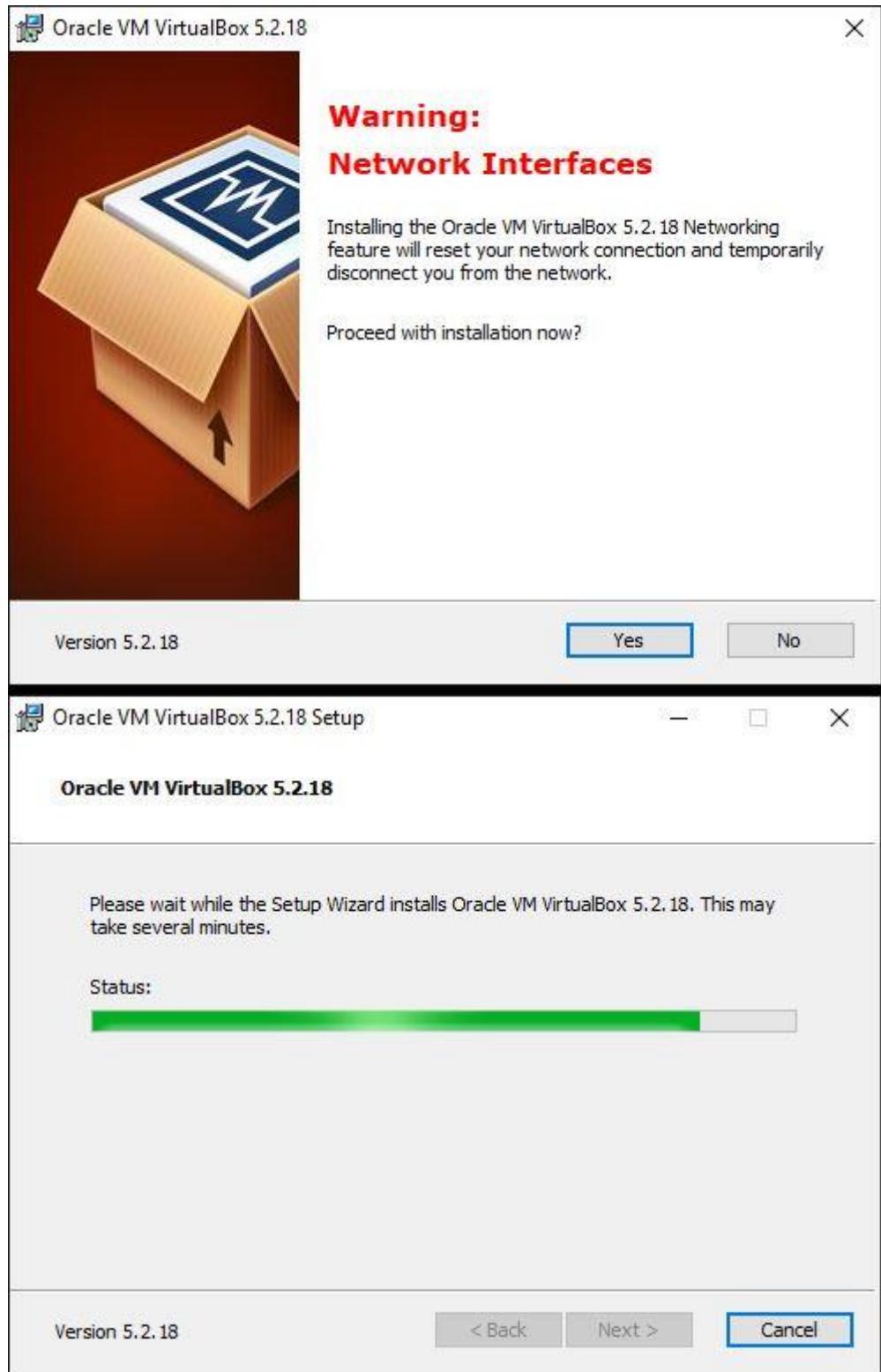
#### Step 5: Install VirtualBox

Click on the **Install** button to begin the installation.



## Step 6: Warning: Network Interfaces

To create Virtual Adapters, the VirtualBox will reset your network connection and disconnect it temporarily for a few seconds and then again it will return to its normal state. So, click on the **YES** button.



## Step 7: Installation is completed

After installing, the installation wizard will show you a **Finish** button, click on that and it will start the VirtualBox on your Windows 10/7/8 machines.



## Creation of Ubuntu VM in VirtualBox

### Step 1: Download Ubuntu OS

The open source Ubuntu Linux comes in different flavors and you can download any of them from the official Ubuntu's website. Here is the Link: [www.ubuntu.com/download/desktop](http://www.ubuntu.com/download/desktop).

Note: If you already have the Ubuntu.iso file then leave this step.



## Download Ubuntu Desktop

### Ubuntu 16.04.3 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2021, of free security and maintenance updates, guaranteed.

[Download](#)

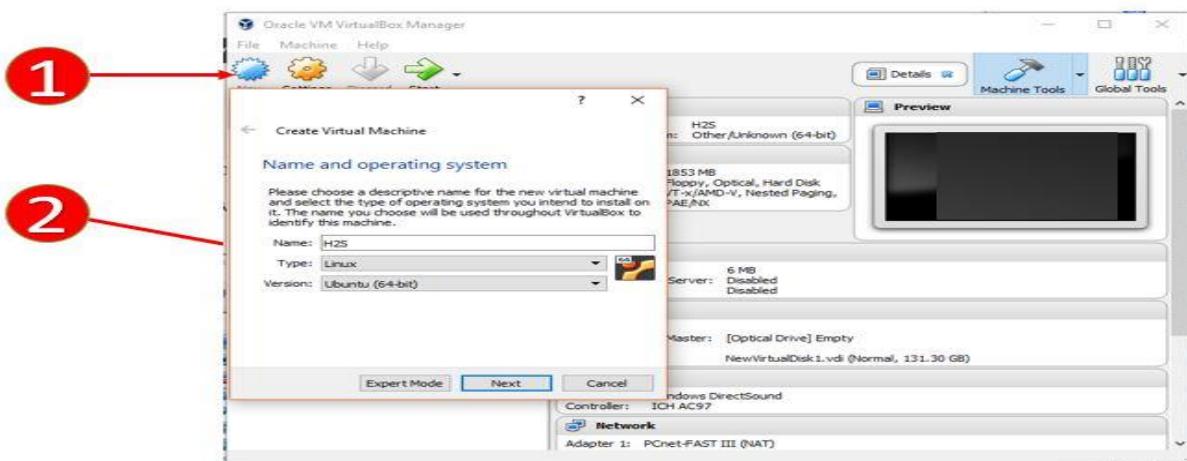
[Alternative downloads and torrents >](#)

[Ubuntu 16.04 LTS release notes](#)

Recommended system requirements:

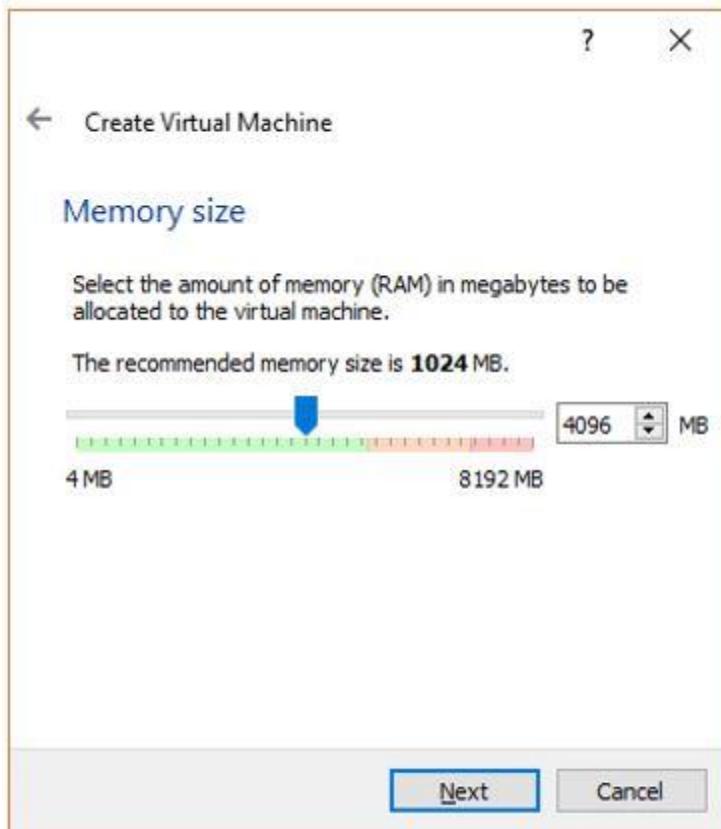
- 2 GHz dual core processor or better
- 2 GB system memory
- 25 GB of free hard drive space
- Either a DVD drive or a USB port for the installer media

**Step 2:** After successful Virtualbox installation, run it to create an **Ubuntu VM**. Click on **New** and give some name to your Ubuntu VM. For example, here we have used **H2S**. From the type drop-down box select the OS type which is **Linux** and Version is **Ubuntu 64 bit**. If you have **Ubuntu 32 bit** version then please select that.



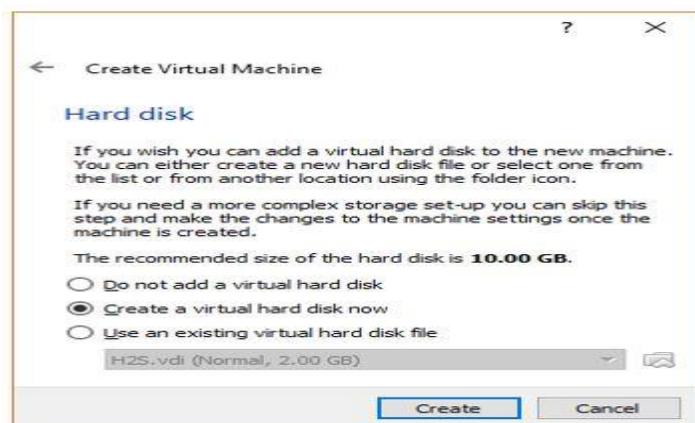
### Step 3: How Much Memory Do You Give Your Virtual Machine

In this step, the Vitrualbox will ask to set the Virtual Machine Memory Size for Ubuntu VM. The recommended RAM for Ubuntu OS is 2 GB or 2048 MB but you can assign more for better performance. For example here in our Windows 10 PC, we have maximum 8GB memory and out of that, we are going to assign 4GB to Ubuntu VM.



### Step 4: Create A Virtual Hard Drive For Ubuntu VM (Virtual Machine)

After assigning the memory, its time to provide some space for the installation of Ubuntu VM. To create a new virtual hard disk select the option “**Create a virtual hard disk now**” and click “Create”.



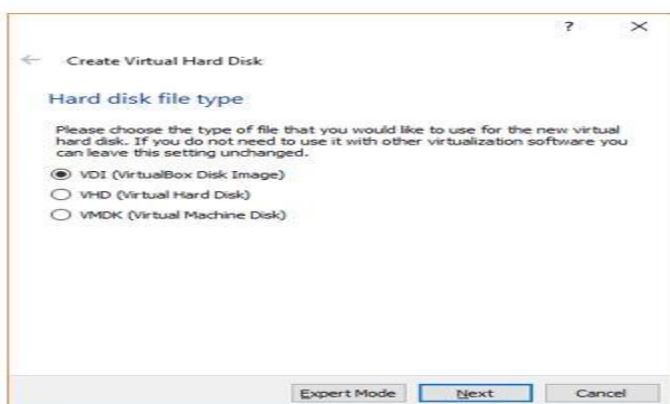
## Step 5: Choose Virtual Hard disk Type

The Virtualbox offers three type of Virtual hard drives:

- 1.VDI- Virtual Disk Image
- 2.VHD- Virtual Hard Disk
- 3.VMDK- Virtual Machine Disk

If you are planning to use the Virtual hard drive with some other virtualization software in future such as with VM player or Windows Hyper-V then you choose according to that otherwise leave it as it is “**VDI**” and click on **NEXT**.

There are a number of different hard drive types that you can choose from. Choose “VDI” and click “Next”.



## Step 6: Storage on Physical Hard disk for Ubuntu VM

To install Ubuntu Virtual Machine files on physical storage of Windows 10, the Virtualbox offers two options:

1. Dynamically allocated
2. Fixed Size

The Dynamically allocated hard disk option will only use space as it required. For example, you assigned 30 GB to Ubuntu VM but if it requires 10Gb initially then the Virtualbox uses only that and not going to block the whole 30GB. And in future, it requires more, expands according to that. It is good in terms of disk space but not performance wise.

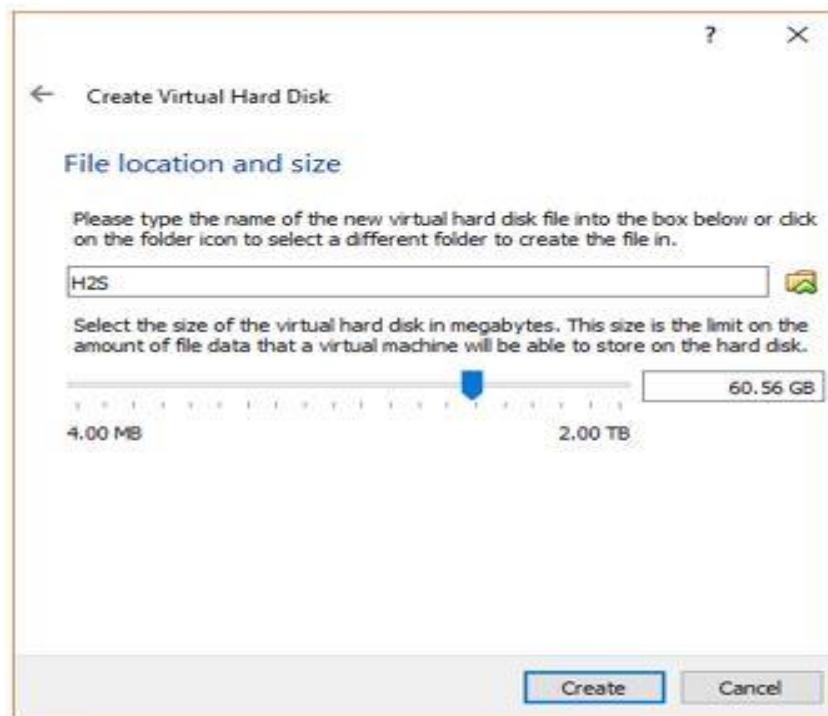
Fixed Size on another hand block whole space you have assigned to the VM. For example, if you allocated the 30GB, then the machine will straight away assign that portion from the physical hard drive to Ubuntu VM. The Fixed size allocation is better for performance but take some time to create if you are assigning a large amount of space.

Choose the option you would like and click “Next”.



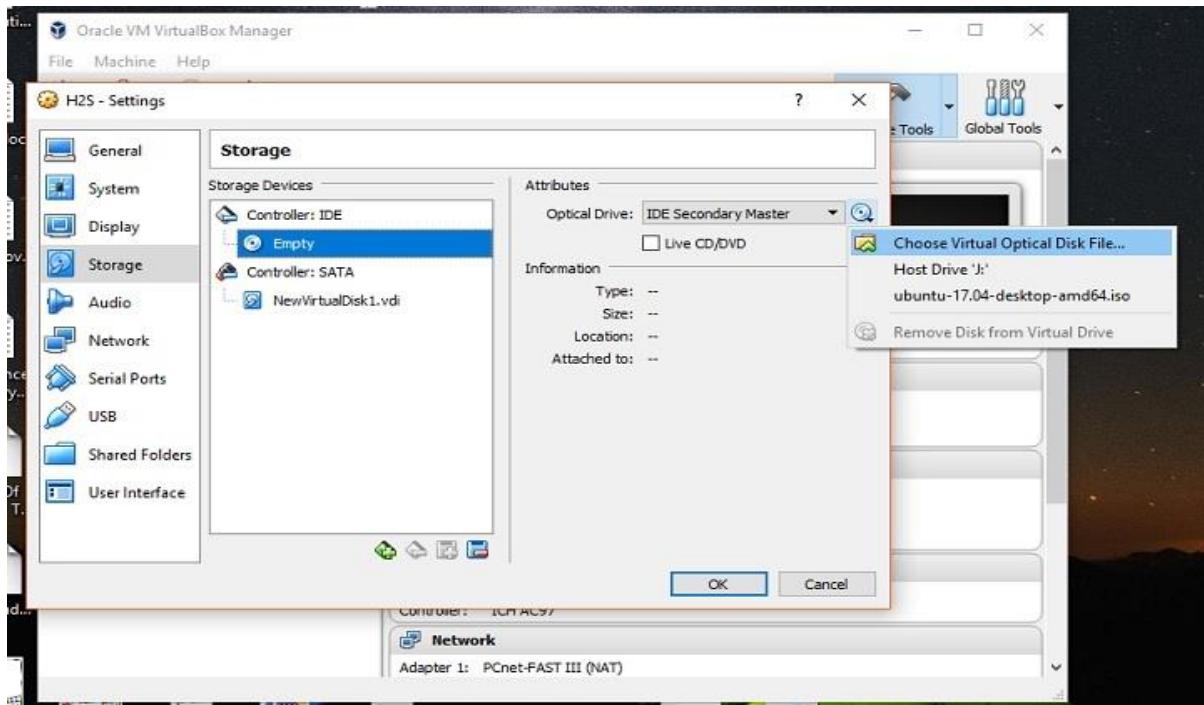
## Step 7: Virtual Harddrive File location and Size

Give some name to your virtual hard disk and select the amount of space you want to assign the Ubuntu VM. The minimum recommended space is 25GB. You can assign more for better performance.



## Step 8: Assign Ubuntu ISO to VirtualBox

Go to setting and from storage click on the empty CD-Rom icon and from the Optical drive option **choose the Virtual optical Disk File** and select the Ubuntu.iso file which is our downloaded beginning of this article. After selecting the ISO file click **OK**.

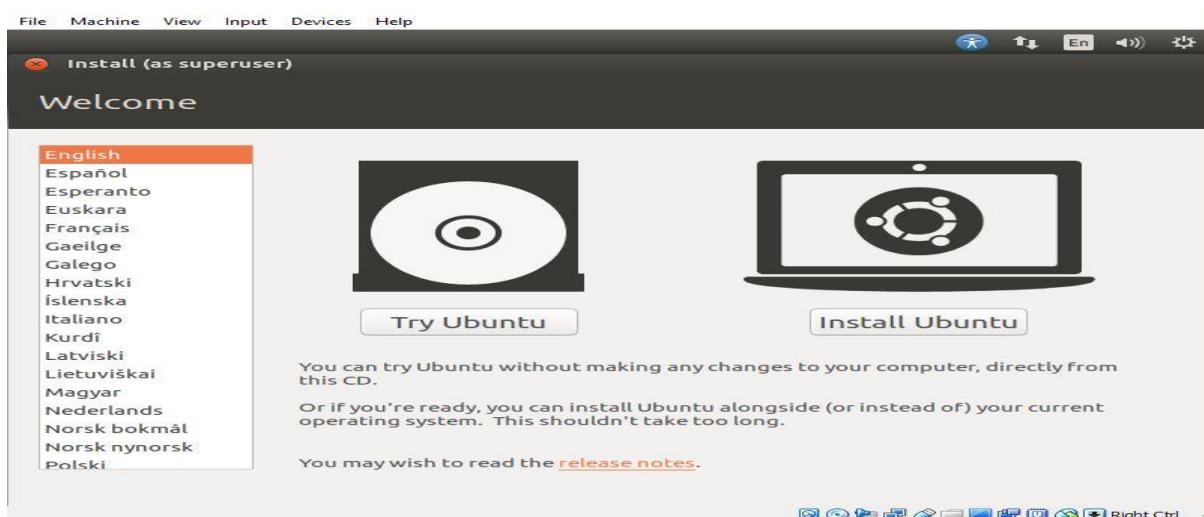


## Installation of Ubuntu OS on VirtualBox

**Step 1:** On the top of the Virtual box you will an option “**START**”, click on that to initialize the Ubuntu installation process on Windows 10.

**Step 2:** The Ubuntu first screen will load two options **Try Ubuntu** or **Install Ubuntu**.

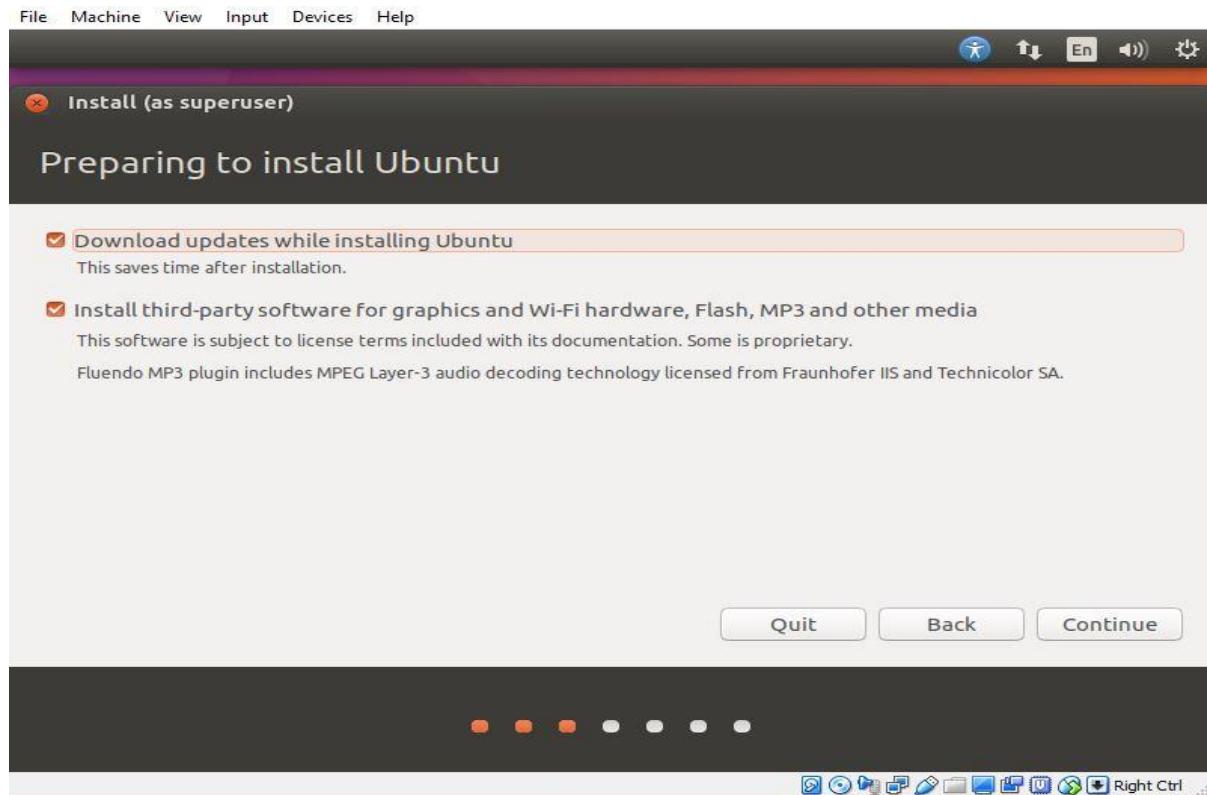
Select the installation language and after that the “**Install Ubuntu**” option.



**Step 3:** If you have enough internet bandwidth and then you can select the option download the updates while installing Ubuntu.

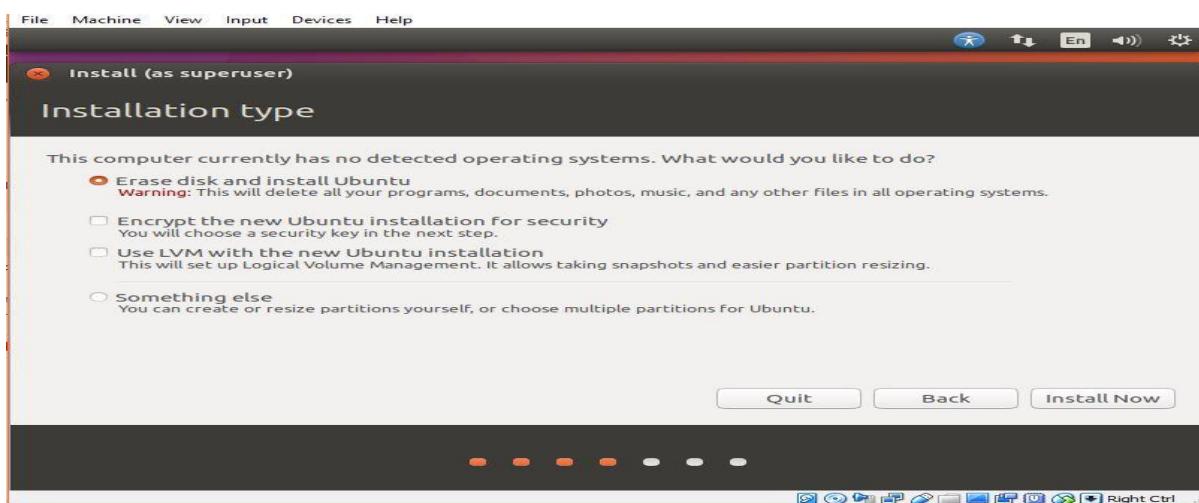
The second option doesn't require an internet connection and recommended to select it to install third party software such as graphics driver, Mp3 player, flash and other media files.

Click "Continue".

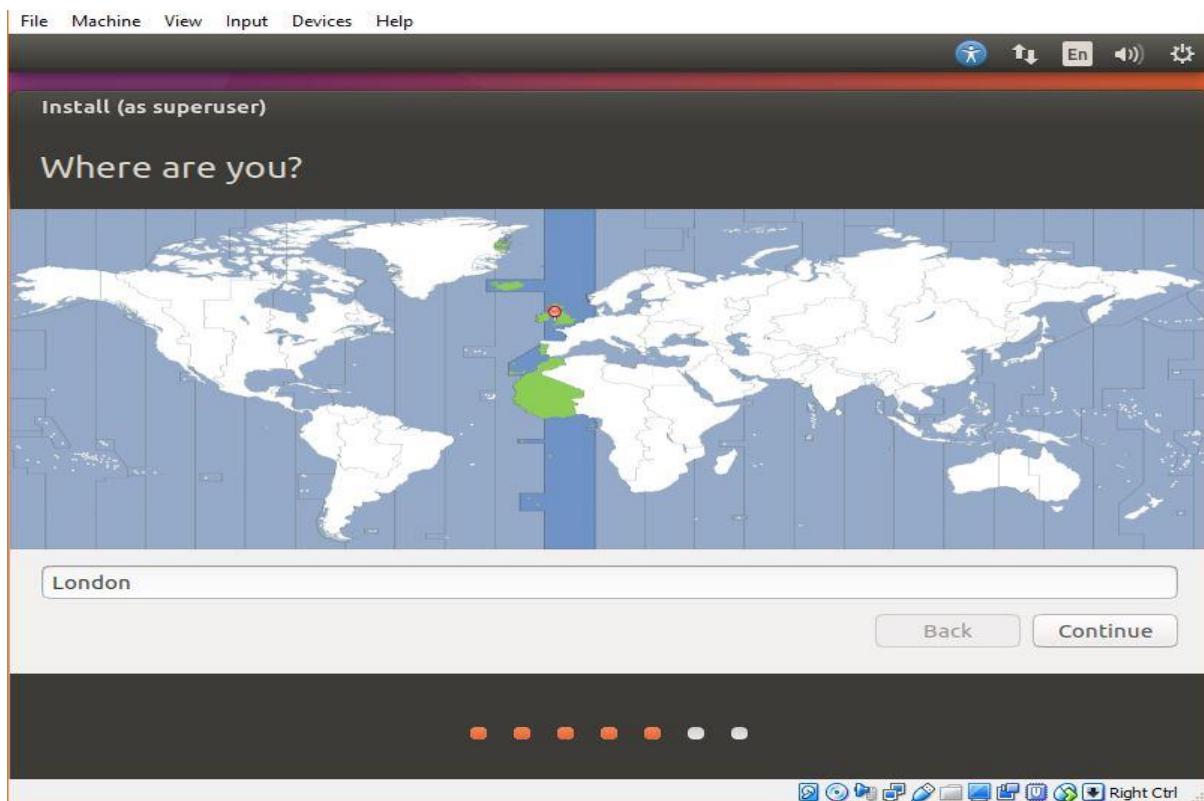


**Step 4:** In this step, you will decide how you want to decide the Ubuntu either clean installation or dual boot with some other OS. Leave the default option the "Erase disk and install Ubuntu" option because it is on the virtual machine won't going to affect the physical Windows 10 machine.

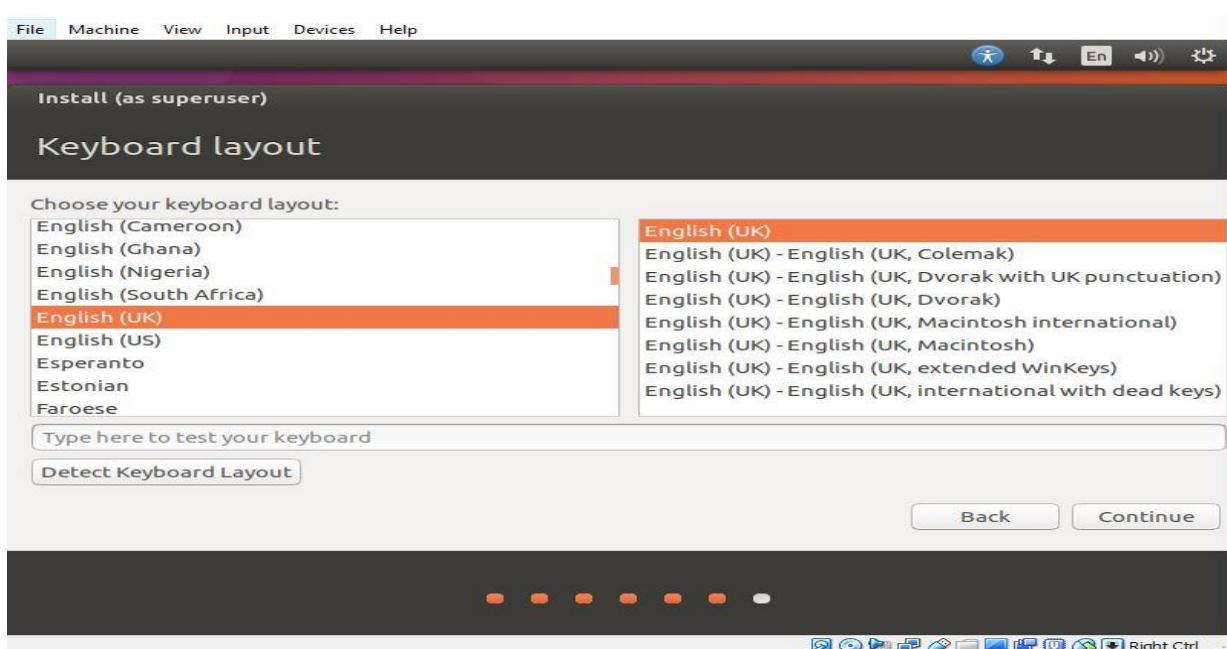
Click "**Install Now**" and "**Continue**".



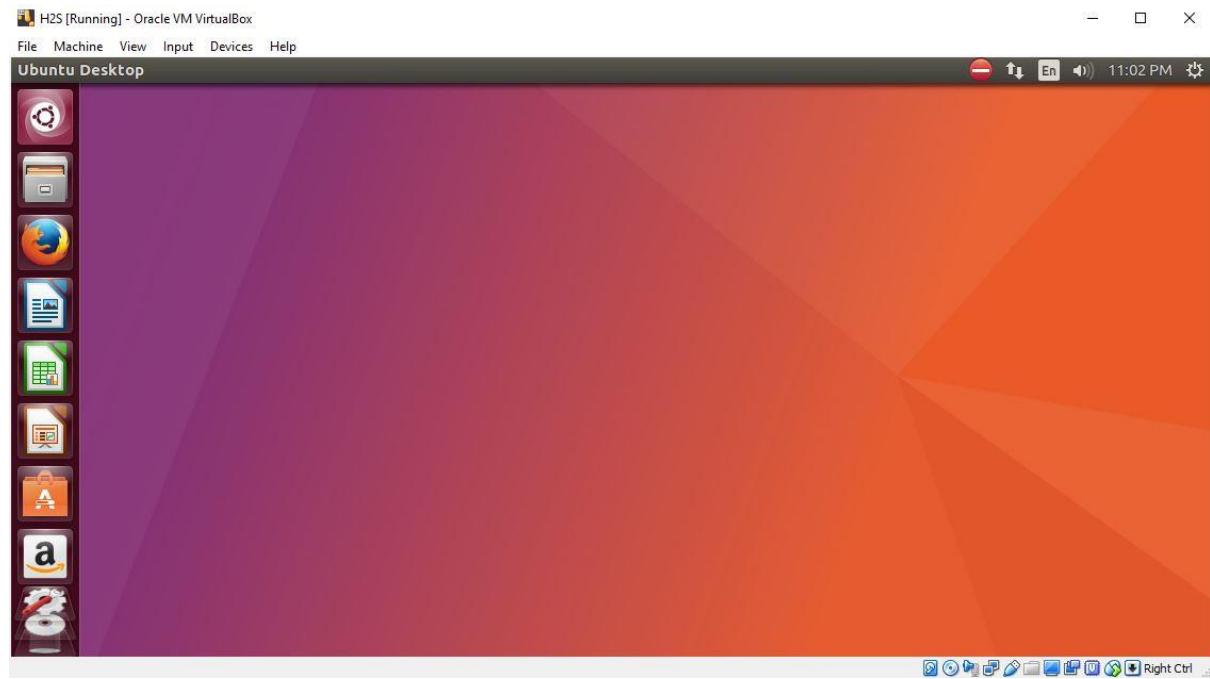
**Step 5:** Select your country to sync the Ubuntu OS time zone with your's and click “Continue”.



**Step 6:** Click on the “Detect Keyboard Layout” to automatically detect your keyboard layout and if the machine not able to do it, you can select it manually. Click “Continue”.



**Step 7:** Create a user and set the password for your Virtual Ubuntu machine and click on continue to install the Ubuntu Virtualbox.



Finally, the Ubuntu is installed on **VirtualBox on Windows 10** as host machine

### **Result:**

Thus, the Installation of VirtualBox with different flavours of Linux OS on top of windows7 or 8 or 10 OS completed successfully.

<b>Exp. No: 2</b>	<b>Installation of C Compiler in VirtualBox</b>
<b>Date:</b>	

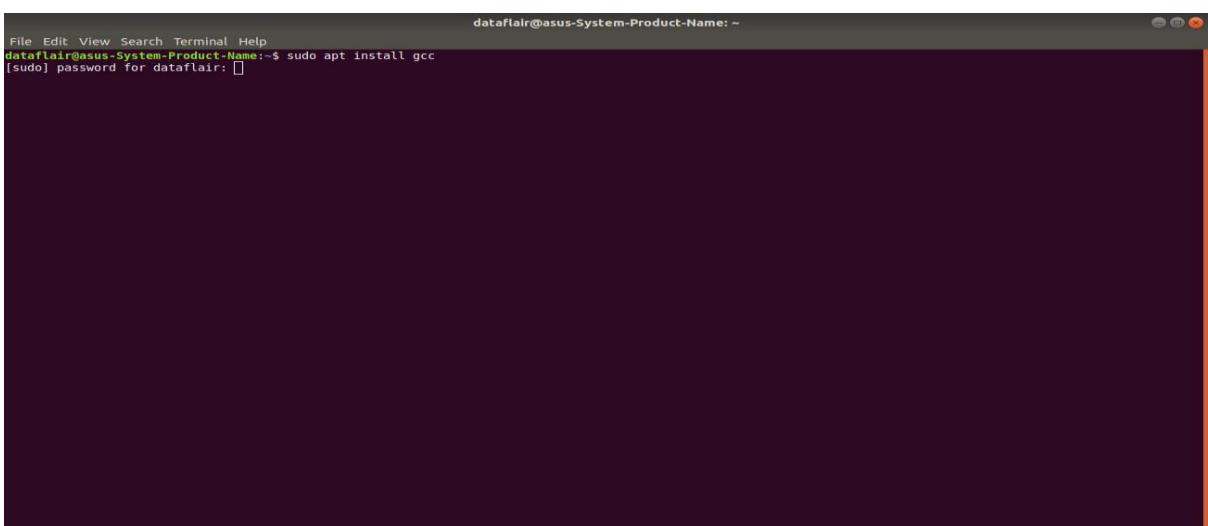
**Aim:**

To Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.

**Procedure:**

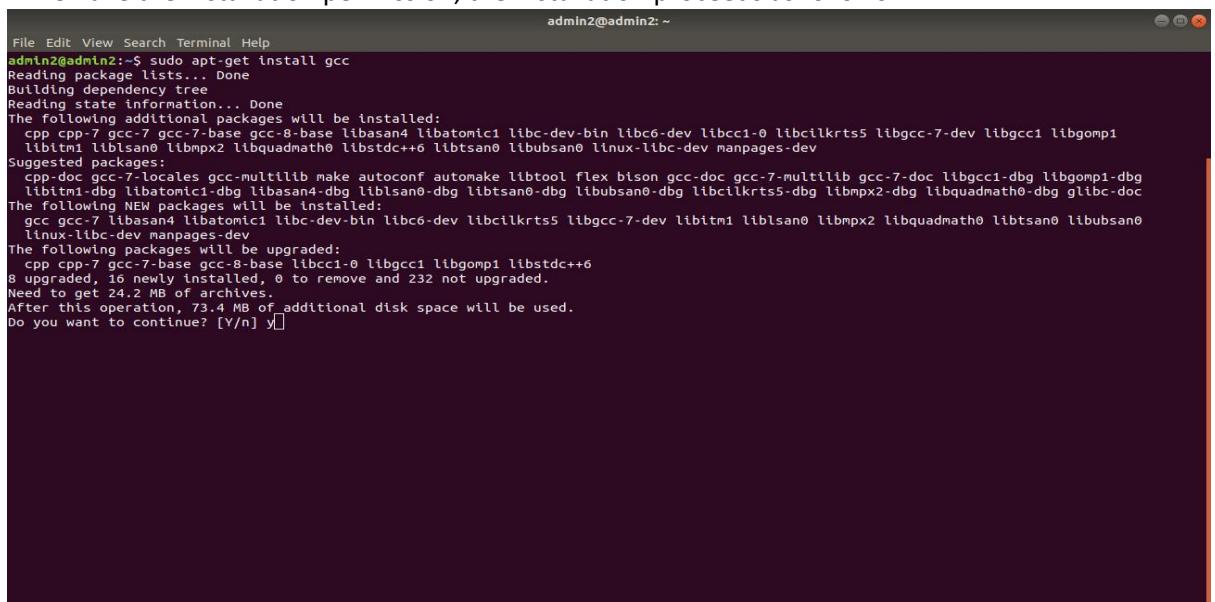
1. Open the VirtualBox application & then Start the Ubuntu Virtual Machine installed.
2. Open the Terminal Command prompt by clicking terminal icon in desktop or using shortcut **Ctrl + Alt + T**
3. Enter the following command in the terminal window  
`sudo apt install GCC`

Here, GCC is the C Compiler. Enter admin password if prompted.



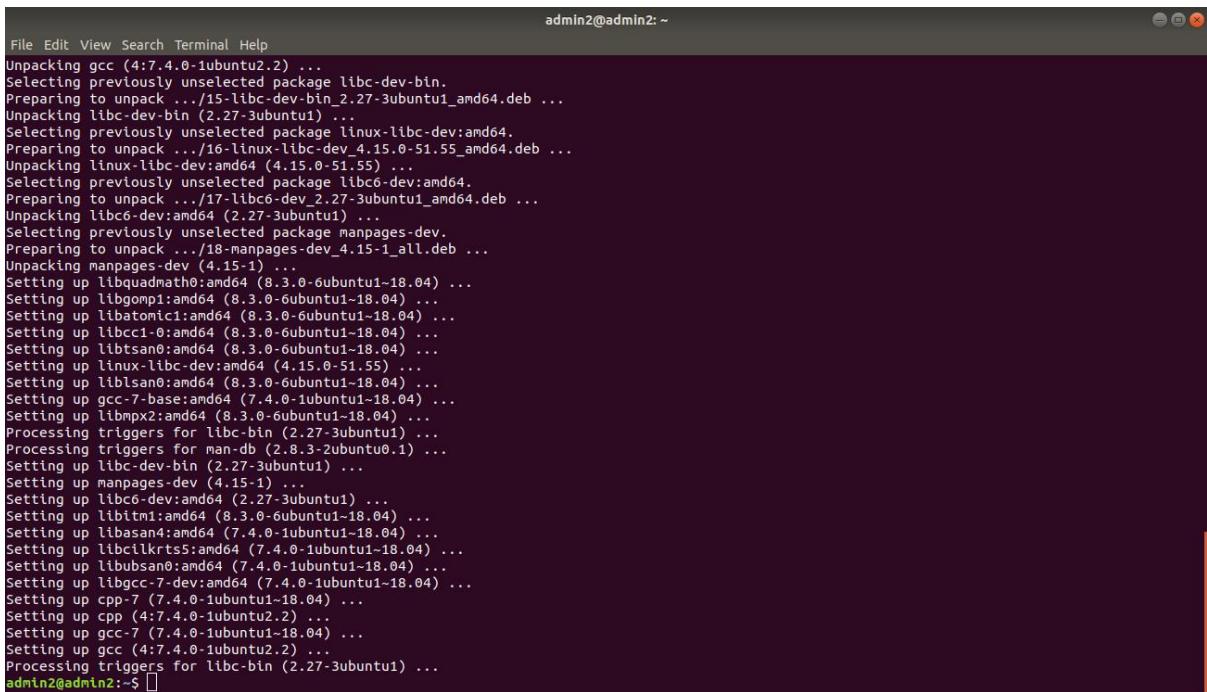
```
File Edit View Search Terminal Help
dataflair@asus-System-Product-Name:~$ sudo apt install gcc
[sudo] password for dataflair: [ ]
```

4. If we have the installation permission, the installation proceeds as follows



```
File Edit View Search Terminal Help
admin2@admin2:~$ sudo apt-get install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cpp cpp-7 gcc-7-base gcc-8-base libasan4 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrt5 libgcc-7-dev libgcc1 libgomp1
  libitm1 liblsan0 libmpx2 libquadmath0 libstdc++6 libtsan0 libubsan0 linux-libc-dev manpages-dev
Suggested packages:
  cpp-doc gcc-7-locales gcc-multilib make autoconf automake libtool flex bison gcc-doc gcc-7-multilib gcc-7-doc libgcc1-dbg libgomp1-dbg
  libitm1-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrt5-dbg libmpx2-dbg libquadmath0-dbg glibc-doc
The following NEW packages will be installed:
  gcc gcc-7 libasan4 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrt5 libgcc-7-dev libitm1 liblsan0 libmpx2 libquadmath0 libtsan0 libubsan0
  linux-libc-dev manpages-dev
The following packages will be upgraded:
  cpp cpp-7 gcc-7-base gcc-8-base libcc1-0 libgcc1 libgomp1 libstdc++6
8 upgraded, 16 newly installed, 0 to remove and 232 not upgraded.
Need to get 24.2 MB of archives.
After this operation, 73.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y[ ]
```

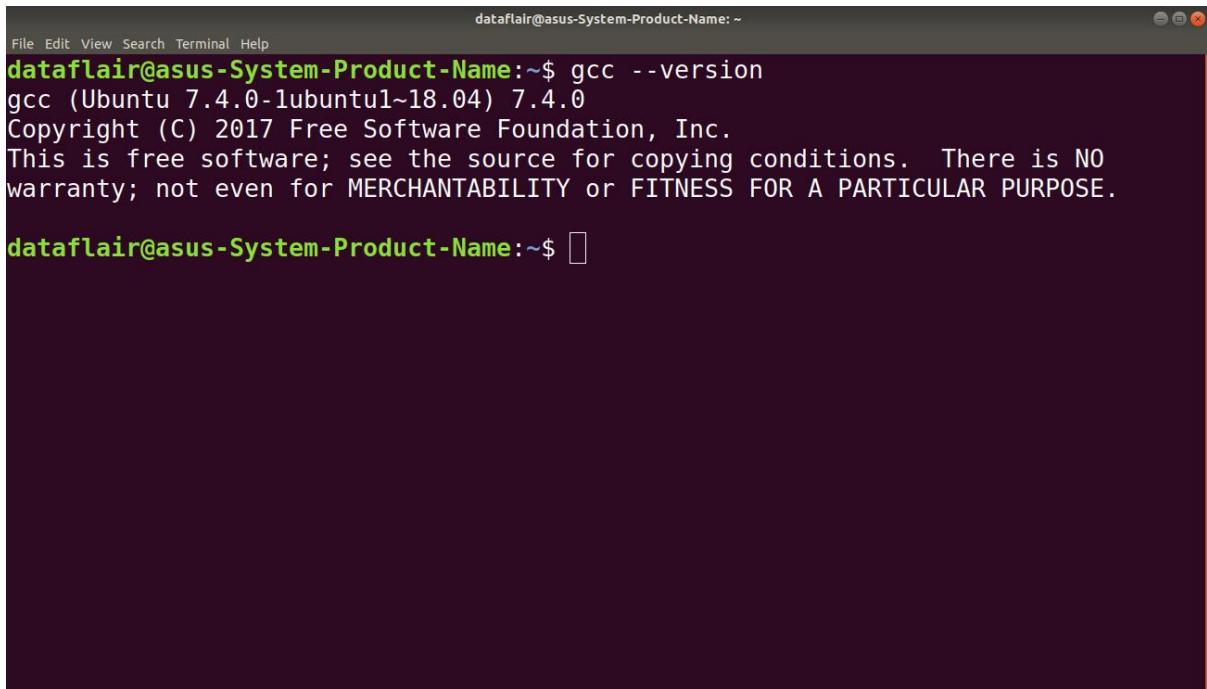
5. Type ‘y’ when the command prompt asks “Do you want to continue?” and then press Enter to continue the installation.



```
File Edit View Search Terminal Help
Unpacking gcc (4:7.4.0-1ubuntu2.2) ...
Selecting previously unselected package libc-dev-bin.
Preparing to unpack .../15-libc-dev-bin_2.27-3ubuntu1_amd64.deb ...
Unpacking libc-dev-bin (2.27-3ubuntu1) ...
Selecting previously unselected package linux-libc-dev:amd64.
Preparing to unpack .../16-linux-libc-dev_4.15.0-51.55_amd64.deb ...
Unpacking linux-libc-dev:amd64 (4.15.0-51.55) ...
Selecting previously unselected package libc6-dev:amd64.
Preparing to unpack .../17-libc6-dev_2.27-3ubuntu1_amd64.deb ...
Unpacking libc6-dev:amd64 (2.27-3ubuntu1) ...
Selecting previously unselected package manpages-dev.
Preparing to unpack .../18-manpages-dev_4.15-1_all.deb ...
Unpacking manpages-dev (4.15-1) ...
Setting up libquadmath0:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up libgomp1:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up libatomic1:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up libcc1-0:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up libtsan0:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up linux-libc-dev:amd64 (4.15.0-51.55) ...
Setting up liblsan0:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up gcc-7-base:amd64 (7.4.0-1ubuntu1-18.04) ...
Setting up libmpx2:amd64 (8.3.0-6ubuntu1-18.04) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up libc-dev-bin (2.27-3ubuntu1) ...
Setting up manpages-dev (4.15-1) ...
Setting up libc6-dev:amd64 (2.27-3ubuntu1) ...
Setting up libitm1:amd64 (8.3.0-6ubuntu1-18.04) ...
Setting up libasan4:amd64 (7.4.0-1ubuntu1-18.04) ...
Setting up libcilkrtts5:amd64 (7.4.0-1ubuntu1-18.04) ...
Setting up libubsan0:amd64 (7.4.0-1ubuntu1-18.04) ...
Setting up libgcc-7-dev:amd64 (7.4.0-1ubuntu1-18.04) ...
Setting up cpp-7 (7.4.0-1ubuntu1-18.04) ...
Setting up cpp (4:7.4.0-1ubuntu2.2) ...
Setting up gcc-7 (7.4.0-1ubuntu1-18.04) ...
Setting up gcc (4:7.4.0-1ubuntu2.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
admin2@admin2:~$
```

6. After successful installation, Verify the installation by checking the version number of gcc using following command.

**GCC — version**



```
File Edit View Search Terminal Help
dataflair@asus-System-Product-Name:~$ gcc --version
gcc (Ubuntu 7.4.0-1ubuntu1~18.04) 7.4.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

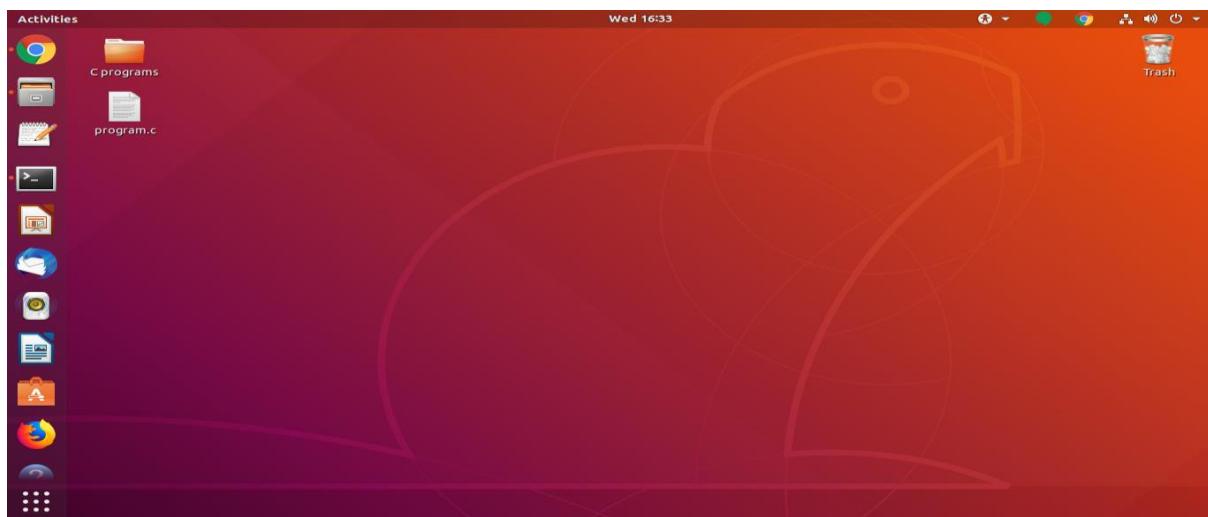
dataflair@asus-System-Product-Name:~$
```

7. In terminal, Move the desired directory (Ex: Desktop) where you want to save the program. **cd Desktop**

```
File Edit View Search Terminal Help  
dataflair@asus-System-Product-Name:~$ cd Desktop  
dataflair@asus-System-Product-Name:~/Desktop$ [ ]
```

8. The command for creating a program in C is: **touch program.c**

Now, a file has been created in our Desktop folder called program.c



8. Open this file and write a basic code – “Hello World!” or any code required.

```
#include<stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

The screenshot shows a terminal window with the following content:

```
program.c
$ ./hello
Hello World!
```

9. To compile the code, we use the GCC command:

**GCC program.c -o program**

```
dataflair@asus-System-Product-Name:~/Desktop
File Edit View Search Terminal Help
dataflair@asus-System-Product-Name:~$ cd Desktop
dataflair@asus-System-Product-Name:~/Desktop$ touch program.c
dataflair@asus-System-Product-Name:~/Desktop$ gcc program.c -o program
dataflair@asus-System-Product-Name:~/Desktop$ ./program
```

10. To run the program, use the command: **./program**

```
dataflair@asus-System-Product-Name:~/Desktop
File Edit View Search Terminal Help
dataflair@asus-System-Product-Name:~$ cd Desktop
dataflair@asus-System-Product-Name:~/Desktop$ touch program.c
dataflair@asus-System-Product-Name:~/Desktop$ gcc program.c -o program
dataflair@asus-System-Product-Name:~/Desktop$ ./program
```

11. On successful execution, the output will look like

```
dataflair@asus-System-Product-Name:~/Desktop
File Edit View Search Terminal Help
dataflair@asus-System-Product-Name:~$ cd Desktop
dataflair@asus-System-Product-Name:~/Desktop$ touch program.c
dataflair@asus-System-Product-Name:~/Desktop$ gcc program.c -o program
dataflair@asus-System-Product-Name:~/Desktop$ ./program
Hello World!
dataflair@asus-System-Product-Name:~/Desktop$
```

## Result:

Thus, the installation of C Compiler & execution of C Program is completed successfully.

<b>Exp. No: 3</b>	<b>Installation of Google App Engine &amp; Create “Hello world” App</b>
<b>Date:</b>	

**Aim:**

To Install Google App Engine and Create hello world app using python.

**Procedure:**

1. Install notepad++ editor or any other editor to write program.
2. Download latest version of Python for windows and Install it from  
<https://www.python.org/downloads/>
3. Download and install Google App Engine from  
<https://cloud.google.com/appengine/downloads?csw=1>  
 (Download the Cloud SDK installer for windows. )
4. Set the environment variable path to point to python & GAE installation directories as follows  
 Path= C:\Program Files\Python39\; C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin
5. Write a program for hello world in notepad++ editor as follows ( the indentations in each line of the code is important)

**Program 1: test.py**

```
import webapp2
class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.write("Hello World")
app=webapp2.WSGIApplication([('/', MainPage), ], debug=True)
```

**Program 2: app.yaml**

```
runtime: python27
api_version: 1
threadsafe: true
handlers:
- url: /
  script: test.app
```

6. Open google cloud SDK , then type the command **dev\_appserver.py** and specify the location of folder where the above two program files are saved as follows

C:\Program Files (x86)\Google\Cloud SDK>dev\_appserver.py  
"E:\OneDrive\RMDCSE\ACADEMIC\Laboratory\CS8711 CLOUD COMPUTING LAB\Lab-Programs\Expt-3"

7. A new window will get open if you are using Python for the first time make sure that you are selecting “Yes” and the press enter

```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.8_3.8.1520.0_x64_qbz5n2kfra8p0\python3.8.exe
Your current Cloud SDK version is: 307.0.0
Installing components from version: 307.0.0

These components will be installed:
Name          Version      Size
Cloud Datastore Emulator    2.1.0       18.4 MiB
gRPC python library        1.39.0       1.3 MiB
gcloud app Python Extensions 1.9.91      6.1 MiB

For the latest full release notes, please visit:
https://cloud.google.com/sdk/release_notes

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

Creating update staging area
Creating update staging area
Installing: Cloud Datastore Emulator
Installing: gRPC python library
Installing: gRPC python library
Installing: gRPC python library
Installing: gcloud app Python Extensions
Creating backup and activating new installation
```

### 8. On successful execution, you will get output as follows

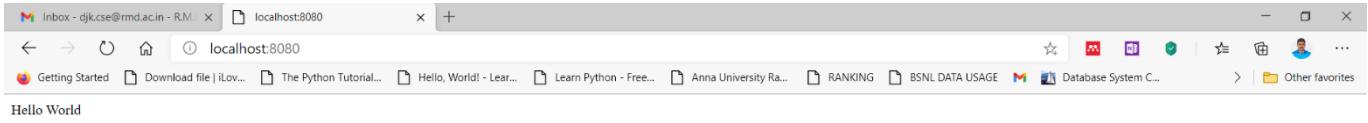
```
Google Cloud Shell - google-cloud-sdk\bin\dev_appserver.py "E:\OneDrive\RMDCSE\ACADEMIC\Laboratory\CS8711 CLOUD COMPUTING LAB\Lab-Programs\Expt-3"
C:\Program Files (x86)\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "E:\OneDrive\RMDCSE\ACADEMIC\Laboratory\CS8711 CLOUD COMPUTING LAB\Lab-Programs\Expt-3"

Updates are available for some Cloud SDK components. To install them,
please run:
$ gcloud components update

INFO 2020-12-11 08:38:15,453 devappserver2.py:289] Skipping SDK update check.
INFO 2020-12-11 08:38:17,767 api_server.py:282] Starting API server at: http://localhost:54686
INFO 2020-12-11 08:38:18,191 dispatcher.py:267] Starting module "default" running at: http://localhost:8080
INFO 2020-12-11 08:38:18,213 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO 2020-12-11 08:38:20,753 instance.py:294] Instance PID: 28272
```

You can see the Server is running at the local host like : <http://localhost:8080>

### 8. Open a web browser and type in the address bar as : http://localhost:8080



### Result:

Thus, the installation of Google App Engine and Creation of hello world app using python has been done successfully.

<b>Exp. No: 4</b>	<b>Use GAE launcher to launch the web applications</b>
<b>Date:</b>	

**Aim:**

To Use GAE launcher to launch the web application and deploy the web application in Google Cloud.

**Procedure:**

1. Install notepad++ editor or any other editor to write program.
2. Download latest version of Python for windows and Install it from

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

3. Download and install Google App Engine from  
<https://cloud.google.com/appengine/downloads?csw=1>

(Download the Cloud SDK installer for windows. )

4. Set the environment variable path to point to python & GAE installation directories as follows  
 Path= C:\Program Files\Python39\; C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin
5. Write a program for hello world in notepad++ editor as follows ( the indentations in each line of the code is important) and save it to desired location.

**Program 1: test.py**

```
import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):
        self.response.write("Hello World")

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

**Program 2: app.yaml**

```
runtime: python27

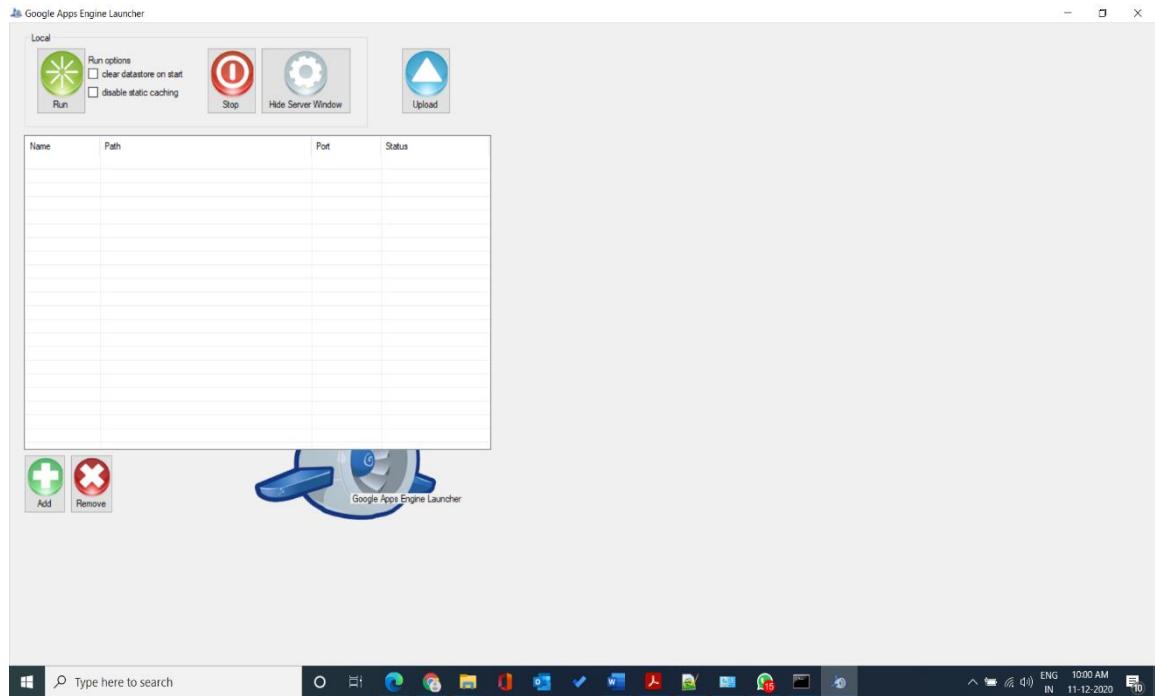
api_version: 1

threadsafe: true

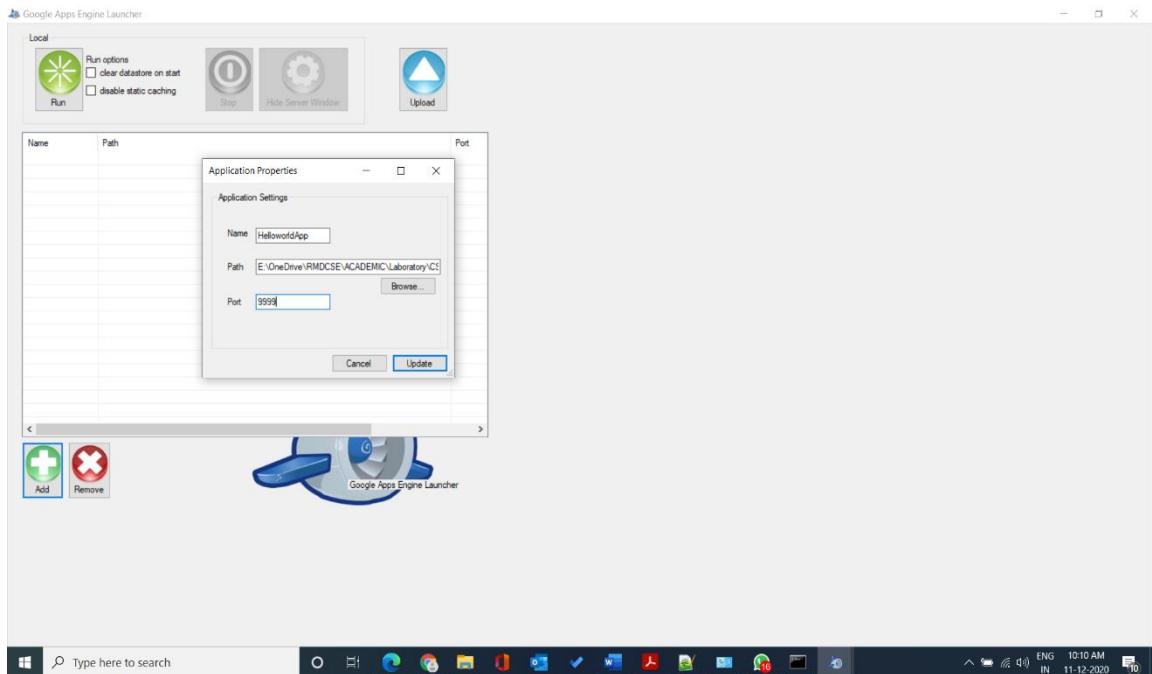
handlers:
- url: /
  script: test.app
```

6. Download and Install the GAE Launcher from  
[download GAE Launcher \(softpedia-secure-download.com\)](http://download.GAE.Launcher.softpedia-secure-download.com)

7. Open the downloaded GAE launcher application



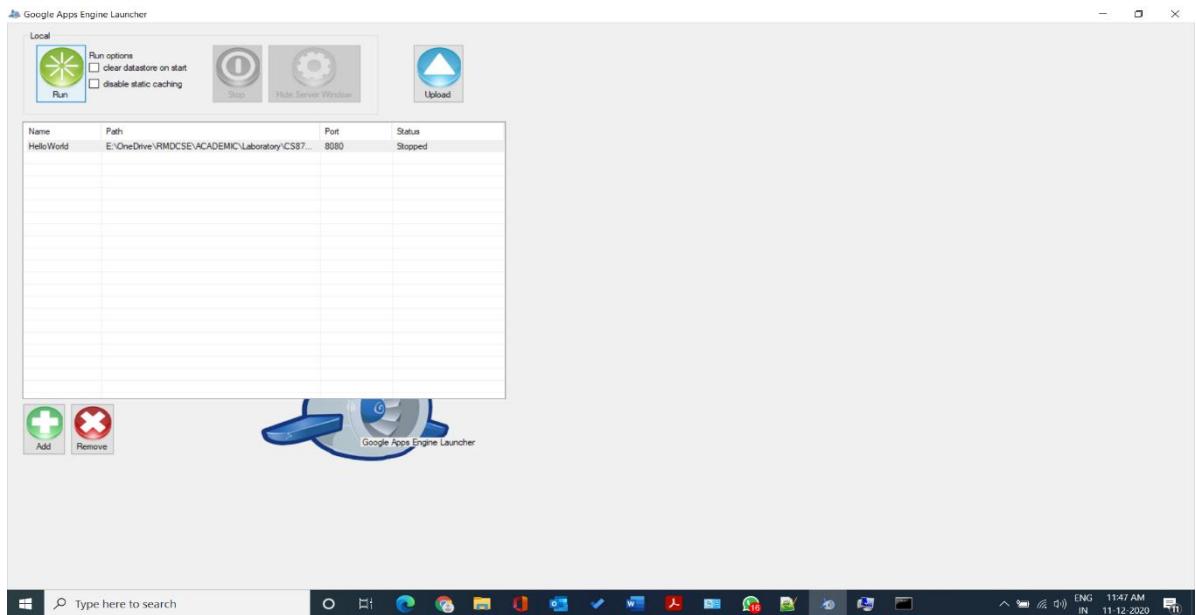
8. Use the Add(+) button to navigate into the directory where you saved your program files



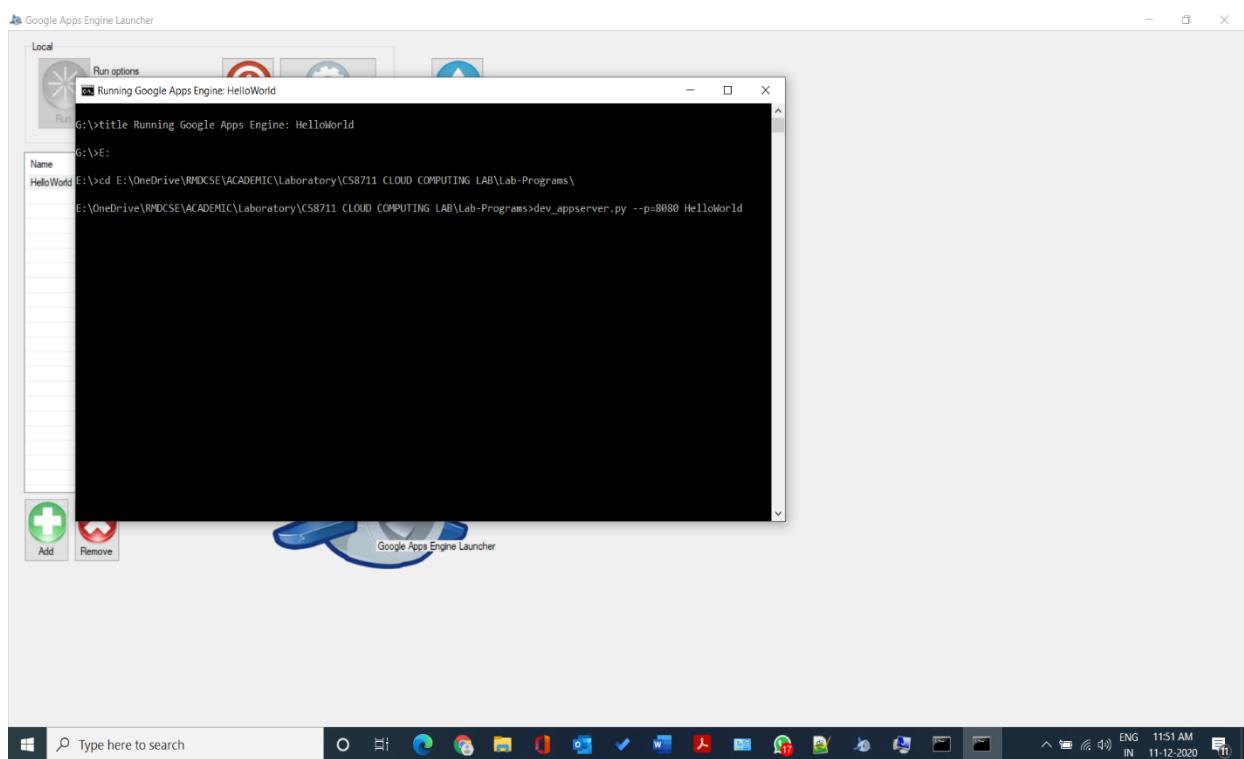
Name: User define name for the app

Path: location of the folder where you saved the program files

9. Select the project and click the run button



10. It will open the command prompt and start running the application as follows

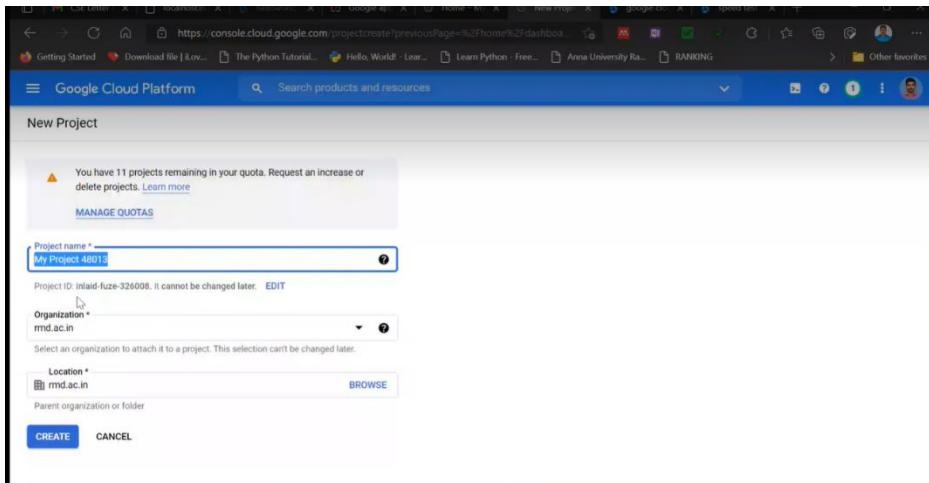


11. On successful execution and if no port conflicts, the application will be running at port number:8080.
12. Open the browser window and Enter <http://localhost:8080>.

It will display the output “Hello World”

## Deploying the web application in Google Cloud

- Go to browser and enter the console.google.com to create new project with unique id & copy the unique id for the project.



Open Google Cloud SDK command prompt and type the following commands

- To set the google account for deployment execute  
C:\Google\Cloud SDK> gcloud init ( do this if not executed it during installation ) and give the google account & create new project/select the project which is already created in previous step.

```
C:\Program Files (x86)\Google\Cloud SDK>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
accessibility:
  screen_reader: 'False'
core:
  account: djk.cs@rmd.ac.in
  disable_usage_reporting: 'False'
  project: molten-avenue-325919

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: 1

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for
this configuration:
[1] djk.cs@rmd.ac.in
[2] djk.cse@rmd.ac.in
[3] Log in with a new account
Please enter your numeric choice:
```

- Execute the command as below to deploy the application  
C:\Google\Cloud SDK> gcloud deploy app "application name"

```

Google Cloud Shell - gcloud app deploy "E:\Lab-Programs\Expt-3"
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the
https://console.developers.google.com/apis page.

Error creating a default .boto configuration file. Please run [gsutil config -n] if you would like to create this file.
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use djk.cs@rmd.ac.in by default
* Commands will reference project 'molten-avenue-325919' by default
Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accounts and/or projects.
Run `gcloud topic configurations` to learn more.

Some things to try next:
* Run `gcloud --help` to see the Cloud Platform services you can interact with. And run `gcloud help COMMAND` to get help on any gcloud command.
* Run `gcloud topic --help` to learn about advanced features of the SDK like arg files and output formatting

To take a quick anonymous survey, run:
$ gcloud survey

C:\Program Files (x86)\Google\Cloud SDK>gcloud app deploy "E:\Lab-Programs\Expt-3"
Services to deploy:
descriptor: [E:\Lab-Programs\Expt-3\app.yaml]
source: [E:\Lab-Programs\Expt-3]
target project: [molten-avenue-325919]
target service: [default]
target version: [20210914t142424]
target url: [https://molten-avenue-325919.de.r.appspot.com]
target service account: [App Engine default service account]

Do you want to continue (Y/n)? y
Beginning deployment of service [default]...

```

- From the above output copy the Target URL and paste in browser. The output will be displayed in the browser

## Result:

Thus launching of web applications using GAE launcher and deploying the application in cloud has been completed successfully.

<b>Exp. No: 5</b>	<b>Simulation of Cloud Scenario using Cloudsim</b>
<b>Date:</b>	

**Aim:**

To simulate the cloud scenario using cloudsim tool and run a scheduling algorithm.

**Procedure:****Simulation of cloud scenario:**

1. Go to <https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-3.0.3>
2. Download cloudsim-3.0.3.zip in this link

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is for the GitHub release of CloudSim 3.0.3. The page title is "cloudsim-3.0.3". It includes a release note by nikolayg, a changelog section titled "Changes from CloudSim 3.0.2 to CloudSim 3.0.3", and a "WHAT'S NEW" section. Below these are sections for "Assets" containing files like "cloudsim-3.0.3.tar.gz" and "cloudsim-3.0.3.zip". The browser's taskbar at the bottom shows other open tabs related to Google App Engine and Python tutorials.

3. Extract the downloaded zip file.

4. Have an eclipse IDE installed.

The screenshot shows a Microsoft Edge browser window displaying the Eclipse Technology download page. The main heading is "Download Eclipse Technology that is right for you". Below it, there's a section for "Get Eclipse IDE 2020-09" with a "Download x86\_64" button. To the right, there's a "Tool Platforms" section featuring "Eclipse Che" and "ORION". A message at the top states, "The Eclipse Installer 2020-09 R now includes a JRE for macOS, Windows and Linux." At the bottom, there's a cookie consent banner from the Eclipse Foundation.

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

6. Download commons-math3-3.6.1-bin.zip from the above link.

The screenshot shows a Microsoft Edge browser window with the Apache Commons Math download page. The URL in the address bar is [https://commons.apache.org/proper/commons-math/download\\_math.cgi](https://commons.apache.org/proper/commons-math/download_math.cgi). The page content includes a sidebar with links for Overview, Downloads, API docs, Javadoc, Issue Tracking, Source Repository, and User Guide. The main content area is titled "Download Apache Commons Math" and "Using a Mirror". It provides instructions on using a mirror for download integrity verification and lists download links for "Apache Commons Math 3.6.1 (requires Java 1.5+)". There are two tables: one for "Binaries" (tar.gz and zip files) and one for "Source" (tar.gz file). Each row in the tables includes SHA256 and PGP checksums.

File Type	Link	sha256	pgp
commons-math3-3.6.1-bin.tar.gz	<a href="#">Link</a>	sha256	pgp
commons-math3-3.6.1-bin.zip	<a href="#">Link</a>	sha256	pgp

File Type	Link	sha256	pgp
commons-math3-3.6.1-erc.tar.gz	<a href="#">Link</a>	sha256	pgp

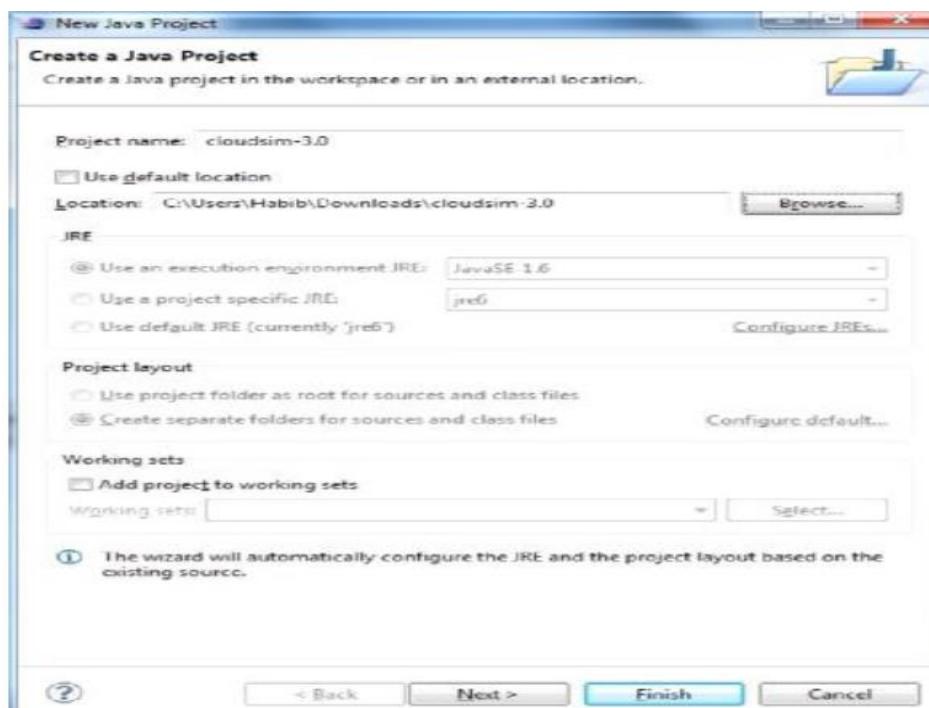
7. Extract the downloaded folder.

8. Open Eclipse SDK and select File--->New--->Java Project.

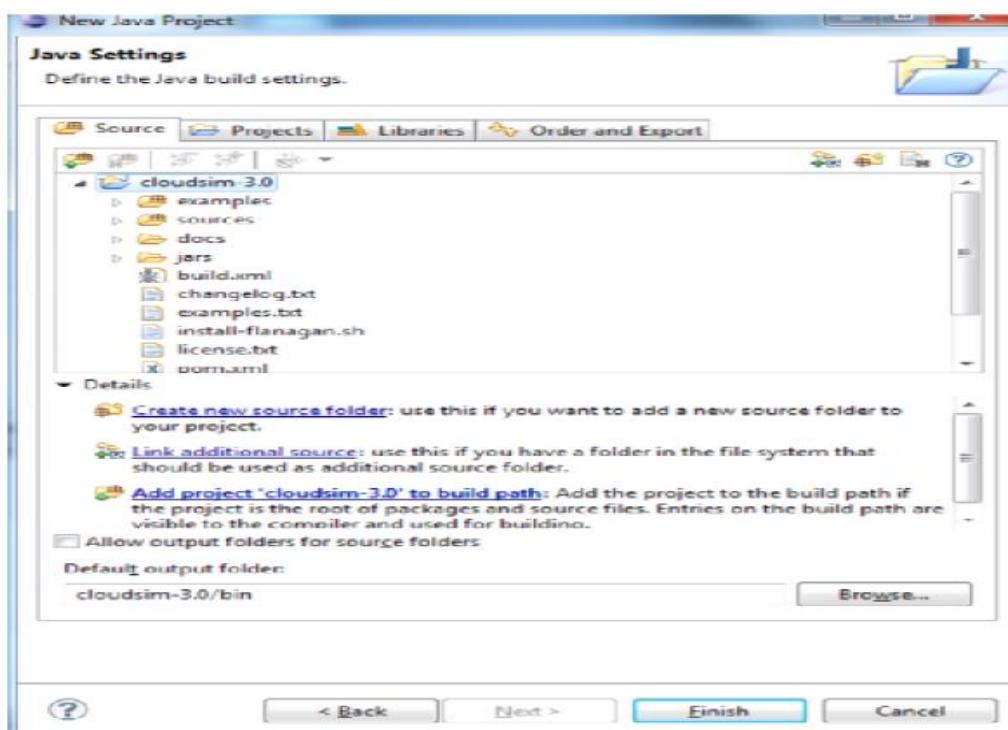
9. Give a project name( Ex: cloudsim-3.0) and then uncheck the Use Default Location checkbox.

10. Click on browse and select the extracted cloudsim-3.0.3 folder as location.

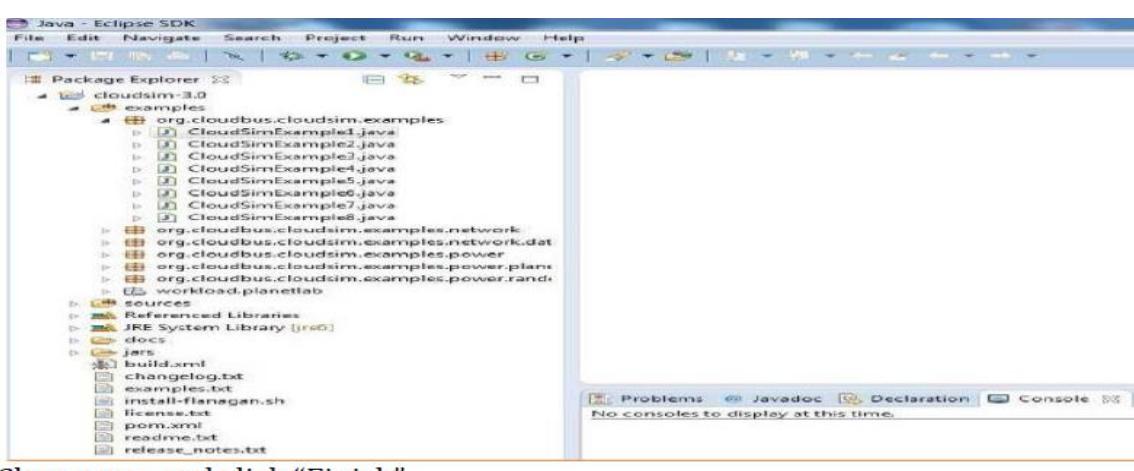
11. Select Next.

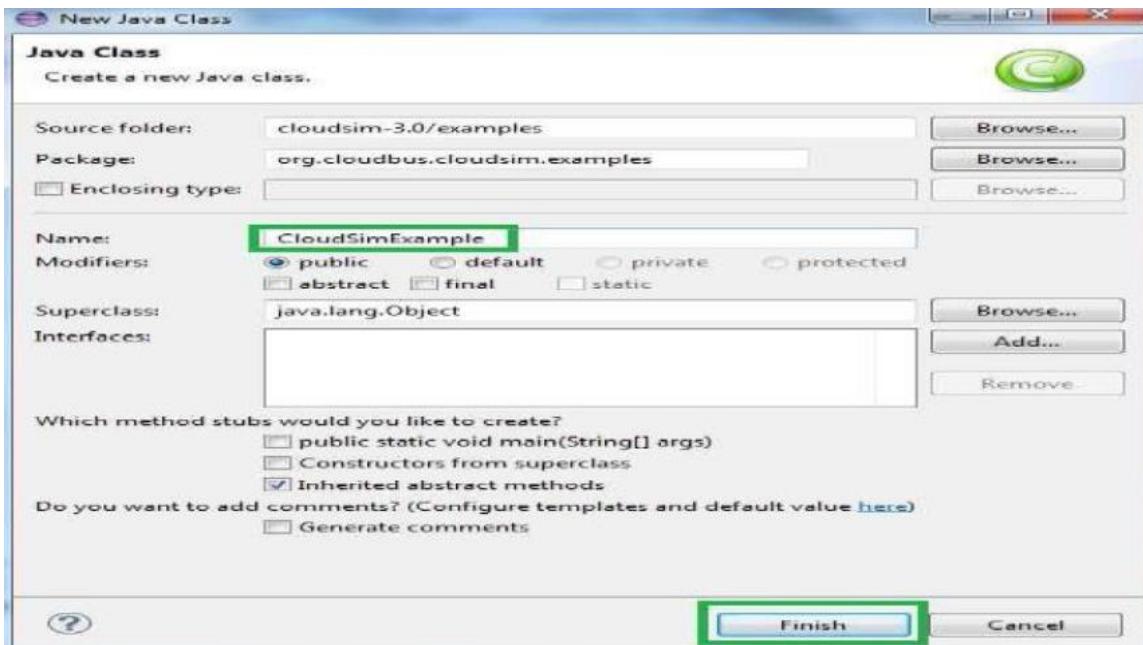


12. In the next window select the Libraries tab.
13. Click on add external JARs button and select commons-math3-3.6.1.jar file from the extracted commons-math3-3.6.1 folder.
14. The existing and added JAR files will be displayed in the libraries tab. Check if the selected file has been added and click finish.



15. Now if there are errors in the project, then right click on project name and select properties.
16. Select the java compiler option in the properties window. Enable the project specific settings checkbox. Change the compiler compliance level to 1.7 and allow rebuilding of the project.
17. Simulation Example: ( available in cloudsim package)
  - CloudSimExample1.java : shows how to create a datacenter with one host and run one cloudlet on it.
  - CloudSimExample2.java : shows how to create a datacenter with one host and run two cloudlets on it.
18. To create a new class just right click from “org.cloudbus.cloudsim.examples”, select “New” then “Class”

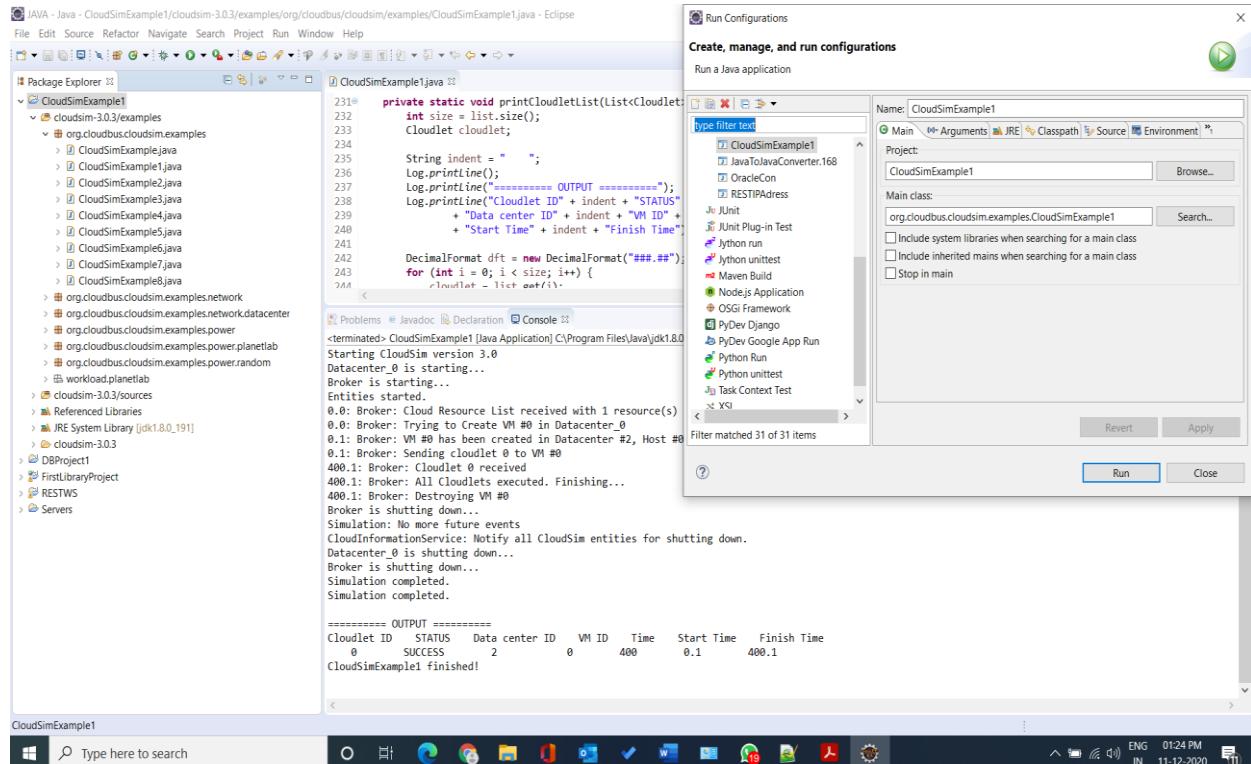




Write the code the application that you want to simulate

19. To run the example, select the project & click run configurations to select the desired class to execute. On successful execution, the output will be displayed in console window as bellow. Observe the output.

(Running CloudsimExample1 class exist in cloudsim package)



### **Simulation of scheduling algorithm in cloudsim:**

The steps to be followed are

1. Initialize the cloudSim package.
2. Create DataCenters to act as resource providers.
3. Create a data center broker. This will help in selecting a data center for usage.
4. Create a list of virtual machines to help in the execution of scheduling algorithm. Submit the list of virtual machines to the broker.
5. Create a list of cloudlet. A cloudlet specifies a set of user requests using an ID and also keeps track of the user to whom the responses has to be sent after processing the request. Submit the list of cloudlets to the broker. Call the required scheduling algorithm using the broker.
6. Now the tasks will get scheduled in the virtual machines.
7. Execute the tasks by starting simulation.
8. Print the results after execution.

Source code:

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

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;

import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

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

```

*/
public class CloudSimExample1 {

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

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

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

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

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

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

            // Second step: Create Datacenters
            // Datacenters are the resource providers in CloudSim. We need at
            // list one of them to run a CloudSim simulation
            Datacenter datacenter0 = createDatacenter("Datacenter_0");

            // Third step: Create Broker
            DatacenterBroker broker = createBroker();
            int brokerId = broker.getId();

            // Fourth step: Create one virtual machine
            vmlist = new ArrayList<Vm>();

            // VM description
            int vmid = 0;
            int mips = 1000;
            long size = 10000; // image size (MB)
            int ram = 512; // vm memory (MB)
            long bw = 1000;
            int pesNumber = 1; // number of cpus
            String vmm = "Xen"; // VMM name

            // create VM
            Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new
CloudletSchedulerTimeShared());
        }
    }
}

```

```

// add the VM to the vmList
vmList.add(vm);

// submit vm list to the broker
broker.submitVmList(vmList);

// Fifth step: Create one Cloudlet
cloudletList = new ArrayList<Cloudlet>();

// Cloudlet properties
int id = 0;
long length = 400000;
long fileSize = 300;
long outputSize = 300;
UtilizationModel utilizationModel = new UtilizationModelFull();

Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize,
utilizationModel, utilizationModel, utilizationModel);
cloudlet.setUserId(brokerId);
cloudlet.setVmId(vmid);

// add the cloudlet to the list
cloudletList.add(cloudlet);

// submit cloudlet list to the broker
broker.submitCloudletList(cloudletList);

// Sixth step: Starts the simulation
CloudSim.startSimulation();

CloudSim.stopSimulation();

//Final step: Print results when simulation is over
List<Cloudlet> newList = broker.getCloudletReceivedList();
printCloudletList(newList);

Log.println("CloudSimExample1 finished!");
} catch (Exception e) {
    e.printStackTrace();
    Log.println("Unwanted errors happen");
}
}

/**
 * Creates the datacenter.
 *
 * @param name the name
 *
 * @return the datacenter
 */
private static Datacenter createDatacenter(String name) {

// Here are the steps needed to create a PowerDatacenter:
// 1. We need to create a list to store

```

```

// our machine
List<Host> hostList = new ArrayList<Host>();

// 2. A Machine contains one or more PEs or CPUs/Cores.
// In this example, it will have only one core.
List<Pe> peList = new ArrayList<Pe>();

int mips = 1000;

// 3. Create PEs and add these into a list.
peList.add(new Pe(0, new PeProvisionerSimple(mips))); // need to store Pe id and MIPS
                                                       Rating

// 4. Create Host with its id and list of PEs and add them to the list
// of machines
int hostId = 0;
int ram = 2048; // host memory (MB)
long storage = 1000000; // host storage
int bw = 10000;

hostList.add(
    new Host(
        hostId,
        new RamProvisionerSimple(ram),
        new BwProvisionerSimple(bw),
        storage,
        peList,
        new VmSchedulerTimeShared(peList)
    )
); // This is our machine

// 5. Create a DatacenterCharacteristics object that stores the
// properties of a data center: architecture, OS, list of
// Machines, allocation policy: time- or space-shared, time zone
// and its price (G$/Pe time unit).
String arch = "x86"; // system architecture
String os = "Linux"; // operating system
String vmm = "Xen";
double time_zone = 10.0; // time zone this resource located
double cost = 3.0; // the cost of using processing in this resource
double costPerMem = 0.05; // the cost of using memory in this resource
double costPerStorage = 0.001; // the cost of using storage in this
                               // resource
double costPerBw = 0.0; // the cost of using bw in this resource
LinkedList<Storage> storageList = new LinkedList<Storage>(); // we are not adding SAN

// devices by now

DatacenterCharacteristics characteristics = new DatacenterCharacteristics(
    arch, os, vmm, hostList, time_zone, cost, costPerMem,
    costPerStorage, costPerBw);

// 6. Finally, we need to create a PowerDatacenter object.
Datacenter datacenter = null;

```

```

        try {
            datacenter = new Datacenter(name, characteristics, new
VmAllocationPolicySimple(hostList), storageList, 0);
        } catch (Exception e) {
            e.printStackTrace();
        }

        return datacenter;
    }

// We strongly encourage users to develop their own broker policies, to
// submit vms and cloudlets according
// to the specific rules of the simulated scenario
/**
 * Creates the broker.
 *
 * @return the datacenter broker
 */
private static DatacenterBroker createBroker() {
    DatacenterBroker broker = null;
    try {
        broker = new DatacenterBroker("Broker");
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
    return broker;
}

/**
 * Prints the Cloudlet objects.
 *
 * @param list list of Cloudlets
 */
private static void printCloudletList(List<Cloudlet> list) {
    int size = list.size();
    Cloudlet cloudlet;

    String indent = "    ";
    Log.printLine();
    Log.printLine("===== OUTPUT =====");
    Log.printLine("Cloudlet ID" + indent + "STATUS" + indent
                + "Data center ID" + indent + "VM ID" + indent + "Time" + indent
                + "Start Time" + indent + "Finish Time");

    DecimalFormat dft = new DecimalFormat("###.##");
    for (int i = 0; i < size; i++) {
        cloudlet = list.get(i);
        Log.print(indent + cloudlet.getCloudletId() + indent + indent);

        if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
            Log.print("SUCCESS");
            Log.printLine(indent + indent + cloudlet.getResourceId())
        }
    }
}

```

```

        + indent + indent + indent + cloudlet.getVmId()
        + indent + indent
        + dft.format(cloudlet.getActualCPUTime()) + indent
        + indent + dft.format(cloudlet.getExecStartTime())
        + indent + indent
        + dft.format(cloudlet.getFinishTime()));

    }

}

}

}

Output:
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

```

**Result:**

Thus the simulation of cloud scenario using cloudsim has been completed successfully.

<b>Exp. No: 6</b>	<b>Transfer the files from one Virtual Machine to another Virtual Machine</b>
<b>Date:</b>	

**Aim:**

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

**Procedure:*****Prerequisite:***

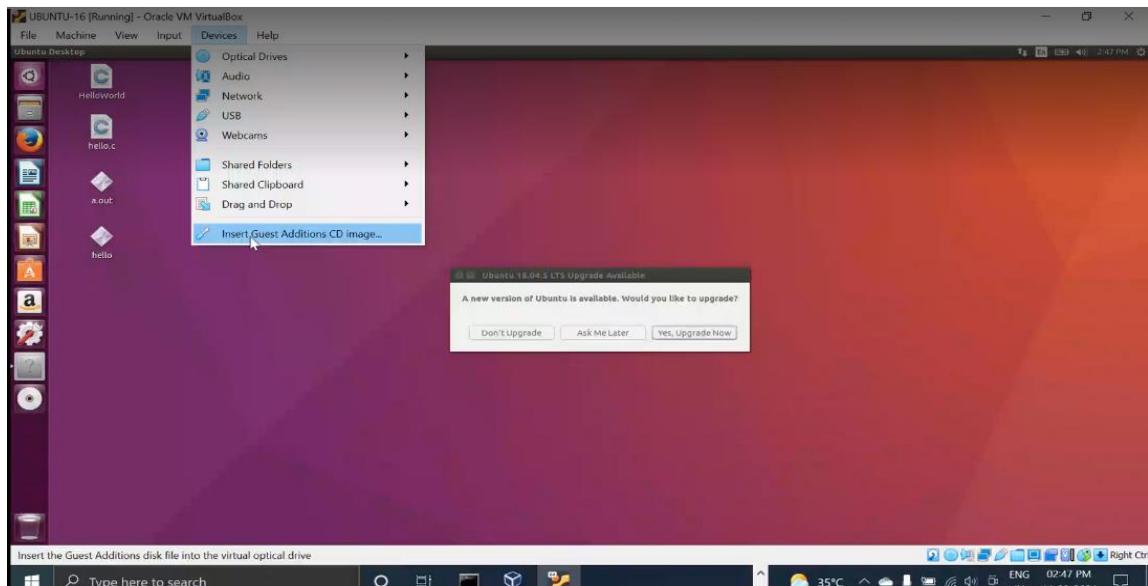
Install two instance of the virtual machine using Virtualbox and follow the methods given below for transferring files between virtual machines.

**Steps**

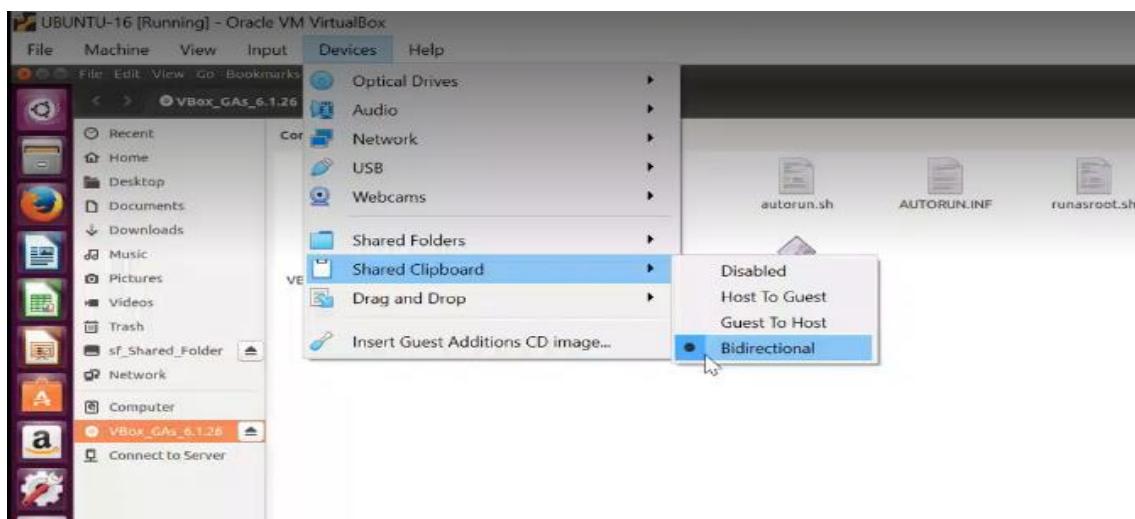
1. You can copy few (or more) lines with copy & paste mechanism.  
For this you need to share clipboard between host OS and guest OS, installing Guest Addition on both the virtual machines (probably setting bidirectional and restarting them).  
You copy from guest OS in the clipboard that is shared with the host OS.  
Then you paste from the host OS to the second guest OS.
2. You can enable drag and drop too with the same method (Click on the machine, settings, general, advanced, drag and drop: set to bidirectional )
3. You can have common Shared Folders on both virtual machines and use one of the directory shared as buffer to copy. Installing Guest Additions you have the possibility to set Shared Folders too. As you put a file in a shared folder from host OS or from guest OS, is immediately visible to the other.(Keep in mind that can arise some problems for date/time of the files when there are different clock settings on the different virtual machines).If you use the same folder shared on more machines you can exchange files directly copying them in this folder.

## Implementation:

1. Select the "Insert Guest Additions CD image" option from the devices.



2. VBox\_GAs software will be downloaded then select Bidirectional from "Shared Clipboard" and "Drag and Drop" options.



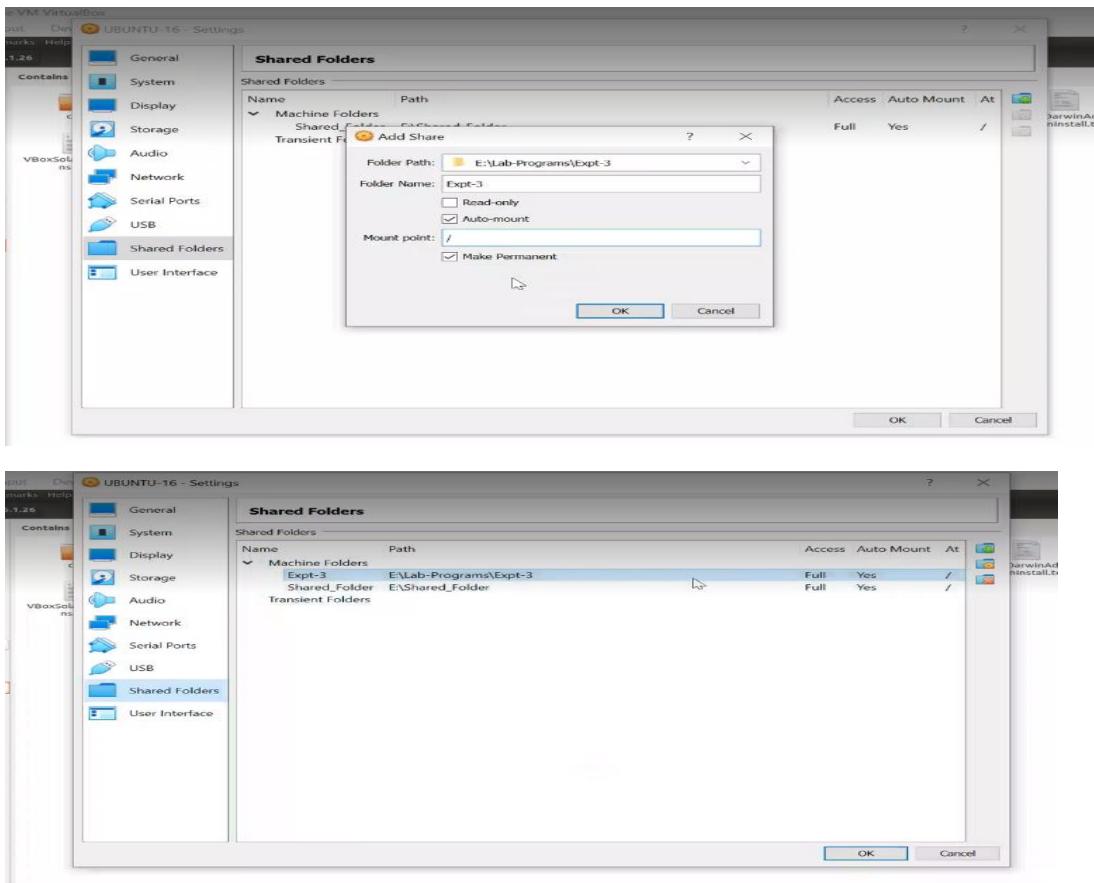
3. Click on devices then select

Shared Folder -> shared folder settings.

4. Now click add new shared folder icon in the right.

5. In the Folder path select the folder from your host machine which you want to share and fill the folder name Eg.SharedFolder.

6. Check the auto-mount option and mention the mount point as "/" and click OK.

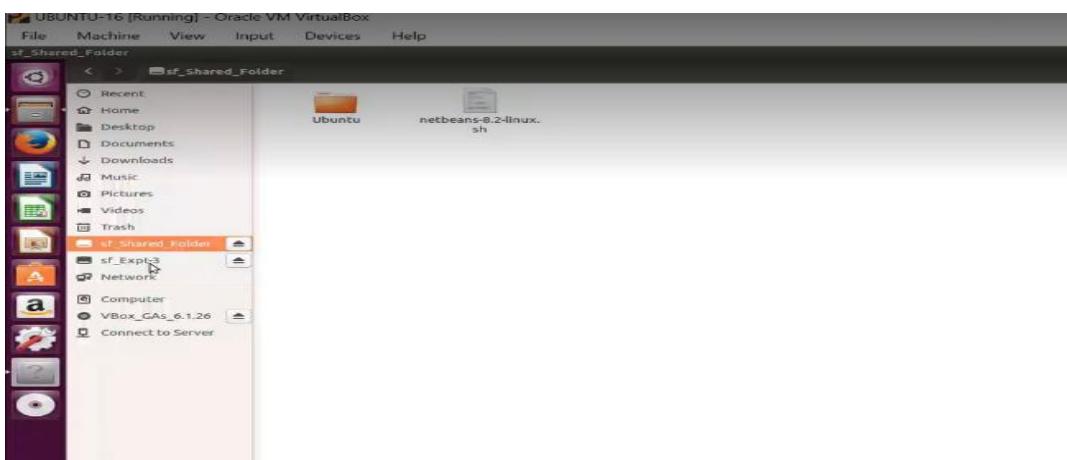


7. Now open the files from your virtual machine you can see the folder you created at step 3 (SharedFolder). To get the access for the shared folder run the following command in the terminal

"sudo adduser <username> vboxsf"

8. Create a text file with some content in the shared folder from your host machine.

9. You can see the created text file in the virtual machine. Edit the text file in your host machine or another virtual machine and the changes will be reflected in your virtual machine shared folder.



Similarly you can drag and drop any file between the virtual machines.

#### **Result:**

Thus, the procedure for file transfer between virtual machines is tested successfully

<b>Exp. No: 7</b>	<b>Find a procedure to launch virtual machine using trystack</b>
<b>Date:</b>	

**Aim:**

To find a procedure to launch virtual machine using trystack /Openstack.

**Procedure:**

**Step 1:** Open the Ubuntu via VirtualBox and open the terminal for the installation of OpenStack.

**Step 2:** Before starting installation process we must update and upgrade our system using the following commands.

**\$ sudo apt-get update**

**\$ sudo apt-get upgrade**

```
Activities Terminal Nov 20 12:17 guest@guest-VirtualBox: ~
guest@guest-VirtualBox:~$ sudo apt-get update
[sudo] password for guest:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 114 kB in 2s (64.9 kB/s)
Reading package lists... Done
Rhythmbox -VirtualBox:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  libsmbclient libwbclient0 samba-libs
The following packages will be upgraded:
  accountsservice alsa-ucm-conf apport apport-gtk bind9-dnsutils bind9-host
  bind9-libs ca-certificates cpio distro-info-data firefox ghostscript
  ghostscript-x gir1.2-accountsservice-1.0 gir1.2-javascriptcoregtk-4.0
  gir1.2-mutter-6 gir1.2-udisks-2.0 gir1.2-webkit2-4.0 gnome-control-center
  gnome-control-center-data gnome-control-center-faces
  gnome-shell-extension-desktop-icons libaccountsservice0 libasound2
  libasound2-data libatopology2 libcaca0 libcurl3-gnutls libcurl4
  libdrm-amdgpu1 libdrm-common libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libdrm2 libegl-mesa0 libgbm1 libgcrypt20 libgd3 libgl1-mesa-dri
```

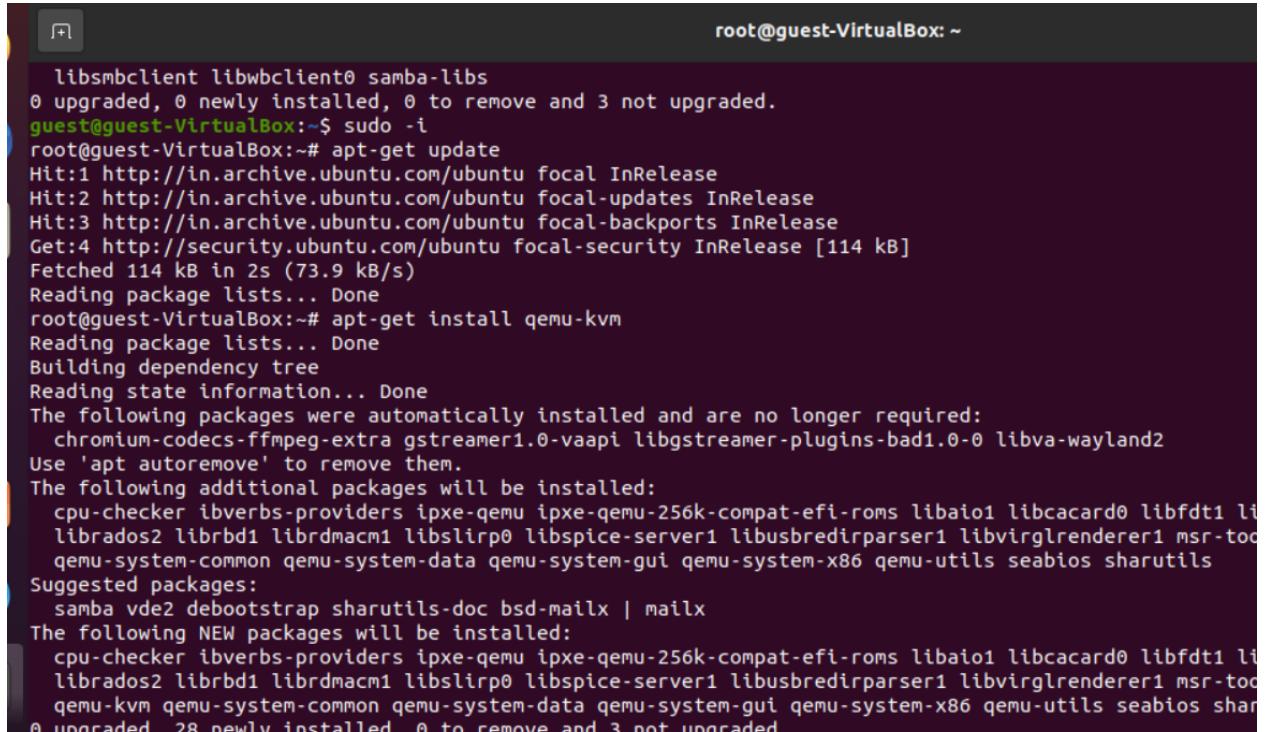
**Step 3:** After updating and upgrading the system, follow the below commands for the OpenStack installation process.

**\$ sudo -i**

**# apt-get update**

**# apt-get install qemu-kvm**

**Do you want to continue [Y/N]? Y**



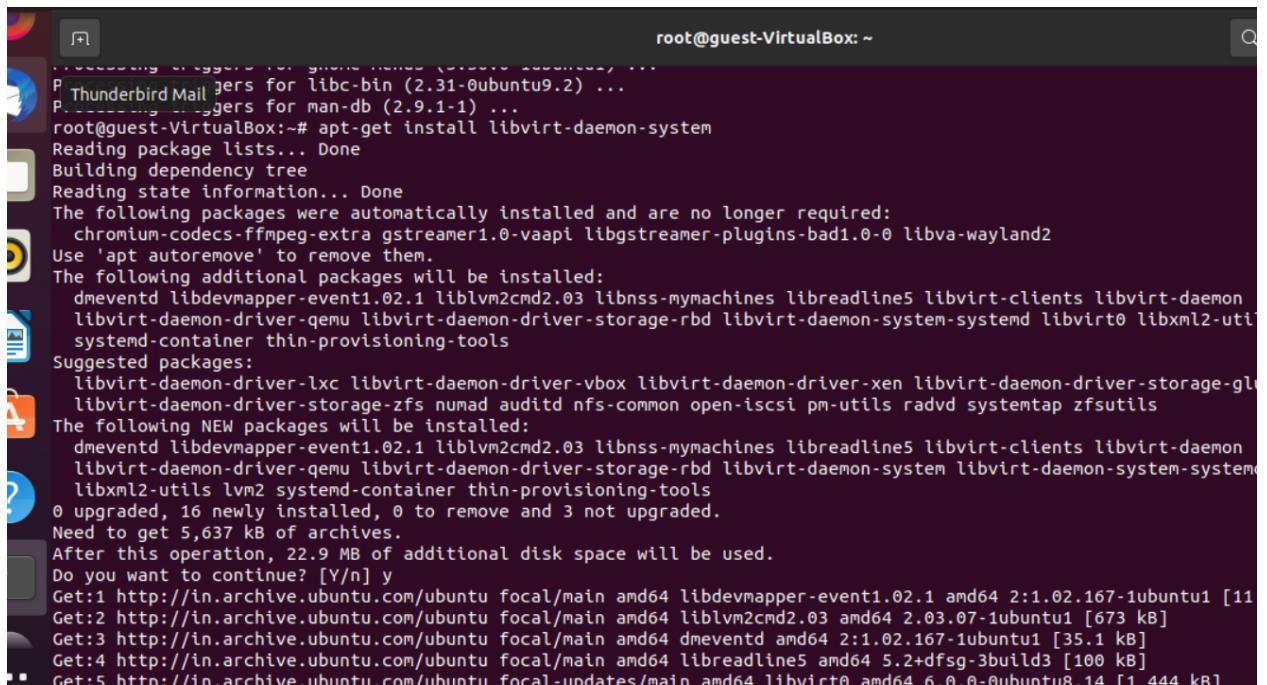
```

root@guest-VirtualBox: ~
libsmbclient libwbclient0 samba-libs
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
guest@guest-VirtualBox:~$ sudo -i
root@guest-VirtualBox:~# apt-get update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 114 kB in 2s (73.9 kB/s)
Reading package lists... Done
root@guest-VirtualBox:~# apt-get install qemu-kvm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaio1 libcacard0 libfdt1 lib
librados2 librbd1 librdmacm1 libslirp0 libspice-server1 libusbredirparser1 libvirglrenderer1 msr-to
qemu-system-common qemu-system-data qemu-system-gui qemu-system-x86 qemu-utils seabios sharutils
Suggested packages:
samba vde2 debootstrap sharutils-doc bsd-mailx | mailx
The following NEW packages will be installed:
cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaio1 libcacard0 libfdt1 lib
librados2 librbd1 librdmacm1 libslirp0 libspice-server1 libusbredirparser1 libvirglrenderer1 msr-to
qemu-kvm qemu-system-common qemu-system-data qemu-system-gui qemu-system-x86 qemu-utils seabios shar
0 upgraded, 28 newly installed, 0 to remove and 3 not upgraded.

```

**# apt-get install libvirt-daemon-system && apt-get install libvirt-clients**

**Do you want to continue [Y/N]? Y**



```

root@guest-VirtualBox: ~
Processing triggers for gnome-themes-standard (3.30.0-1ubuntu1) ...
P Thunderbird Mail triggers for libc-bin (2.31-0ubuntu9.2) ...
P Thunderbird Mail triggers for man-db (2.9.1-1) ...
root@guest-VirtualBox:~# apt-get install libvirt-daemon-system
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
dmeventd libdevmapper-event1.02.1 liblvm2cmd2.03 libnss-mymachines libreadline5 libvirt-clients libvirt-daemon
libvirt-daemon-driver-qemu libvirt-daemon-storage-rbd libvirt-daemon-system-systemd libvirt0 libxml2-util
systemd-container thin-provisioning-tools
Suggested packages:
libvirt-daemon-driver-lxc libvirt-daemon-driver-vbox libvirt-daemon-driver-xen libvirt-daemon-driver-storage-gl
libvirt-daemon-driver-storage-zfs numad auditd nfs-common open-iscsi pm-utils radvd systemtap zfsutils
The following NEW packages will be installed:
dmeventd libdevmapper-event1.02.1 liblvm2cmd2.03 libnss-mymachines libreadline5 libvirt-clients libvirt-daemon
libvirt-daemon-driver-qemu libvirt-daemon-driver-storage-rbd libvirt-daemon-system libvirt-daemon-system-systemd
libxml2-utils lvm2 systemd-container thin-provisioning-tools
0 upgraded, 16 newly installed, 0 to remove and 3 not upgraded.
Need to get 5,637 kB of archives.
After this operation, 22.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libdevmapper-event1.02.1 amd64 2:1.02.167-1ubuntu1 [11
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 liblvm2cmd2.03 amd64 2.03.07-1ubuntu1 [673 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/main amd64 dmeventd amd64 2:1.02.167-1ubuntu1 [35.1 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libreadline5 amd64 5.2+dfsg-3build3 [100 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libvirt0 amd64 6.0.0-0ubuntu8.14 [1,444 kB]

```

**# apt-get install bridge-utils**

```
[+] root@guest-VirtualBox: ~
Building dependency tree
Reading state information... Done
libvirt-clients is already the newest version (6.0.0-0ubuntu8.14).
libvirt-clients set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@guest-VirtualBox:~# apt-get install bridge-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
Suggested packages:
  ifupdown
The following NEW packages will be installed:
  bridge-utils
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 30.5 kB of archives.
After this operation, 112 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Fetched 30.5 kB in 0s (103 kB/s)
Selecting previously unselected package bridge-utils.
(Reading database ... 185697 files and directories currently installed.)
Preparing to unpack .../bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
Setting up bridge-utils (1.6-2ubuntu1) ...
```

# apt-get install virt-manager

Do you want to continue [Y/N]? Y

```
Setting up bridge-utils (1.6-2ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
root@guest-VirtualBox:~# apt-get install virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0 gir1.2-spiceclientglib-2.0
  gir1.2-spiceclientgtk-3.0 libgovirt-common libgovirt2 libgtk-vnc-2.0-0 libgvnc-1.0-0 libosinfo-1.0-0 libphodav-2.0-0
  libphodav-2.0-common libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5 libusbredirhost1 libvirt-glib-1.0-0 osinfo-db
  python3-distutils python3-libvirt python3-libxml2 spice-client-glib-usb-acl-helper virt-viewer virtinst
Suggested packages:
  libosinfo-l10n gstreamer1.0-plugins-bad python3-guestfs ssh-askpass
The following NEW packages will be installed:
  gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0 gir1.2-spiceclientglib-2.0
  gir1.2-spiceclientgtk-3.0 libgovirt-common libgovirt2 libgtk-vnc-2.0-0 libgvnc-1.0-0 libosinfo-1.0-0 libphodav-2.0-0
  libphodav-2.0-common libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5 libusbredirhost1 libvirt-glib-1.0-0 osinfo-db
  python3-distutils python3-libvirt python3-libxml2 spice-client-glib-usb-acl-helper virt-manager virt-viewer virtinst
0 upgraded, 25 newly installed, 0 to remove and 3 not upgraded.
Need to get 2,626 kB of archives.
After this operation, 18.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 gir1.2-appindicator3-0.1 amd64 12.10.1+20.04.20200408.1-0ubuntu1 []
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 libgvnc-1.0-0 amd64 1.0.0-1build1 [56.4 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 libgtk-vnc-2.0-0 amd64 1.0.0-1build1 [25.9 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 gir1.2-gtk-vnc-2.0 amd64 1.0.0-1build1 [10.3 kB]
```

# apt-get install qemu-system

Do you want to continue [Y/N]? Y

```

Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
root@guest-VirtualBox:~# apt-get install qemu-system
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  qemu-efi-aarch64 qemu-efi-arm qemu-slof qemu-system-arm qemu-system-mips qemu-system-misc qemu-sy
  qemu-system-sparc
Suggested packages:
  qemu samba vde2 openbios-ppc openhwware openbios-sparc
The following NEW packages will be installed:
  qemu-efi-aarch64 qemu-efi-arm qemu-slof qemu-system qemu-system-arm qemu-system-mips qemu-system-
  qemu-system-s390x qemu-system-sparc
0 upgraded, 10 newly installed, 0 to remove and 3 not upgraded.
Need to get 81.8 MB of archives.
After this operation, 640 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 qemu-system-arm amd64 1:4.2-3ubu
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 qemu-system-mips amd64 1:4.2-3ubu
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 qemu-system-mipsel i386 1:4.2-3ubu
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main armhf qemu-system-mipsel armhf 1:4.2-3ubu
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main arm64 qemu-system-mipsel arm64 1:4.2-3ubu
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main ppc64el qemu-system-mipsel ppc64el 1:4.2-3ubu
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main s390x qemu-system-mipsel s390x 1:4.2-3ubu
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main sparc64 qemu-system-mipsel sparc64 1:4.2-3ubu
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main aarch64 qemu-system-mipsel aarch64 1:4.2-3ubu
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/main armv7hl qemu-system-mipsel armv7hl 1:4.2-3ubu

```

```
# virsh --connect qemu:///system list --all
```

```
# logout
```

**Step 4:** Create a new user named stack and give permission to start the OpenStack installation.

```
$ sudo adduser stack
```

Is the information correct [Y/N]? Y

```
$ sudo -i
```

```

Setting up qemu-system (2:4.2-3ubuntu1.10) ...
Processing triggers for man-db (2.9.1-1) ...
root@guest-VirtualBox:~# virsh --connect qemu:///system list --all
  Id  Name   State
  ----
root@guest-VirtualBox:~# logout
guest@guest-VirtualBox:~$ sudo adduser stack
Adding user `stack' ...
Adding new group `stack' (1001) ...
Adding new user `stack' (1001) with group `stack' ...
Creating home directory `/home/stack' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for stack
Enter the new value, or press ENTER for the default
  Help  Full Name []:
          Room Number []:
          Work Phone []:
          Home Phone []:
          Other []:
Is the information correct? [Y/n] y
guest@guest-VirtualBox:~$ sudo -i

```

```
# echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

```
# apt-get install sudo -y || yum install -y sudo
```

```
# echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
# su - stack
```

**Step 4: Download the devstack from github.com.**

```
$ sudo apt-get install git
```

Do you want to continue [Y/N]? Y

```
Is the information correct? [Y/n] y
guest@guest-VirtualBox:~$ sudo -i
root@guest-VirtualBox:~# echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
root@guest-VirtualBox:~# apt-get install sudo -y || yum install -y sudo
Reading package lists... Done
Building dependency tree
Reading state information... Done
sudo is already the newest version (1.8.31-1ubuntu1.2).
sudo set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@guest-VirtualBox:~# echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
root@guest-VirtualBox:~# su - stack
stack@guest-VirtualBox:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git
The following NEW packages will be installed:
  git git-man liberror-perl
```

```
$ git clone https://git.openstack.org/openstack-dev/devstack
```

```
Setting up git-man (1:2.25.1-1ubuntu3.2) ...
Setting up git (1:2.25.1-1ubuntu3.2) ...
Processing triggers for man-db (2.9.1-1) ...
stack@guest-VirtualBox:~$ git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
warning: redirecting to https://opendev.org/openstack/devstack/
remote: Enumerating objects: 27708, done.
remote: Counting objects: 100% (27708/27708), done.
remote: Compressing objects: 100% (9182/9182), done.
remote: Total 47917 (delta 27053), reused 18526 (delta 18526), pack-reused 20209
Receiving objects: 100% (47917/47917), 10.20 MiB | 1.16 MiB/s, done.
Resolving deltas: 100% (33667/33667), done.
```

**Step 5:** Run the following commands to avoids errors before installation.

```
$ sudo rm /var/lib/dpkg/lock
```

```
$ sudo rm -rf /var/lib/apt/lists/lock
```

```
$ sudo rm /var/cache/apt/archives/lock
```

### Step 6: Configure the local.conf file.

```
$ su - stack
$ cd devstack
$ ls
$ cd samples
$ ls
$ cp local.conf ..
$ cd ..
$ ls
```

```
stack@guest-VirtualBox:~/devstack/samples$ su - stack
Password:
stack@guest-VirtualBox:~$ cd devstack
$ Help guest-VirtualBox:~/devstack$ ls
clean.sh doc functions gate lib openrc roles setup.cfg stack.sh tox.ini
CONTRIBUTING.rst extras.d functions-common HACKING.rst LICENSE playbooks run_tests.sh setup.py tests unstack.sh
data files FUTURE.rst inc Makefile README.rst samples stackrc tools
stack@guest-VirtualBox:~/devstack$ cd samples
stack@guest-VirtualBox:~/devstack/samples$ ls
local.conf local.sh
stack@guest-VirtualBox:~/devstack/samples$ cp local.conf ../
stack@guest-VirtualBox:~/devstack/samples$ cd ..
stack@guest-VirtualBox:~/devstack$ ls
clean.sh doc functions gate lib Makefile README.rst samples stackrc tools
CONTRIBUTING.rst extras.d functions-common HACKING.rst LICENSE openrc roles setup.cfg stack.sh tox.ini
data files FUTURE.rst inc local.conf playbooks run_tests.sh setup.py tests unstack.sh
stack@guest-VirtualBox:~/devstack$ nano local.conf
```

\$ nano local.conf

**ADMIN\_PASSWORD=secret**

**DATABASE\_PASSWORD=secret**

**RABBIT\_PASSWORD=secret**

**SERVICE\_PASSWORD=secret**

**HOST\_IP=10.0.2.15**

**FLOATING\_RANGE=10.0.2.224/27**

```
GNU nano 4.8                                     local.conf

# Minimal Contents
# -----
#
# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:
#
# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh`` and they will be added to ``local.conf``.
ADMIN_PASSWORD=password
DATABASE_PASSWORD=password
RABBIT_PASSWORD=password
SERVICE_PASSWORD=password

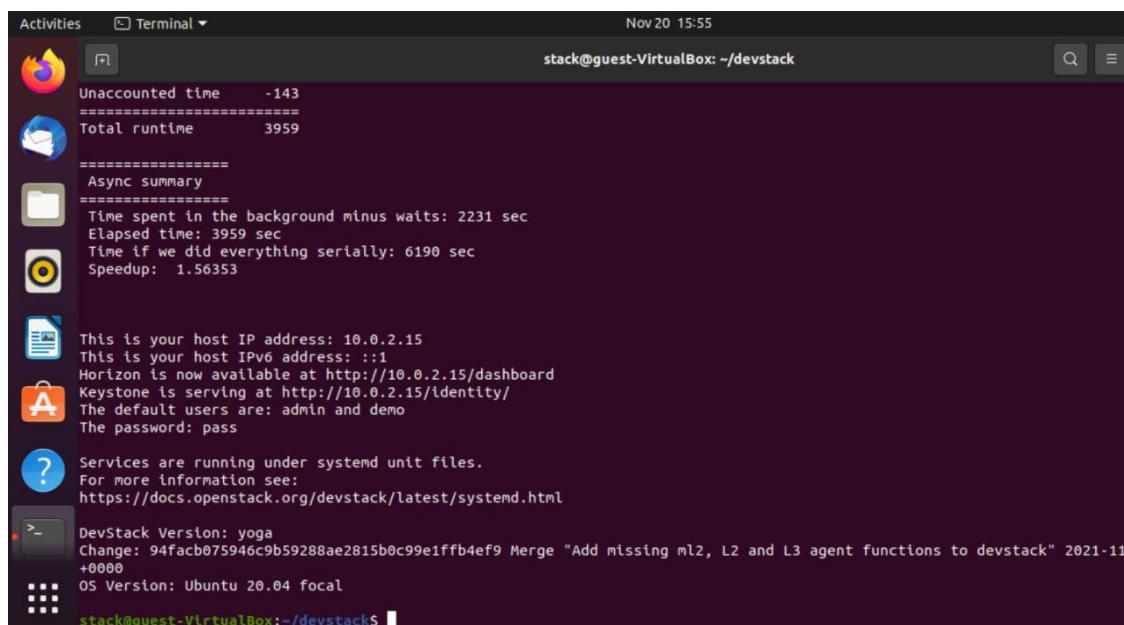
HOST_IP=10.0.2.15
FLOATING_RANGE=10.0.2.224/27
#
# Help ``ST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# one NIC configuration of the host is unusual, i.e. ``eth1`` has the default
# route but ``eth0`` is the public interface. They are auto-detected in
# ``stack.sh`` but often is indeterminate on later runs due to the IP moving
# from an Ethernet interface to a bridge on the host. Setting it here also
# makes it available for ``openrc`` to include when setting ``OS_AUTH_URL``.
# Neither is set by default.
#HOST_IP=w.x.y.z
#HOST_IPV6=2001:db8::7
```

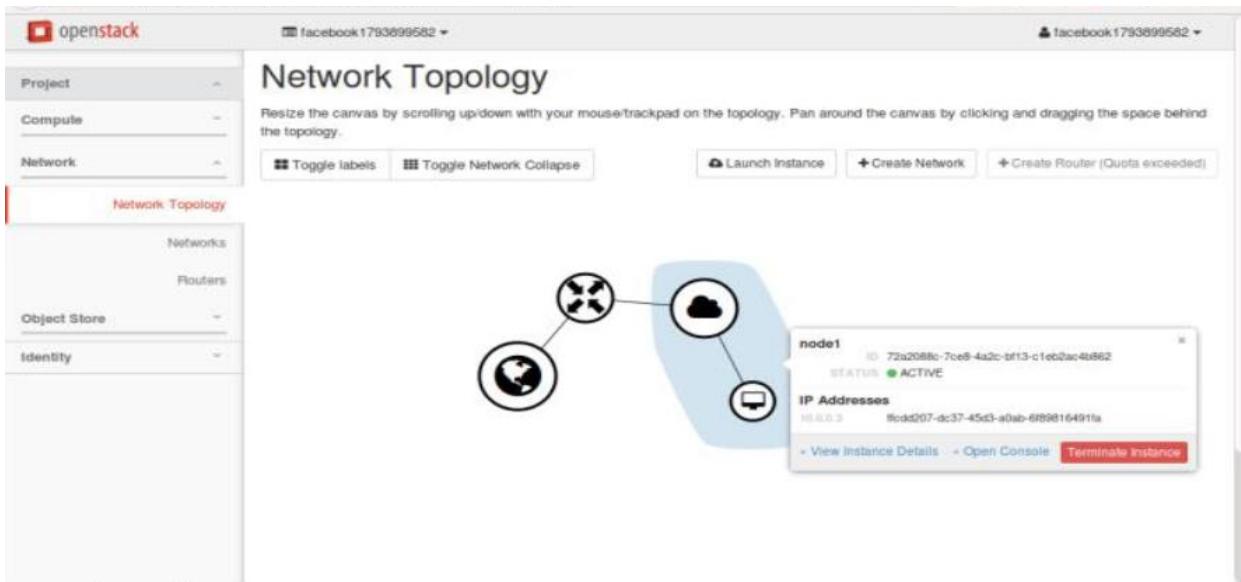
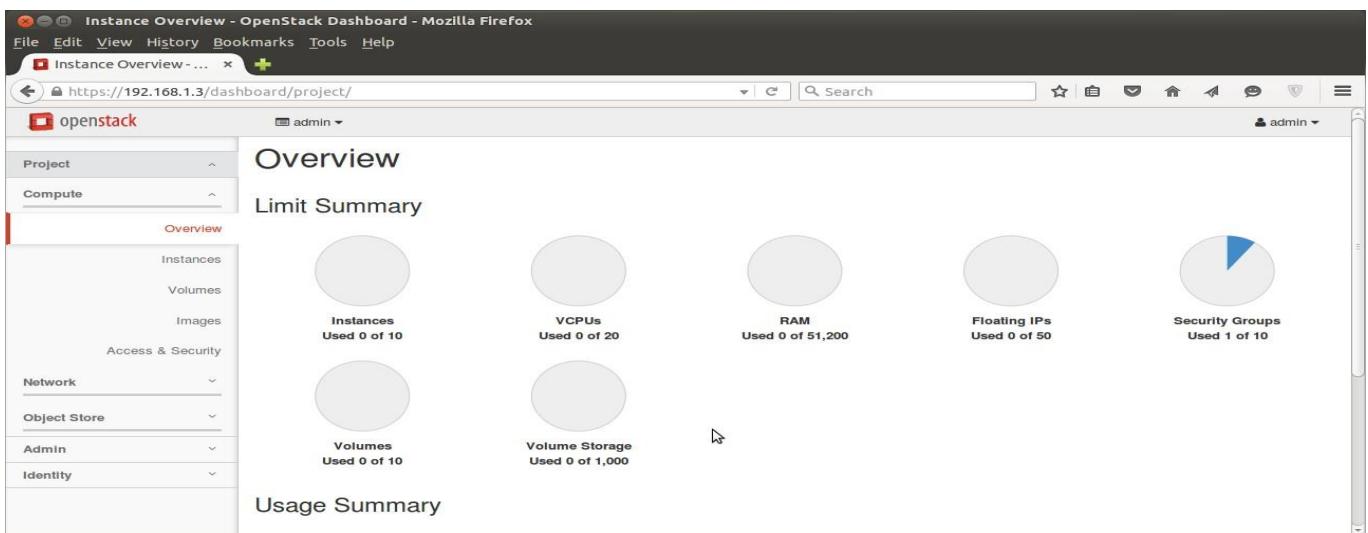
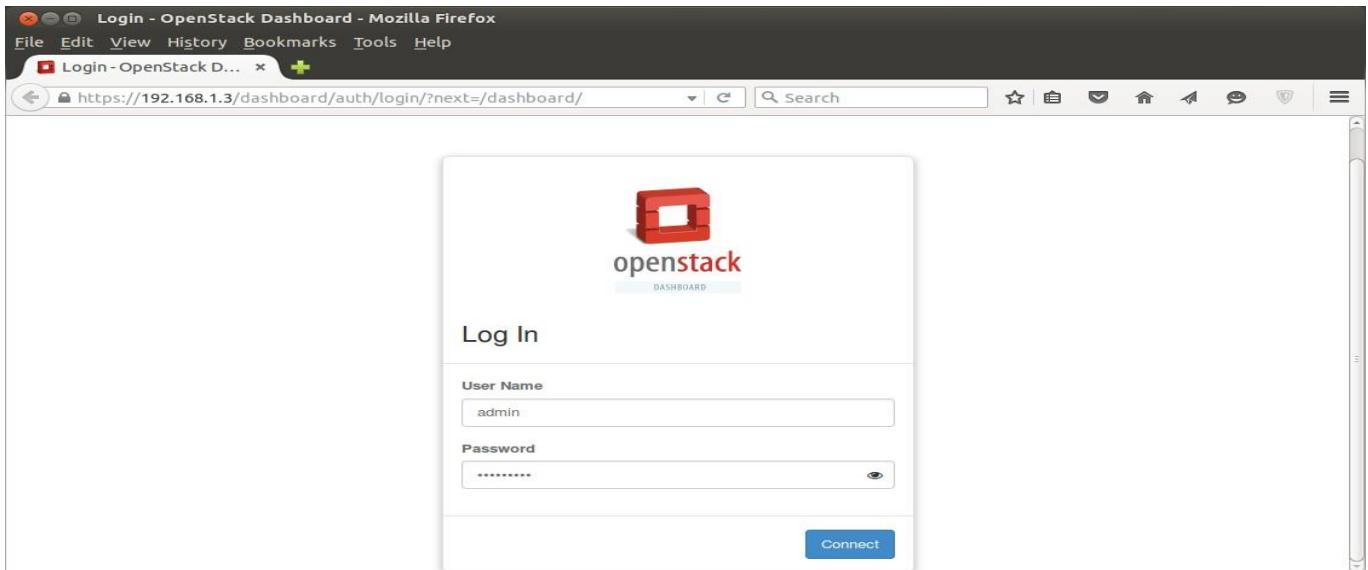
**Step 7:** Now, we can start the installation process of OpenStack using the below command.

```
$ ./stack.sh
```

```
stack@guest-VirtualBox:~/devstack$ nano local.conf
stack@guest-VirtualBox:~/devstack$ ./stack.sh
+ unset GREP_OPTIONS
+ unset LANG
+ unset LANGUAGE
+ LC_ALL=en_US.utf8
+ export LC_ALL
++ grep -E '^OS_'
++ cut -d = -f 1
++ env
+ unset
+ umask 022
+ Help=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
/sbin
+++ dirname ./stack.sh
++ cd .
++ pwd
+ TOP_DIR=/home/stack/devstack
+ NOUNSET=
+ [[ -n '' ]]
++ date +%s
+ DEVSTACK_START_TIME=1637399900
```

**Step 8:** After successful installation, go to <http://localhost:8080> and give the username and password as provided in the local.conf file.





Network topology

As you see from the image above, the instance will be connected to a local network and the local network will be connected to internet.

### Step 1: Create Network

Network? Yes, the network in here is our own local network. So, your instances will be not mixed up with the others. You can imagine this as your own LAN (Local Area Network) in the cloud.

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,
  1. Fill **Network Address** with appropriate CIDR, for example 192.168.1.0/24. Use **private network CIDR block** as the best practice.
  2. Select **IP Version** with appropriate IP version, in this case IPv4.
  3. 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,
  1. Fill **Instance Name**, for example Ubuntu 1.
  2. Select **Flavor**, for example m1.medium.
  3. Fill **Instance Count** with 1.
  4. Select **Instance Boot Source** with **Boot from Image**.
  5. 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,
  1. 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.
  2. In **Import Key Pair** dialog,
    1. Fill **Key Pair Name** with your machine name (for example Edward-Key).
    2. 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.
    3. Click **Import key pair**.
  3. In **Security Groups**, mark/check **default**.
  4. In **Networking** tab,
    1. 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

I guess you already know what router is. 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:

1. Select **External networks** with **external**.
2. Then **OK**.
5. Click **Add Interface** button.
  1. Select **Subnet** with the network that you have been created in Step 1.
  2. 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 are 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`.

#### **Result:**

Thus, the virtual machine is launched by using openstack by following the procedure successfully.

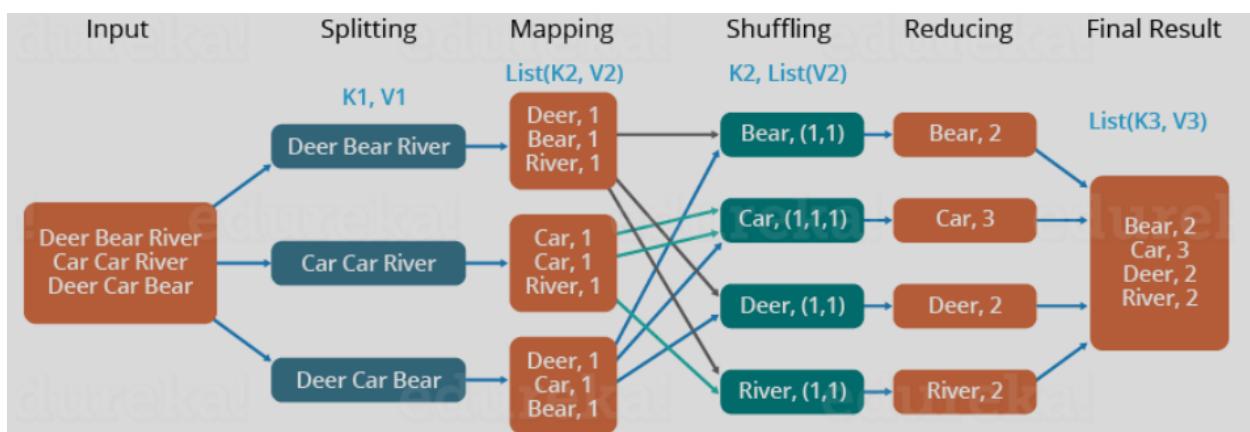
<b>Exp. No: 8</b>	<b>Install Hadoop single node cluster and run word count program</b>
<b>Date:</b>	

**Aim:**

To Install Hadoop single node cluster and run word count program using Mapreduce algorithm.

**Procedure:****MapReduce Word Count**

In MapReduce word count example, we find out the frequency of each word. Here, the role of Mapper is to map the keys to the existing values and the role of Reducer is to aggregate the keys of common values. So, everything is represented in the form of Key-value pair.

**Dear, Bear, River, Car, Car, River, Deer, Car and Bear**

- A. Java Installation
- B. SSH Installation
- C. Hadoop installation

**A) Commands for Java Installation:**

```
$ sudo apt update // Update Command
$ sudo apt install openjdk-8-jdk -y // installing JDK
// verifying JDK installation commands
$ java -version
$ javac -version
```

**B) Commands for SSH Installation:**

```
$ sudo apt update // Update Command
$ sudo apt install openssh-server -y // installing OpenSSH server command
$ sudo apt install openssh-client -y // installing OpenSSH client command
```

### C) Commands for Hadoop Installation:

// Setting up a non-root user for hadoop commands

```
$ sudo adduser hadoop
```

```
$ sudo usermod -aG sudo hadoop
```

```
$ su – hadoop
```

// Passwordless SSH settings for hadoop commands

```
$ ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
```

```
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

```
$ chmod 0600 ~/.ssh/authorized_keys
```

```
$ ssh localhost
```

```
$ exit
```

```
$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz
```

// command for downloading hadoop

```
$ tar xzf hadoop-3.2.2.tar.gz // untar the file using "tar" command
```

```
$ ls // checking the contents in the directory
```

```
$ cd /home/hadoop/hadoop-3.2.2 // change the current directory
```

// Hadoop Installation and Configuration commands

```
$ sudo vi ~/.bashrc
```

#Hadoop Related Options

```
export HADOOP_HOME=/home/hadoop/hadoop-3.2.2
```

```
export HADOOP_INSTALL=$HADOOP_HOME
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_HOME=$HADOOP_HOME
```

```
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

```
export YARN_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

```
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

```
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

// Press esc → : → wq

```
$ source ~/.bashrc
```

```
$ cd /home/hadoop/hadoop-3.2.2 // change the current directory
```

```
$ which javac
```

```

$ readlink -f /usr/bin/javac

$ sudo vi etc/hadoop/hadoop-env.sh

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

// Press esc → : → wq

$ cd /home/hadoop/hadoop-3.2.2 // change the current directory

$ mkdir tmpdata

$ sudo vi etc/hadoop/core-site.xml

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/hadoop/hadoop-3.2.2/tmpdata</value>

</property>

<property>

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

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>

// Press esc → : → wq

$ cd home/hadoop/hadoop-3.2.2 // change the current directory

$ mkdir -p dfsdata/namenode

$ mkdir -p dfsdata/datanode

$ sudo vi etc/hadoop/hdfs-site.xml

<configuration>

<property>

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

<value>/home/hadoop/hadoop-3.2.2/dfsdata/namenode</value>

</property>

<property>

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

<value>/home/hadoop/hadoop-3.2.2/dfsdata/datanode</value>

</property>

<property>

```

```

<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
// Press esc → : → wq

$ cd /home/hadoop/hadoop-3.2.2 // change the current directory
$ sudo vi etc/hadoop/mapred-site.xml

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
// Press esc → : → wq

$ cd /home/hadoop/hadoop-3.2.2 // change the current directory
$ sudo vi etc/hadoop/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>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>127.0.0.1</value>
</property>
<property>
<name>yarn.acl.enable</name>
<value>0</value>

```

```
</property>

<property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PER_PEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>

// Press esc → : → wq

$ cd /home/hadoop/hadoop-3.2.2 // change the current directory

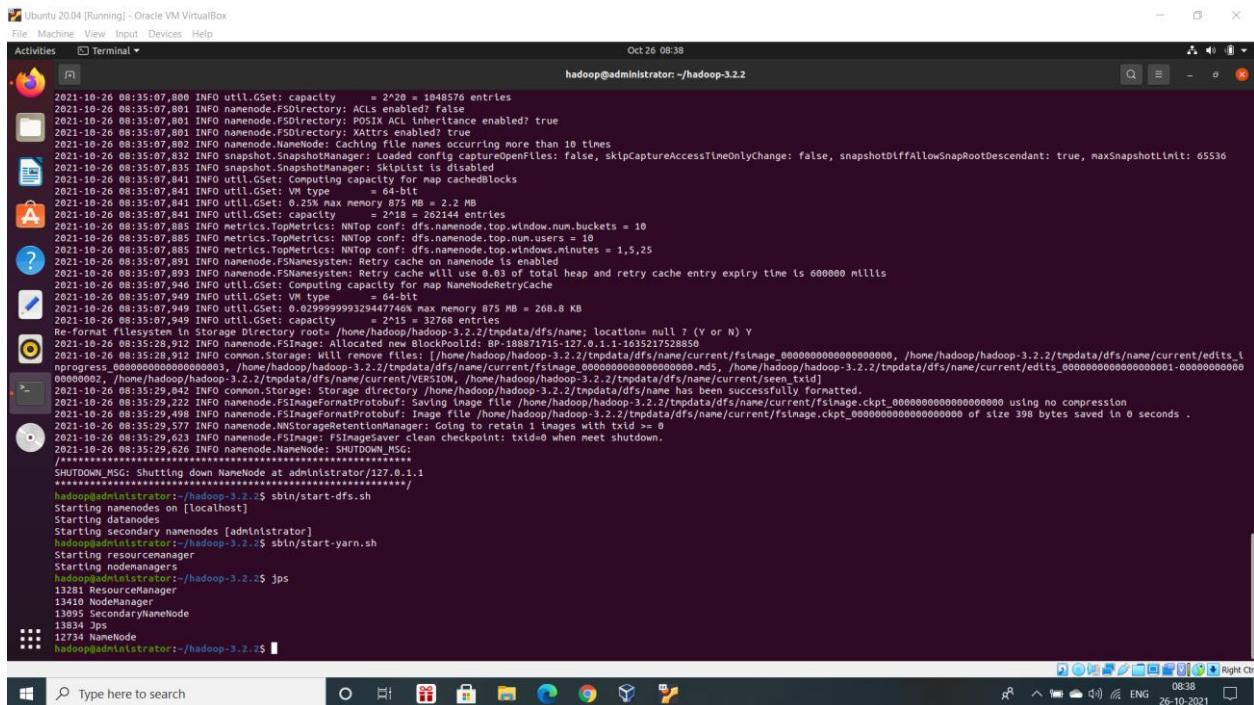
$ hdfs namenode -format // formatting hdfs namenode

$ sbin/start-dfs.sh // start hadoop cluster

$ sbin/start-yarn.sh // start yarn resource and node manager

$ jps // checking if all daemons are active and running

// Accessing Hadoop UI from browser
```



**localhost:9870** // Namenode Interface

The screenshot shows the HDFS Health Overview page for the Namenode. It displays basic system information such as start time, version, and cluster ID. Below this is a summary section showing configured capacity (28.91 GB), remote capacity (0 B), and usage details (DFS Used: 24 KB (0%), Non DFS Used: 13.12 GB, DFS Remaining: 14.3 GB (49.47%)).

## localhost:9864 // Individual Datanodes

The screenshot shows the DataNode information page. It displays the Cluster ID and Version. Below this is a Block Pools section showing a single pool for the Namenode address (localhost:9000). The Volume Information section shows a single disk directory (/home/hadoop/hadoop-3.2.2/dfsdata/datanode) with 24 KB used, 14.3 GB left, and 0 B reserved. The status bar indicates the date and time as Oct 26 14:55 and 26-10-2021.

## localhost:8088 // Yarn Resource Manager

The screenshot shows the YARN Resource Manager All Applications page. It displays cluster metrics like Apps Submitted (0), Apps Pending (0), and Apps Running (0). The Scheduler Metrics section shows the Capacity Scheduler with no data available in the table. The status bar indicates the date and time as Oct 26 14:55 and 26-10-2021.

### Steps to execute MapReduce Word Count example:

**Step 1:** Login to hadoop user account in Ubuntu and Open the terminal and follow the below steps.

```
$ hadoop version // check the version of hadoop
$ javac -version // check the version of JDK
$ cd /home/hadoop/hadoop-3.2.2 // change the current directory
$ sbin/start-dfs.sh // start hadoop cluster
$ sbin/start-yarn.sh // start yarn resource and node manager
$ jps // checking if all daemons are active and running
// Accessing Hadoop UI from browser
localhost:9870 // Namenode Interface
localhost:9864 // Individual Datanodes
localhost:8088 // Yarn Resource Manager
$ cd .. // change the current directory
// setting classpath for hadoop
$ export HADOOP_CLASSPATH=$(hadoop classpath)
$ echo $ HADOOP_CLASSPATH
```

**Step 2:** Create a folder in the desktop as **WordCountTutorial**. Inside **WordCountTutorial** folder, place a java program for word count as **WordCount.java**.

**Step 3:** Source code link: <https://www.dropbox.com/s/yp9i7nwmgr3nkx/WordCount.java?dl=0>.

### WordCount.java Program:

```
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);

}

}

```

**Step 4:** Create a folder inside **WordCountTutorial** as **input\_data** folder.

**Step 5:** Inside **input\_data** folder, place an input file as **input.txt**.

**Step 6:** Type a few names in the file **input.txt** and save.

**Step 7:** Create an empty folder inside **WordCountTutorial** as **tutorial\_classes**.

**Step 8:** Follow the below commands to create folders in Hadoop MapReduce.

```
$ hadoop fs -mkdir /WordCountTutorial // creating WordCountTutorial folder in hadoop
```

```
$ hadoop fs -mkdir /WordCountTutorial/Input // creating Input folder inside WordCountTutorial folder
```

```
$ hadoop fs -put '/home/hadoop/Desktop/WordCountTutorial/input_data/input.txt'
/WordCountTutorial/Input // putting files from desktop to hadoop
```

```
$ cd /home/hadoop/Desktop/WordCountTutorial // change the current directory
```

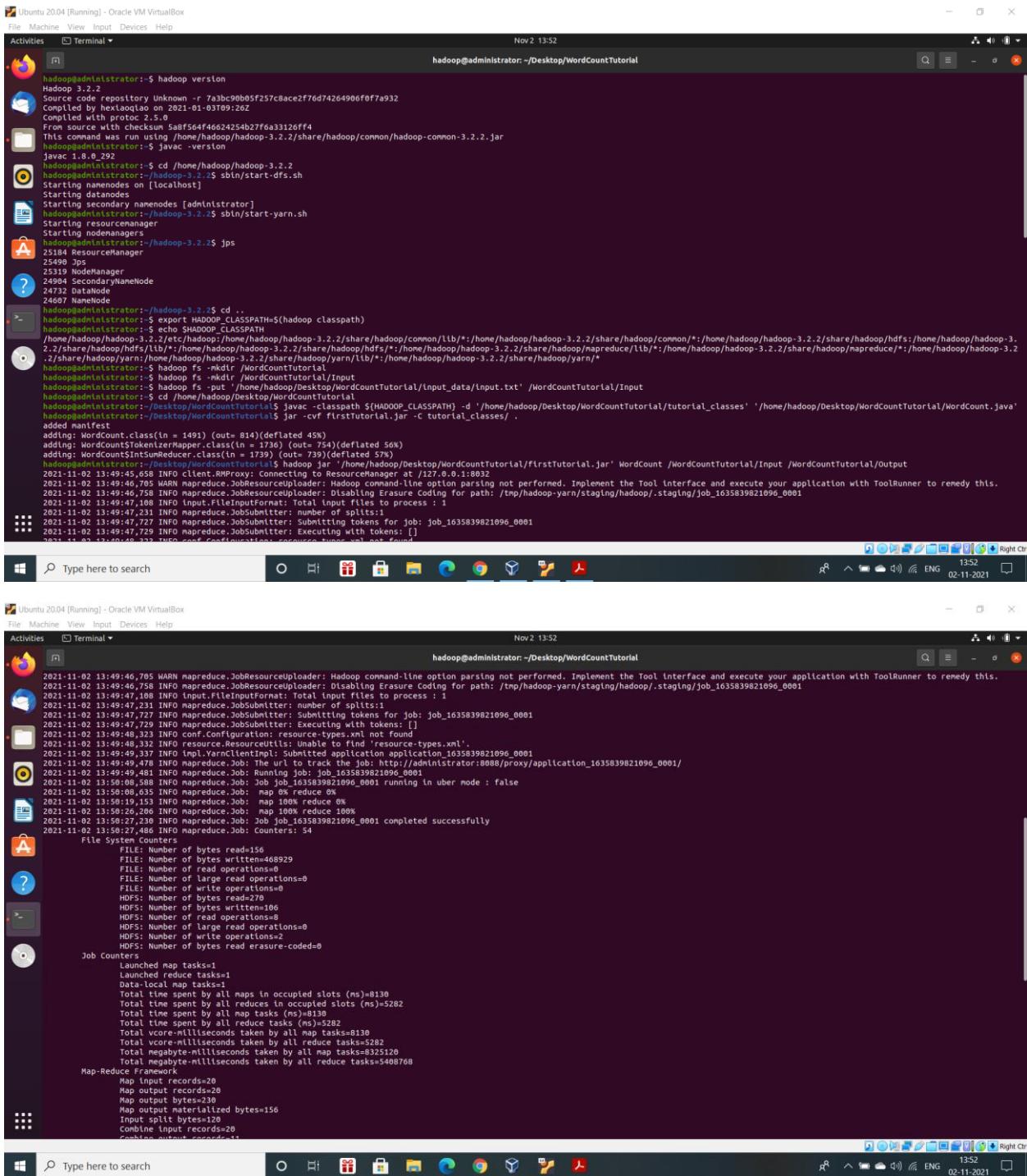
**Step 9:** Compile the WordCount.java program using the following command.

```
$ javac -classpath ${HADOOP_CLASSPATH} -d '/home/hadoop/Desktop/WordCountTutorial/tutorialclasses'
'/home/hadoop/Desktop/WordCountTutorial/WordCount.java' // compiling the java code WordCount.java
and thus, creating classes in the tutorial_classes folder
```

**Step 10:** Create a jar file and compile the jar file in Hadoop using the below commands.

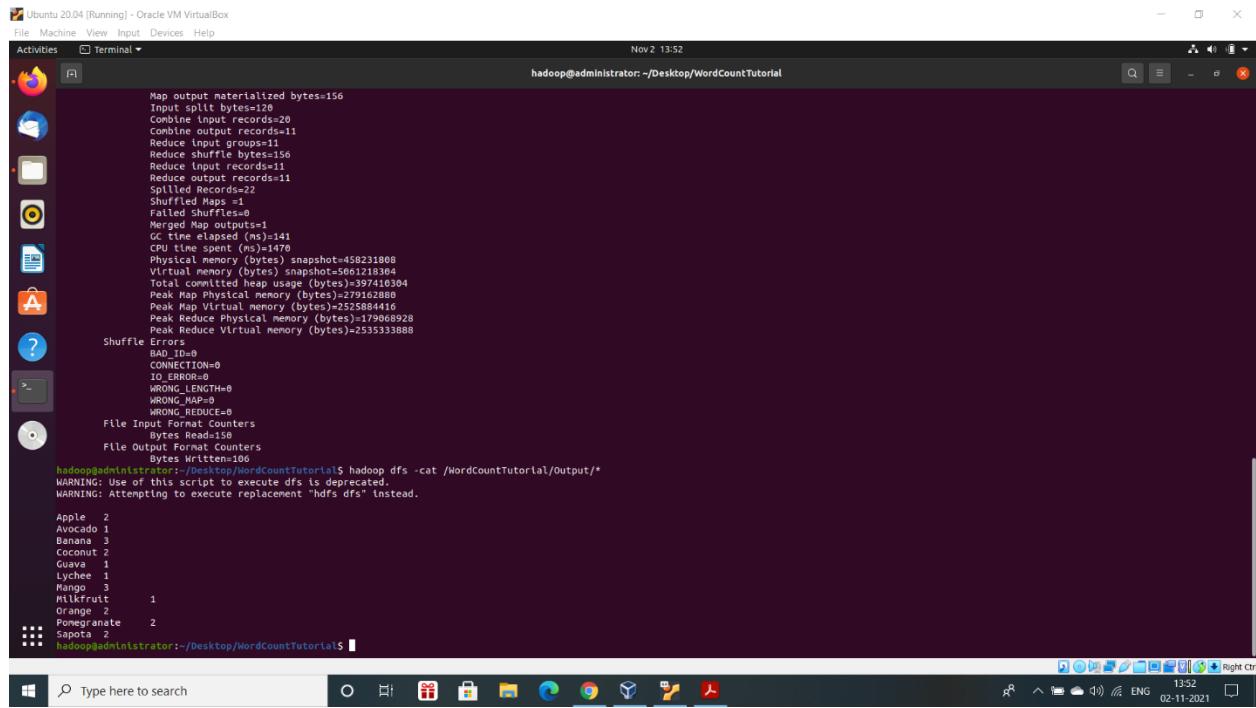
```
$ jar -cvf firstTutorial.jar -C tutorial_classes/. // create a jar folder as first_Tutorial.jar and copy the class files
from tutorial_classes to it
```

```
$ hadoop jar '/home/hadoop/Desktop/WordCountTutorial/firstTutorial.jar' WordCount
/WordCountTutorial/Input /WordCountTutorial/Output // compile the jar file in Hadoop
```



**Step 11:** Display the output using the below command.

```
$ hadoop dfs -cat /WordCountTutorial/Output/* // display the output
```



```

Map output materialized bytes=156
Input split bytes=120
Combine input records=20
Combine output records=11
Reduce input records=11
Reduce shuffle bytes=156
Reduce input records=11
Reduce output records=11
Spilled Records=22
Shuffled Maps =1
Failed Maps=0
Mapper Map outputs=1
GC time elapsed (ms)=141
CPU time spent (ms)=1470
Physical memory (bytes) snapshot=459231808
Virtual memory (bytes) snapshot=5061218364
Total memory (bytes)=5061218364
Peak Map Physical memory (bytes)=279162880
Peak Map Virtual memory (bytes)=2525884416
Peak Reduce Physical memory (bytes)=179066928
Peak Reduce Virtual memory (bytes)=2535333888
Shuffle Errors
  Bad Src=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=156
File Output Format Counters
  Bytes Written=106
hadoop@administrator:/Desktop/WordCountTutorial$ hadoop dfs -cat /WordCountTutorial/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

Apple 2
Avocado 1
Banana 3
Coconut 2
Dragon 1
Lychee 1
Mango 3
Milkfruit 1
Orange 2
Pomegranate 2
Sapota 2
hadoop@administrator:/Desktop/WordCountTutorial$
```

## Result:

Thus the installation of Hadoop one node cluster has been completed successfully.