

ANNA UNIVERSITY

REGIONAL CAMPUS, COIMBATORE



CS8711- CLOUD COMPUTING LABORATORY

LABORATORY RECORD
2021 - 2022

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

**ANNA UNIVERSITY-REGIONAL CAMPUS
COIMBATORE-641 046**

ANNA UNIVERSITY

REGIONAL CAMPUS, COIMBATORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



BONAFIDE CERTIFICATE

Certified that this is the bonafide record of Practical done in CS8711–
CLOUD COMPUTING LABORATORY by _____.

Register No. _____ in Fourth Year - Seventh Semester
during 2021-2022.

STAFF IN-CHARGE

HEAD OF THE DEPARTMENT

University Register No:

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

INTERNAL EXAMINER

EXTERNAL EXAMINER

INDEX

S.N o	Date	Topic	Page no	Mark	Signature
1	02/09/2021	Installing VirtualBox/VMware Workstation on Windows	1		
2	03/09/2021	Install C compiler in Virtual Machine and execute simple programs	15		
3	04/09/2021	Install Google App Engine.	39		
4	06/09/2021	Use GAE to launch web applications	43		
5	07/09/2021	Install and simulate CloudSim for a scheduling algorithm	51		
6	08/09/2021	To transfer files from one virtual machine to another	57		
7	09/09/2021	To launch virtual machine using OpenStack	61		
8	13/09/2021	Install Hadoop	71		

EX.No:1**Install Virtualbox/VMware Workstation****Date:02/09/2021****Aim:**

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

PROCEDURE TO INSTALL**Step 1- Download Link**

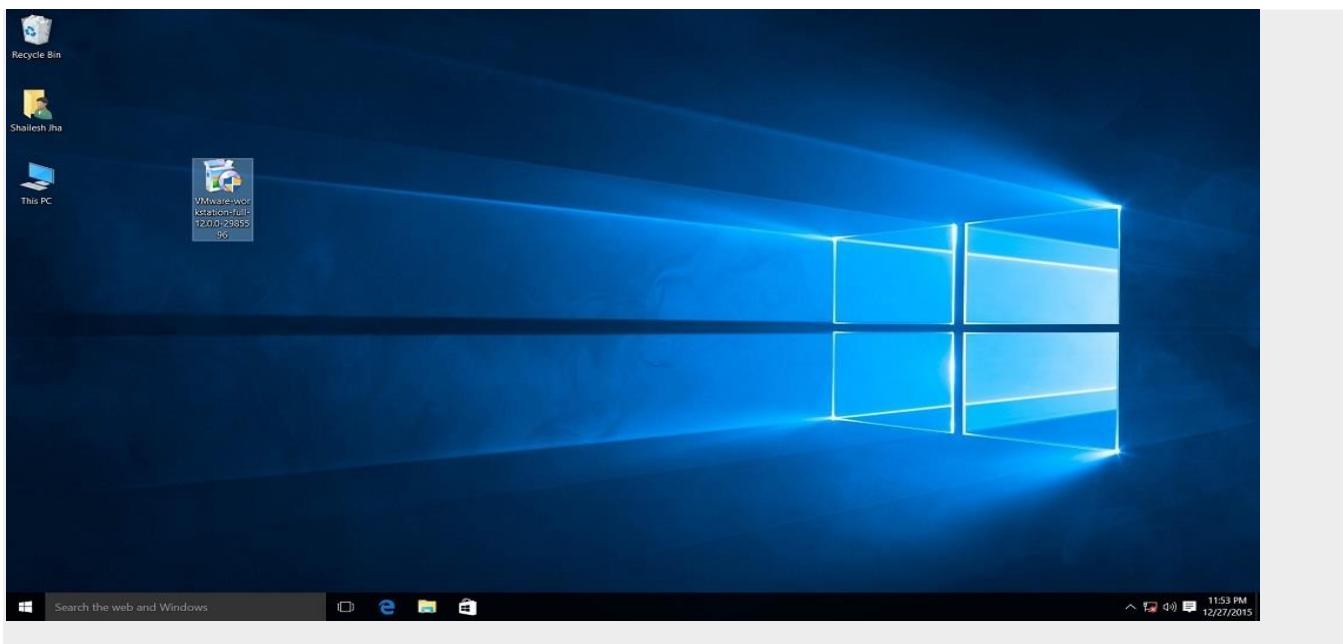
The link for downloading the software is <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>. Download the software for windows. The software will be around 541 MB.

Step 2- Download the installer file

It should probably be in the download folder by default, if the settings in our browser is not changed to any other custom folder. The File name will be like [VMware-workstation-full-15.5.1-15018445.exe](#). This file name can change depending on the version of the software currently available for download.

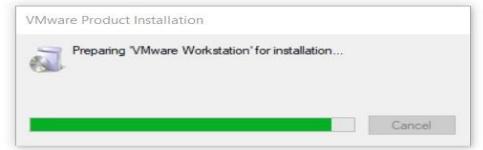
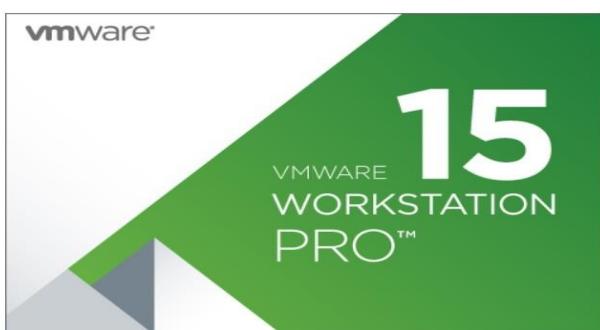
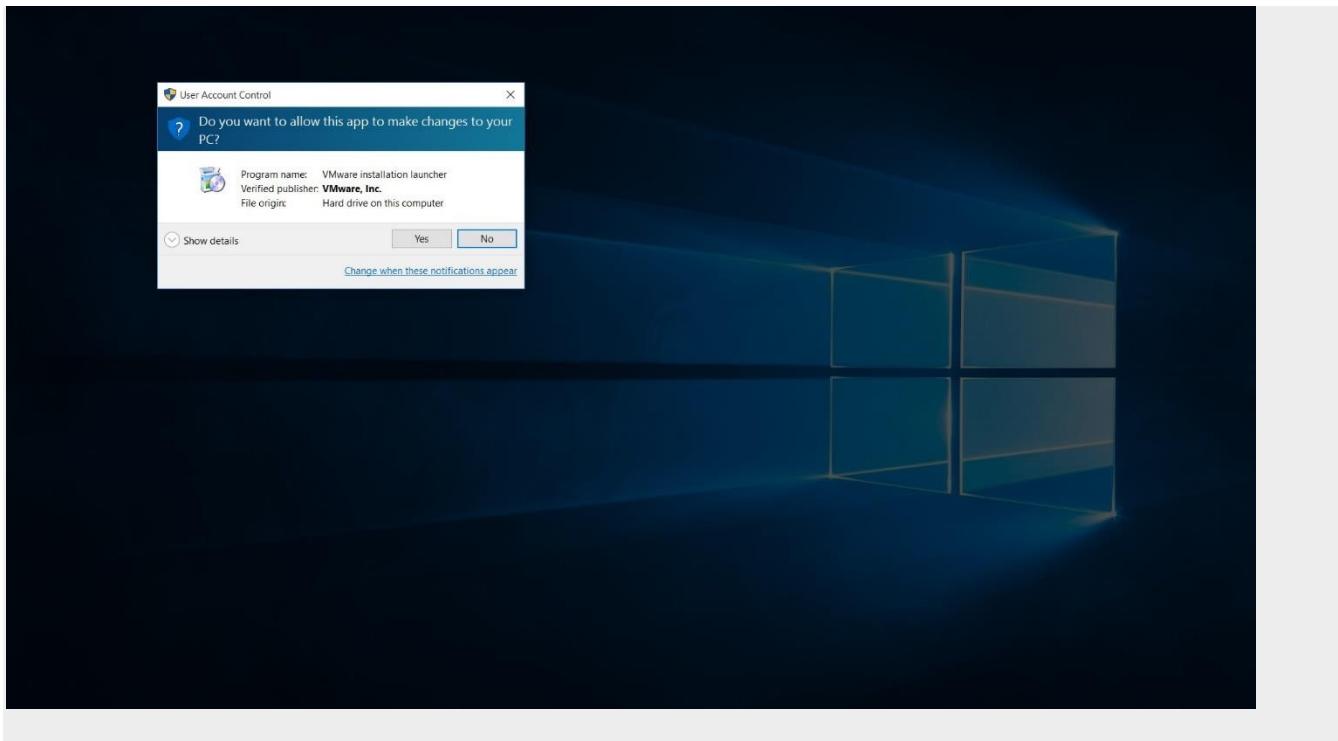
Step 3- Locate the downloaded installer file

For demonstration purpose, we have placed the downloaded installer on the desktop. Find the installer on the Desktop and double click to launch the application.



Step 4- User Access Control (UAC) Warning

Now the User Access Control (UAC) dialog box will appear. Click yes to continue. Initial Splash screen will appear. Wait for the process to complete.



Step 5- VMware Workstation Setup wizard

Now the VMware Workstation setup wizard dialog box will appear. Click next to continue.



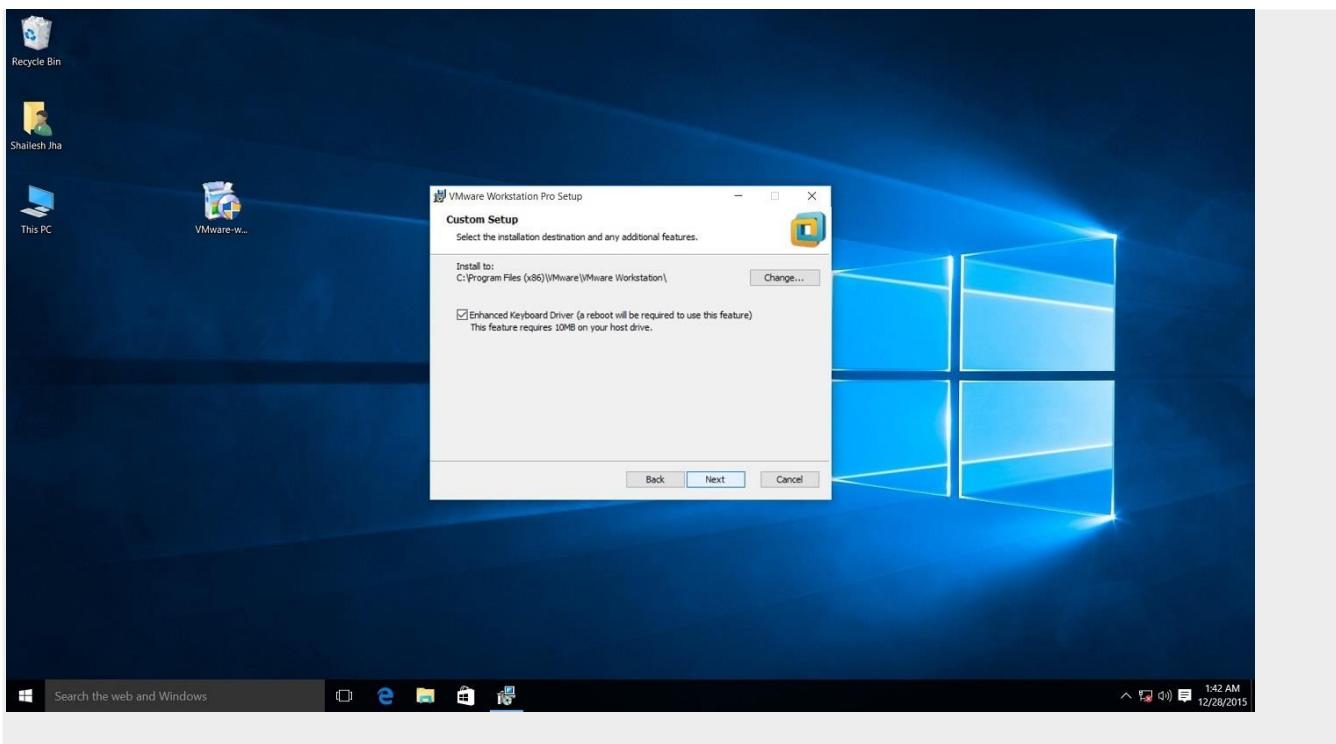
Step 6- End User License Agreement

The End User License Agreement dialog box is visible. Check “I accept the terms in the License Agreement” box and press next to continue.



Step 7- Custom Setup options

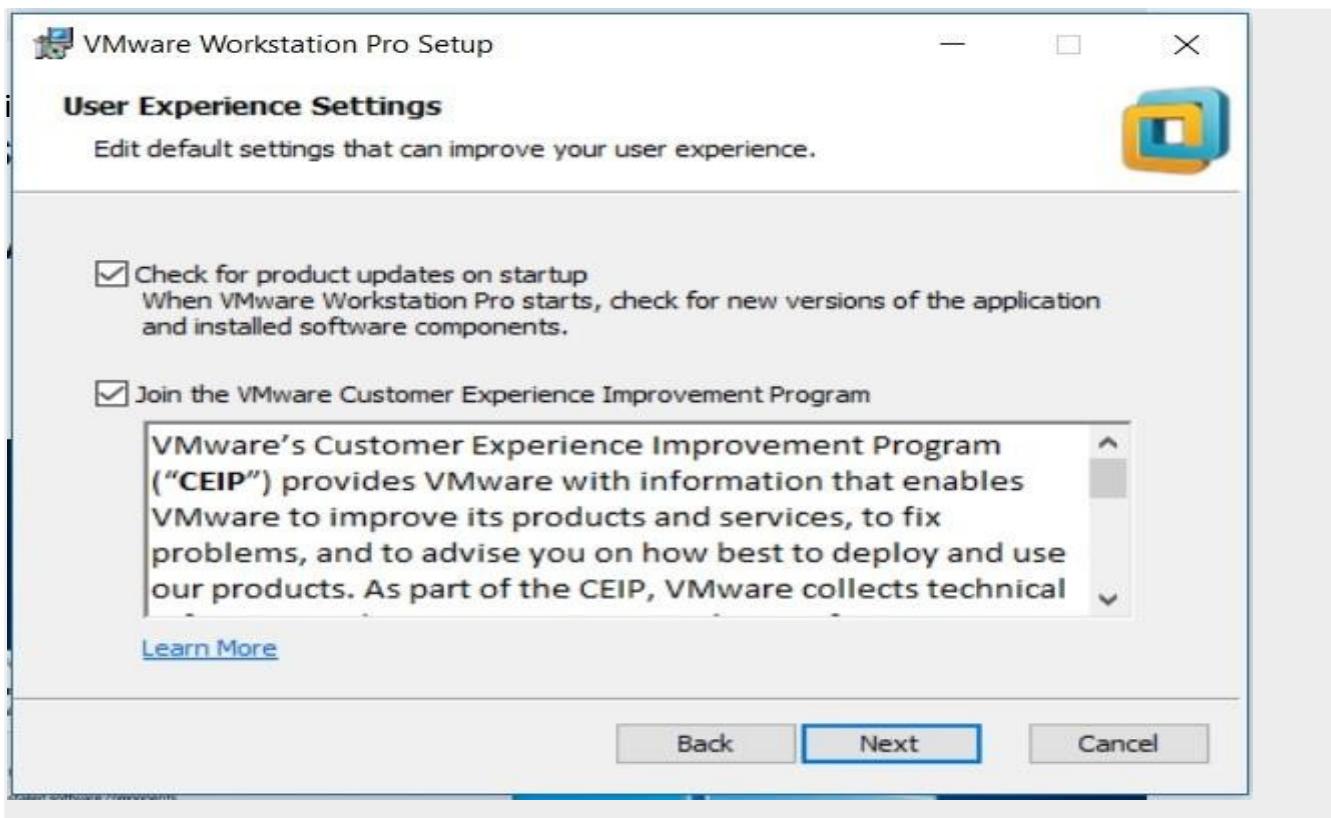
Select the folder in which the software is likely to be installed. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.



Step 8- User Experience Settings

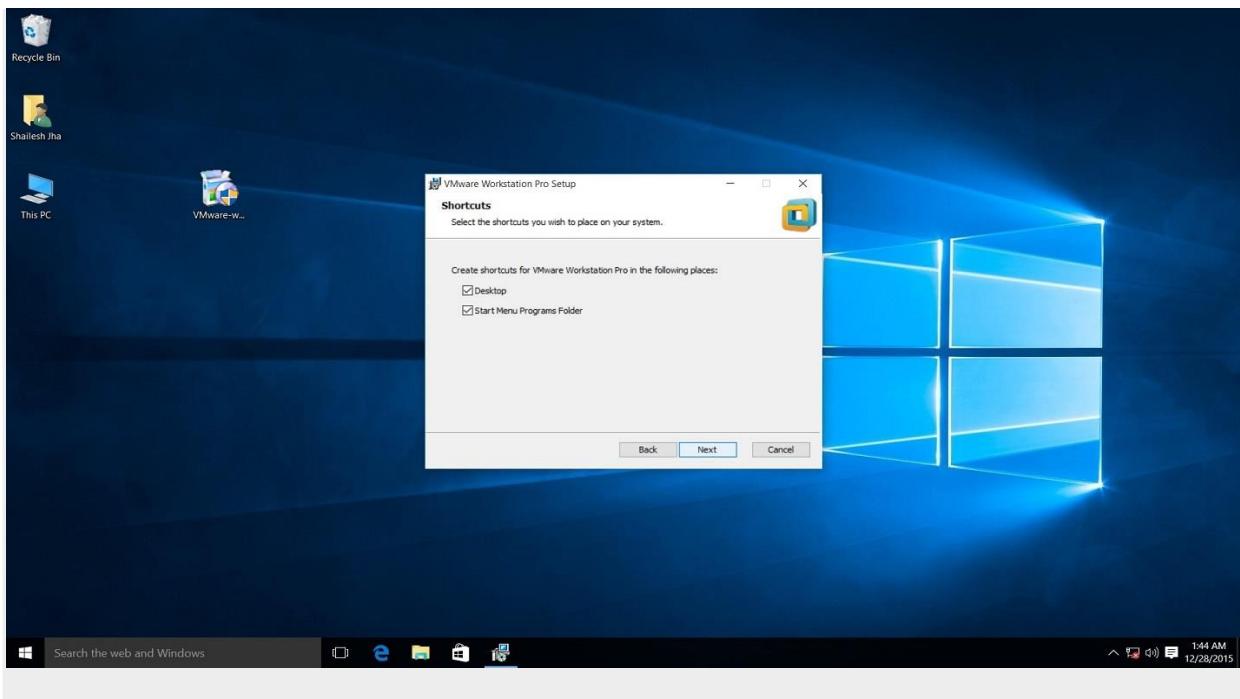
Next we are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”.

It's normally good to leave it to defaults that are unchecked.



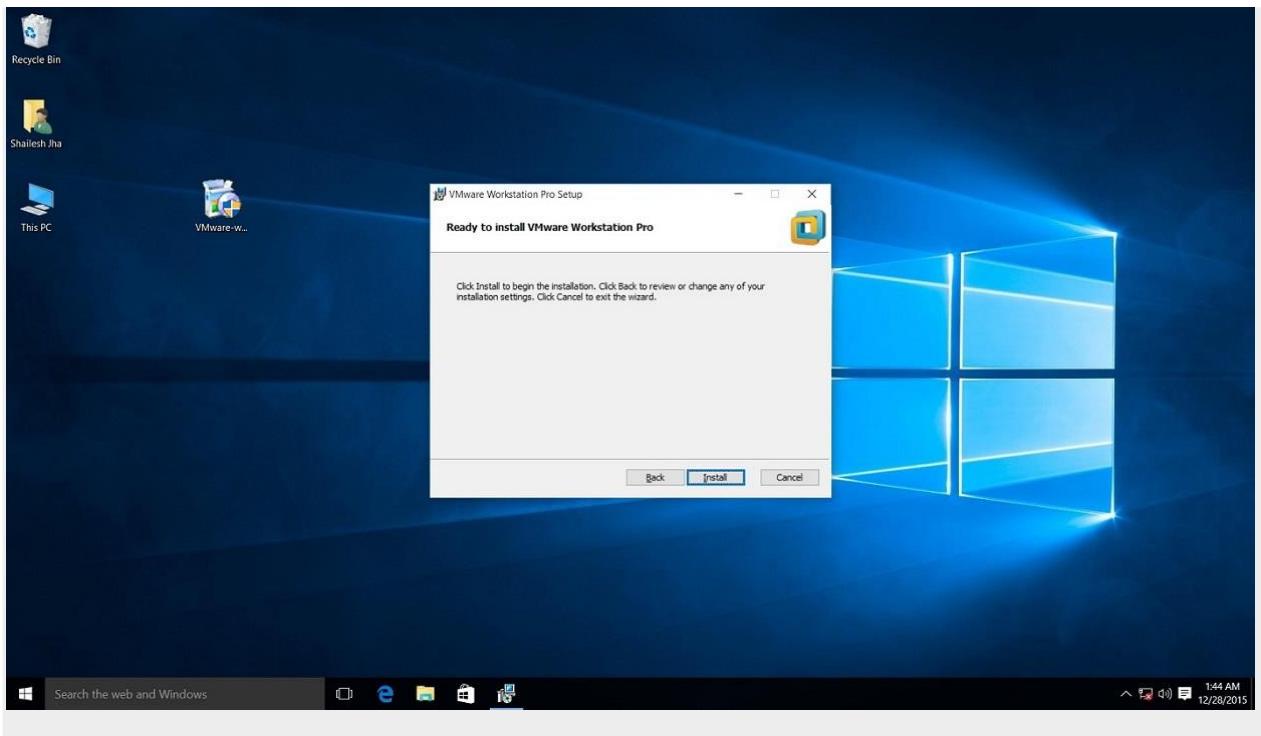
Step 9- Application Shortcuts preference

Next step is to select the place the shortcut icons to be placed on the system to launch the application. Select both the options, desktop and start menu and click next.

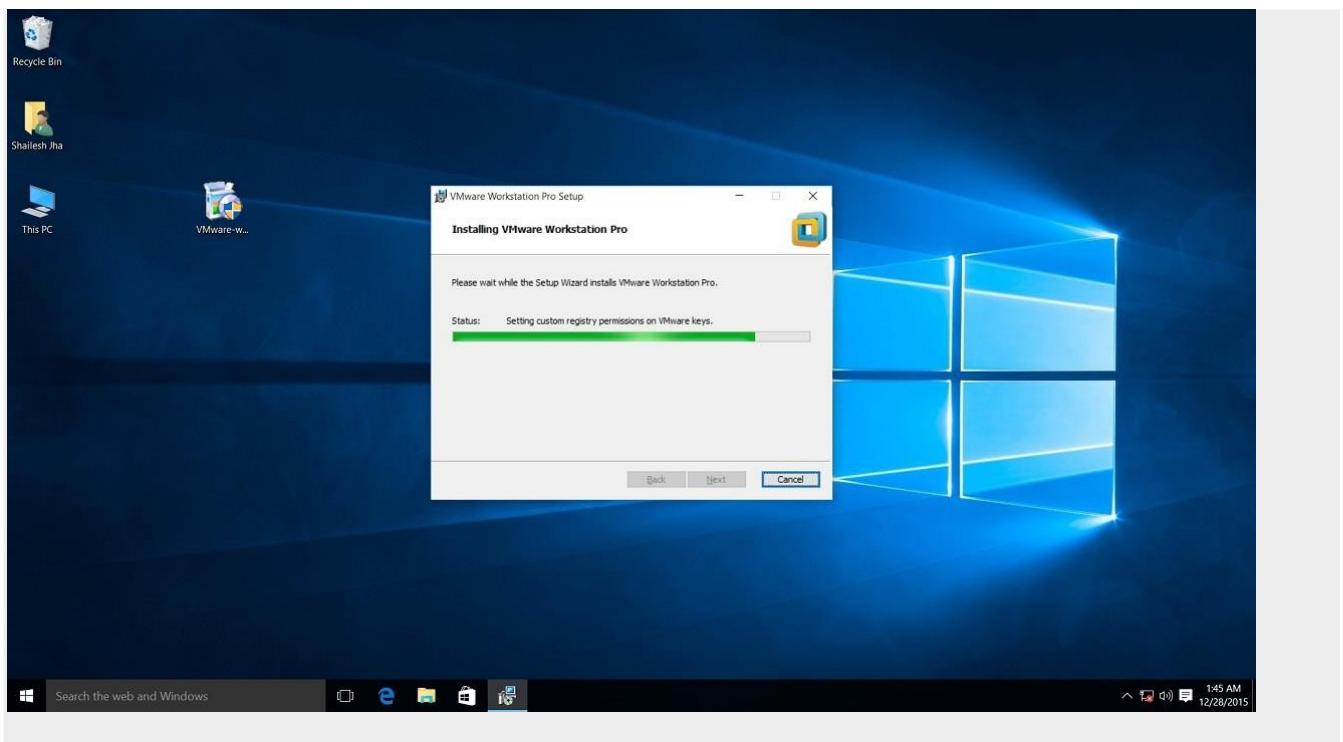


Step 10- Installation begins

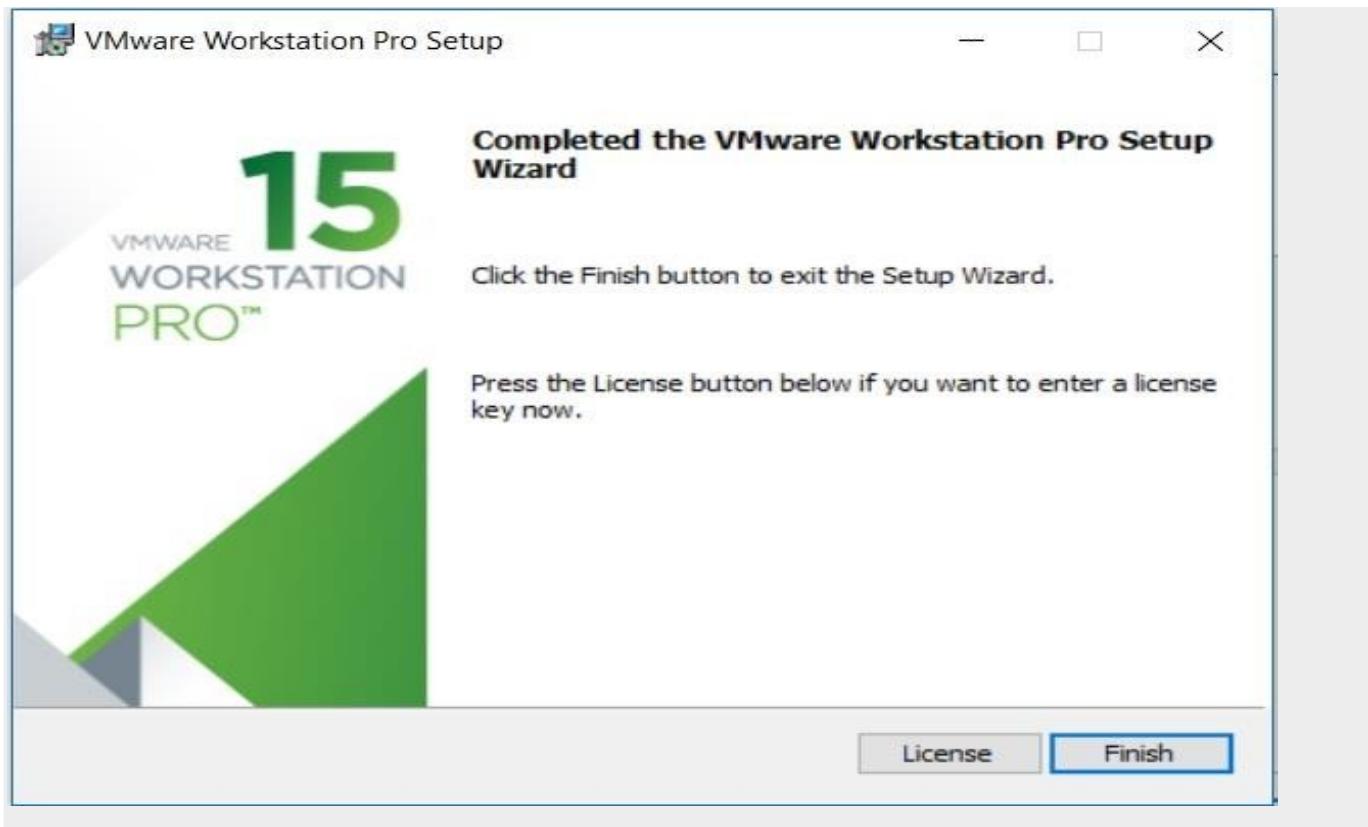
Now we can see the begin installation dialog box. Click install to start the installation process.



Below screenshot shows Installation in progress. Wait for this to complete.

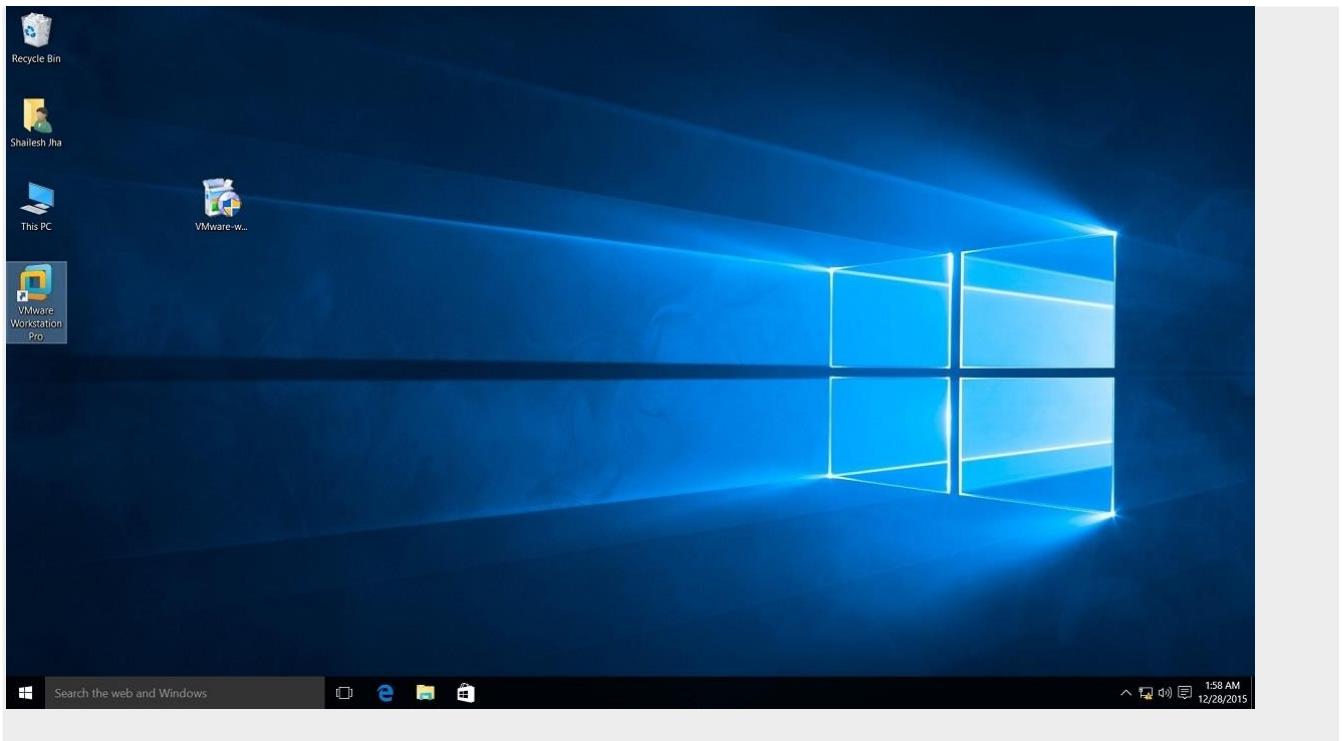


At the end, the installation will be complete. Click finish and we are done with the installation process. We need to restart your computer. Click on Yes to restart.



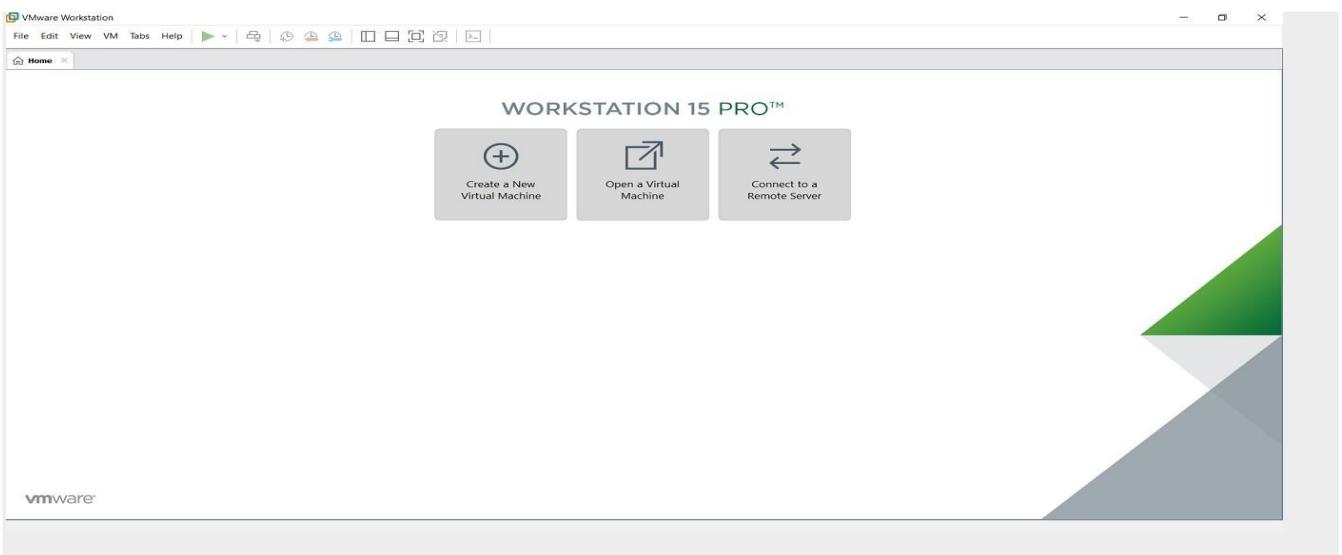
Step 11- Launch VMware Workstation

After the installation completes, VMware Workstation icon on the desktop is visible. Double click on it to launch the application.



Step 12- Licence Key

The dialog box asking for license key appears, click on trial or enter the license key. Then the VMware Workstation 15 Pro running on your windows 10 desktop. If don't have the license key, we can use 30 days trial.



Result:

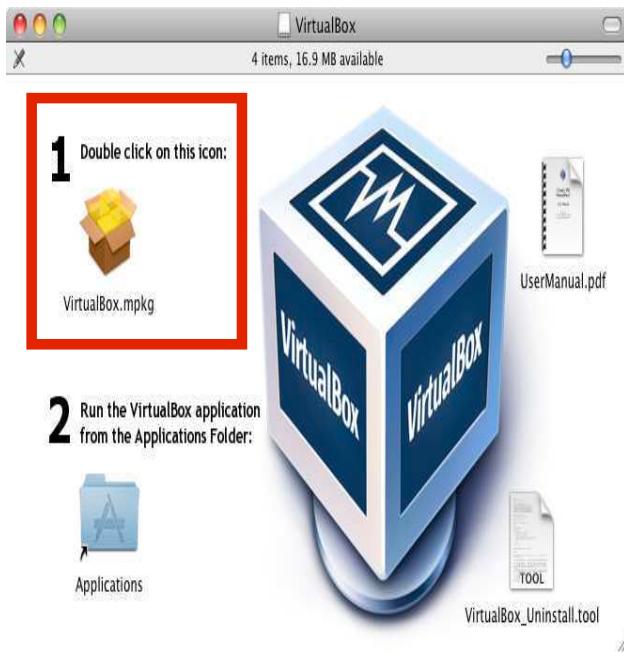
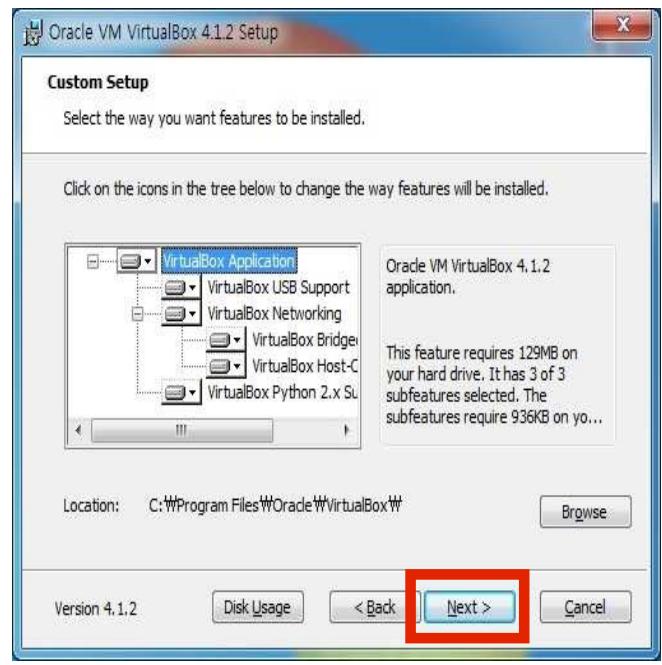
Thus to install Virtualbox/VMware Workstation with different flavours of Linux or Windows OS on top of Windows 10 is successfully completed and the working is tested.

EX.No:2**Install a C compiler in the virtual machine****Date: 03/09/2021****Aim:**

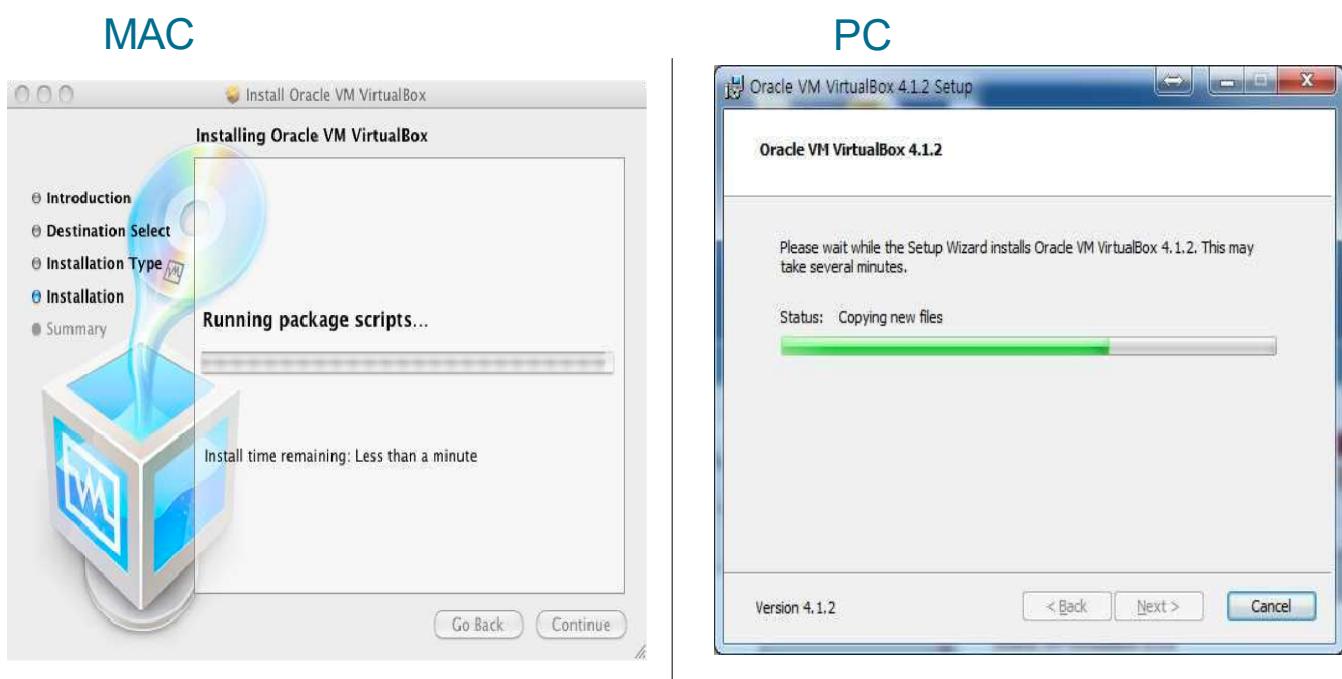
To Install a C compiler on the virtual machine created using virtual box and to execute Simple C programs using the installed C compiler.

PROCEDURE**Install VirtualBox**

1. Visit <http://www.virtualbox.org/wiki/downloads>.
2. Download Virtual Box platform packages for the corresponding OS.
3. Open the Installation Package by double clicking it.

MAC**PC**

4. Click continues and it finishes installing Virtual Box.



5. Close the window, after the installation is completed.

Download Linux

1. Visit the page <http://www.ubuntu.com/download/ubuntu/download>
2. Choose the Latest version of Ubuntu 64-bit and click “Start Download”

1 Download Ubuntu

Click the big orange button to download the latest version of Ubuntu. You will need to create a CD or USB stick to install Ubuntu.

Our long-term support (LTS) releases are supported for three years on the desktop. Perfect for organisations that need more stability for larger deployments.

Download options

Ubuntu 11.04 – Latest version

32-bit (recommended)

Download started

Ubuntu 11.04
32-bit

Direct url for this download

Additional options

Take a look at a full list of our previous versions and [alternative downloads](#).

If you're running Windows

You can use Ubuntu Windows installer to run Ubuntu alongside your current system.

[Ubuntu Windows Installer](#)

Other ways to get Ubuntu

[Order CDs](#)

[Ubuntu Server](#)

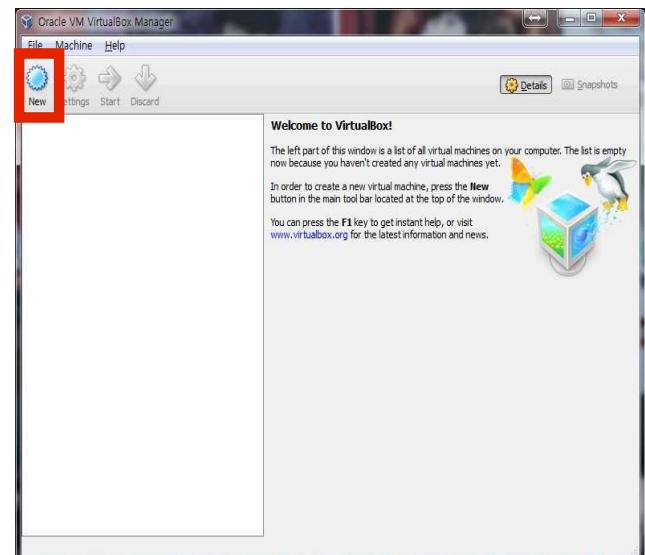
Install Linux using Virtual Box

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

MAC

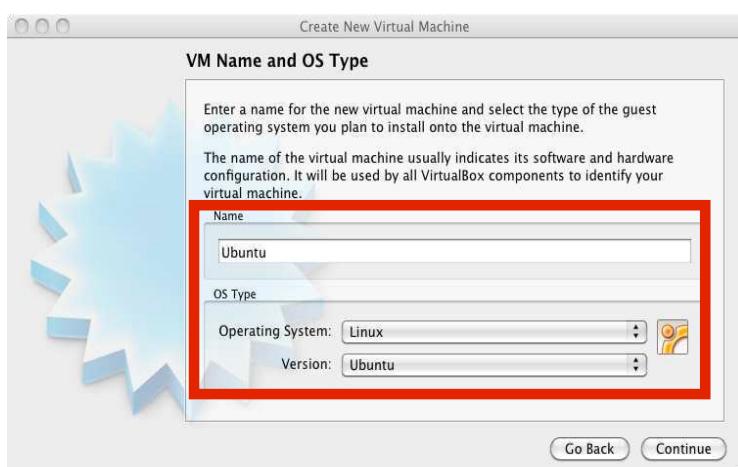


PC



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

MAC

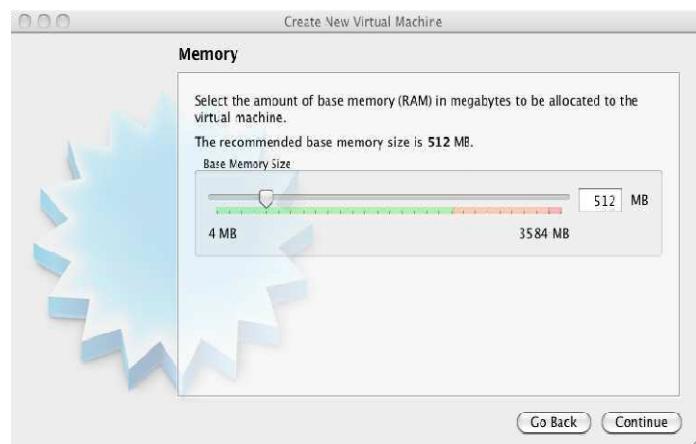


PC

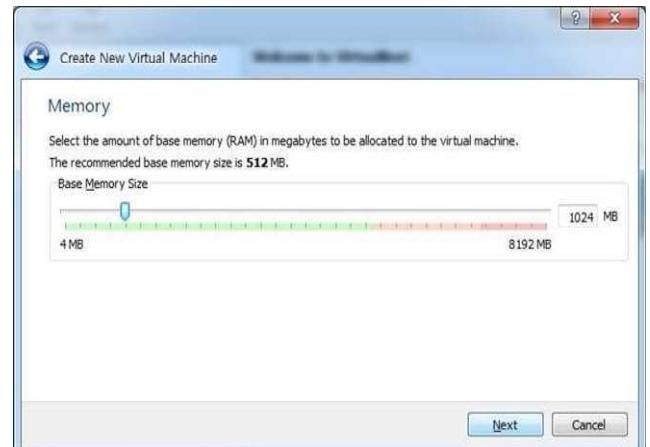


5. Choose the amount of memory to allocate (Minimum 1024MB, Recommended – 4096MB).
6. Click Continue or Next

MAC



PC



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

MAC



PC



9. Choose VDI (VirtualBox Disk Image).
10. Click Continue or Next.

MAC

PC



- II. Choose “Dynamically Allocated” click continue. This helps us to change the size of the Hard Disk of our Virtual Machine.

MAC

PC

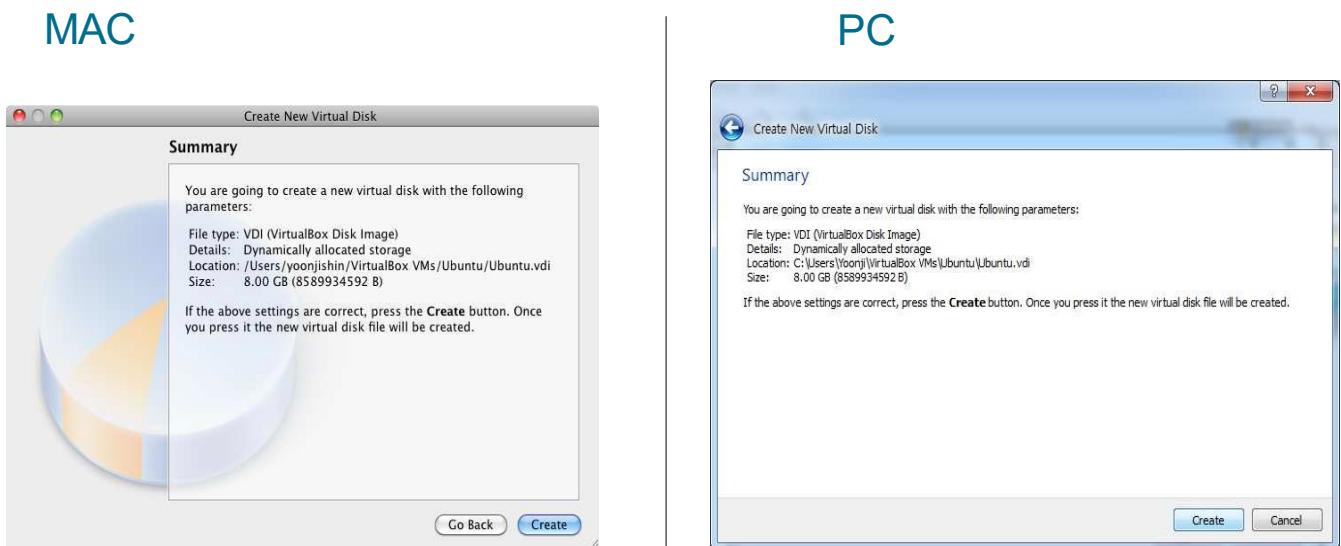


I2. Click the folder icon and choose the ubuntu.iso file, downloaded.

I3. Select the size of the Virtual Disk(Recommended 10 GB) and click continue

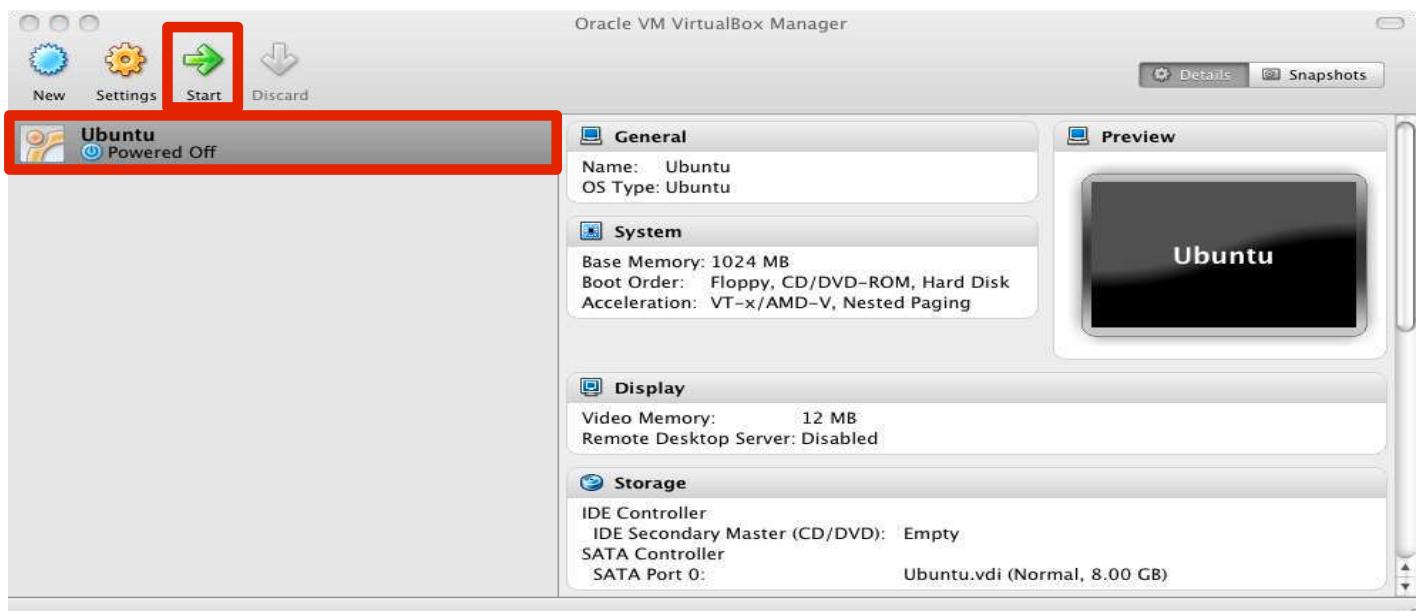


I4. Click Create.



Running Linux

I. Choose Ubuntu from left column and click Start



2. Click continue on pop-up window

MAC



PC

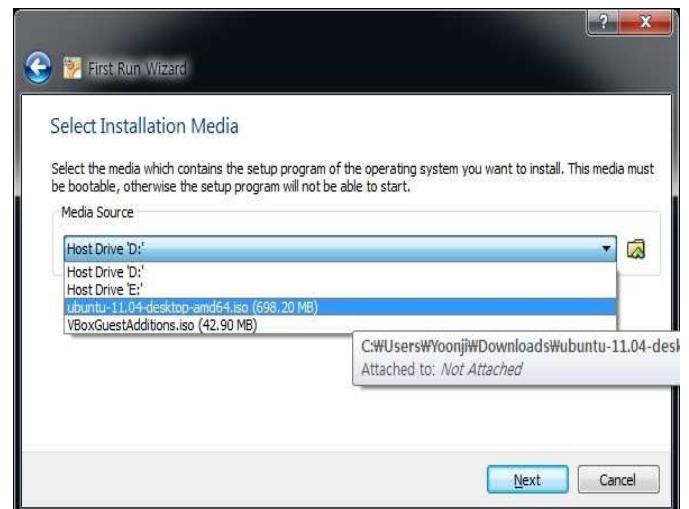


3. Click the folder icon and choose the ubuntu.iso file you downloaded and click continue and start

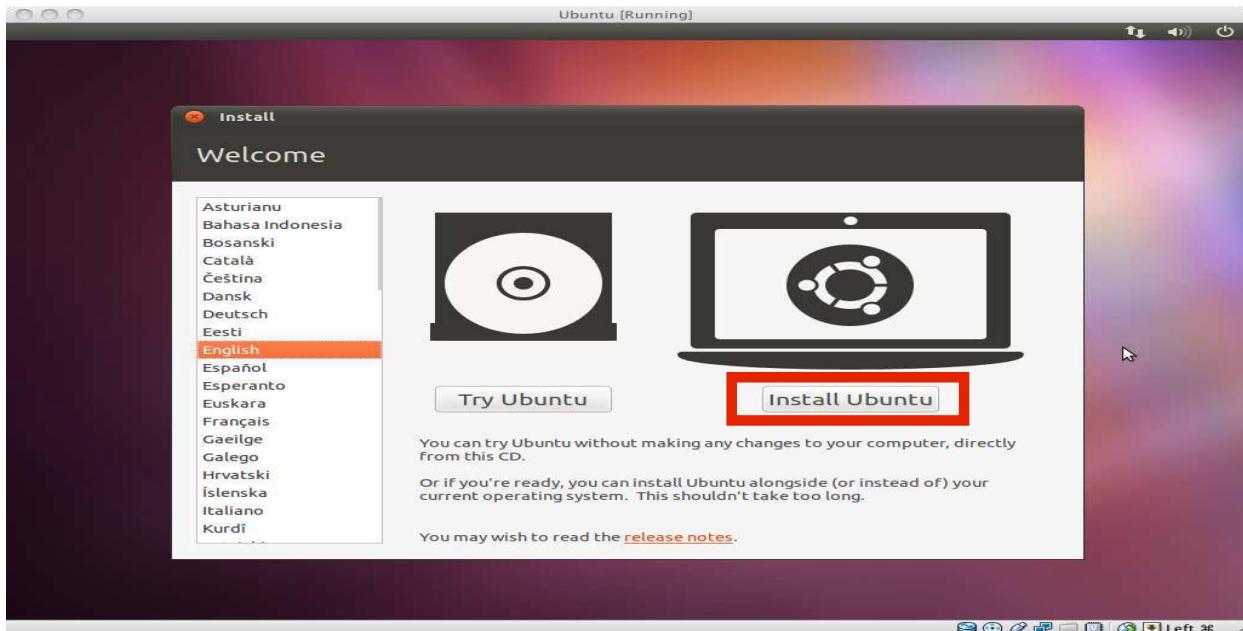
MAC



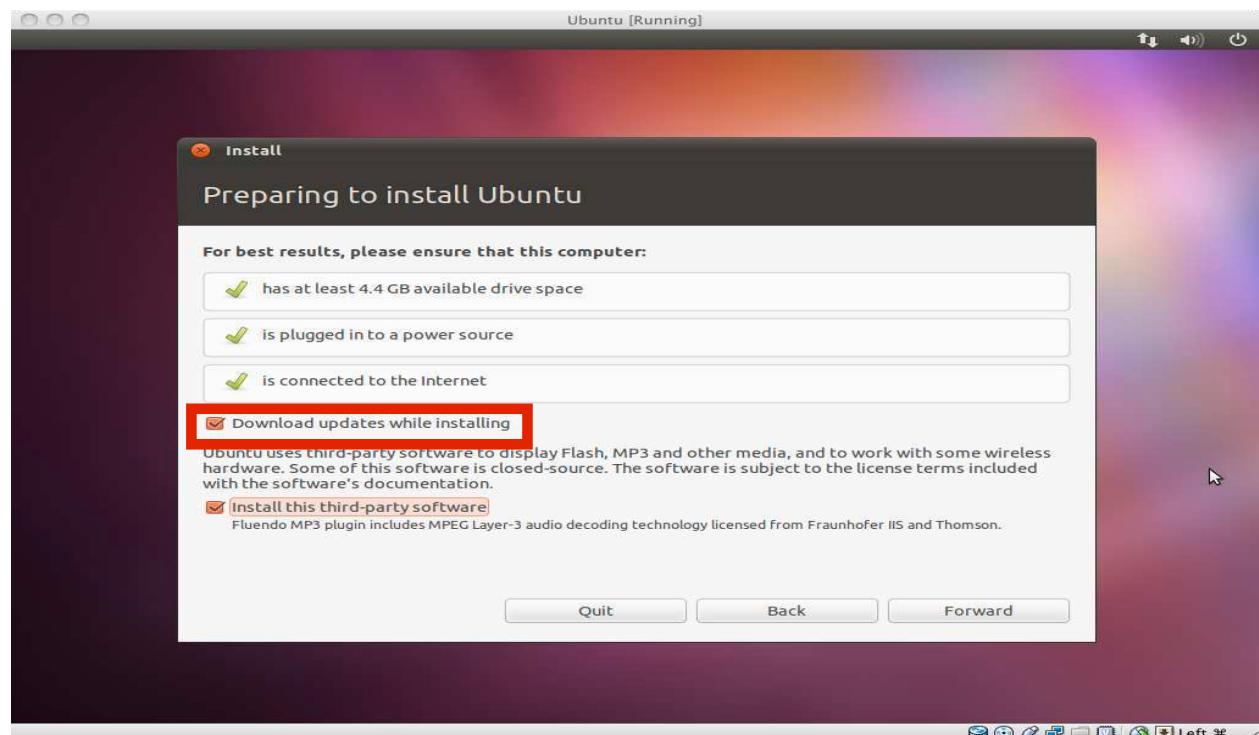
PC



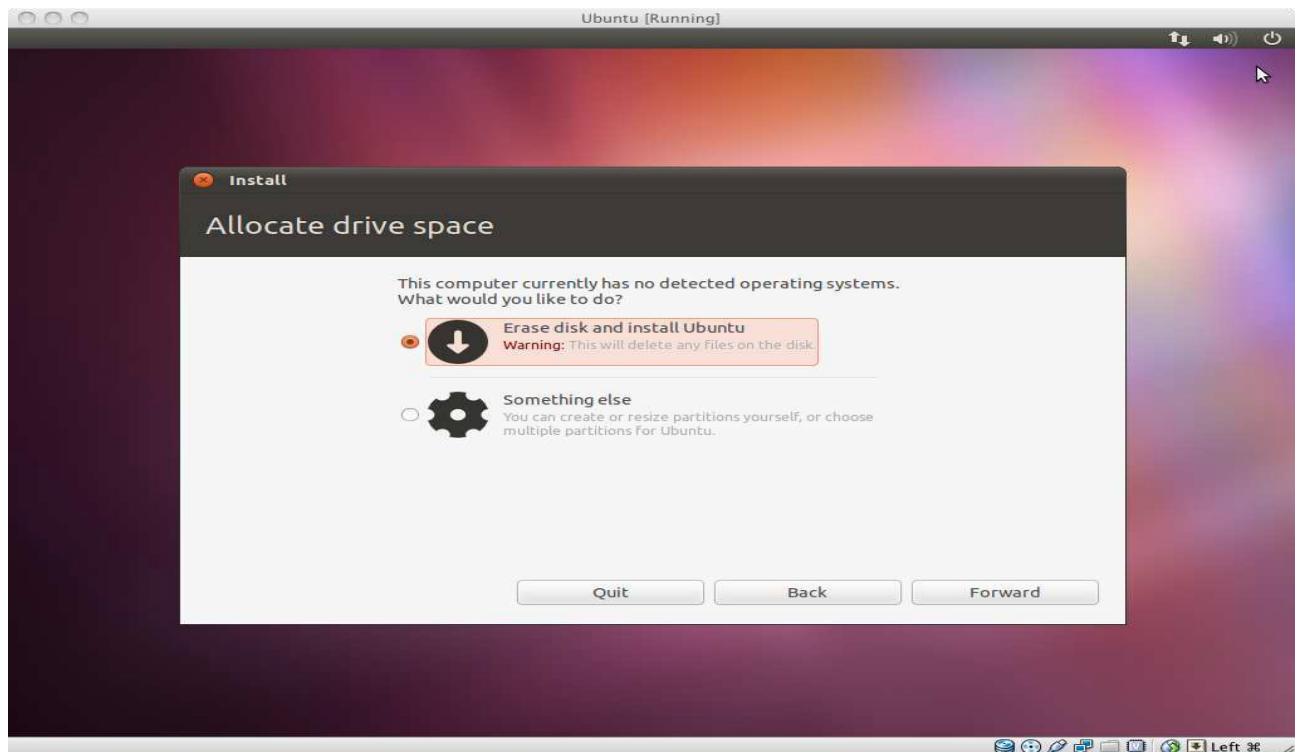
4. Click Install Ubuntu



5. Check “Download updates” and click Forward.

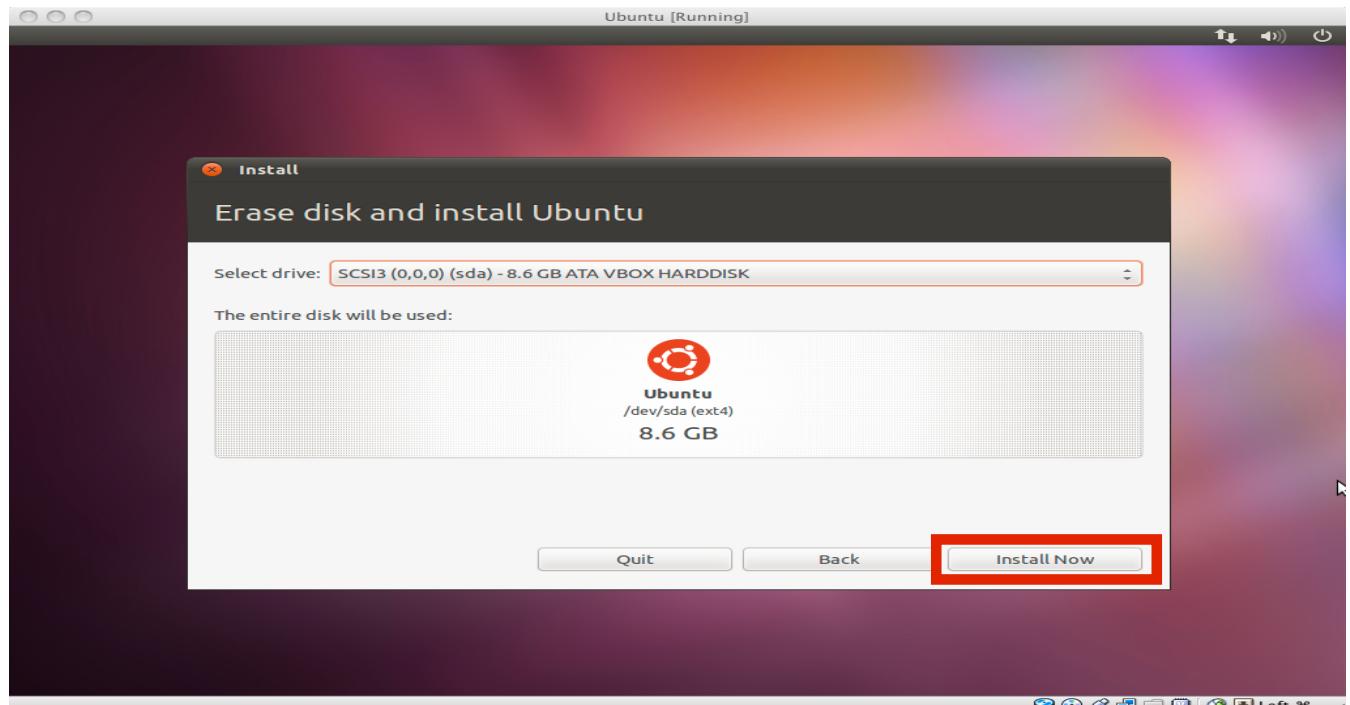


6. Choose “Erase disk and install Ubuntu” and click Forward.



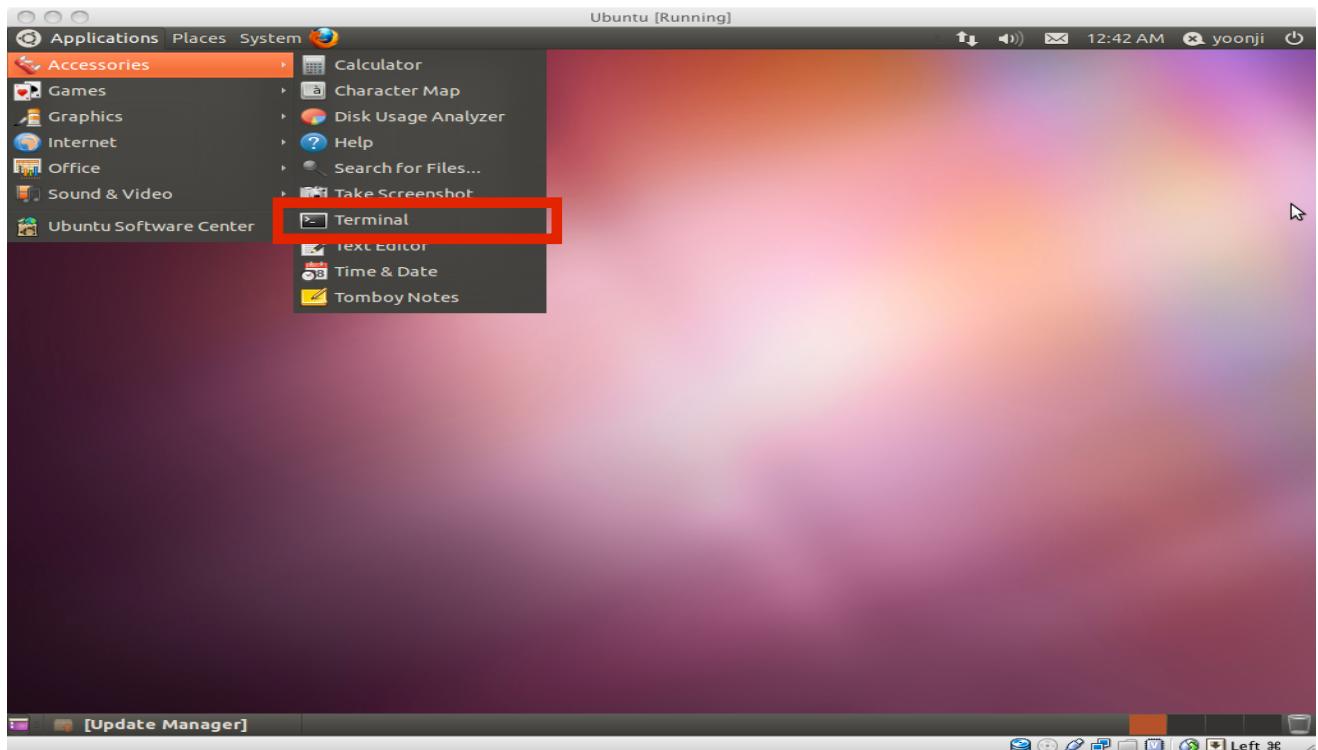
7. Click “Install Now” and wait. The installation takes place for a certain amount of time depending on the system.

8. When finished, click Restart and press Enter.

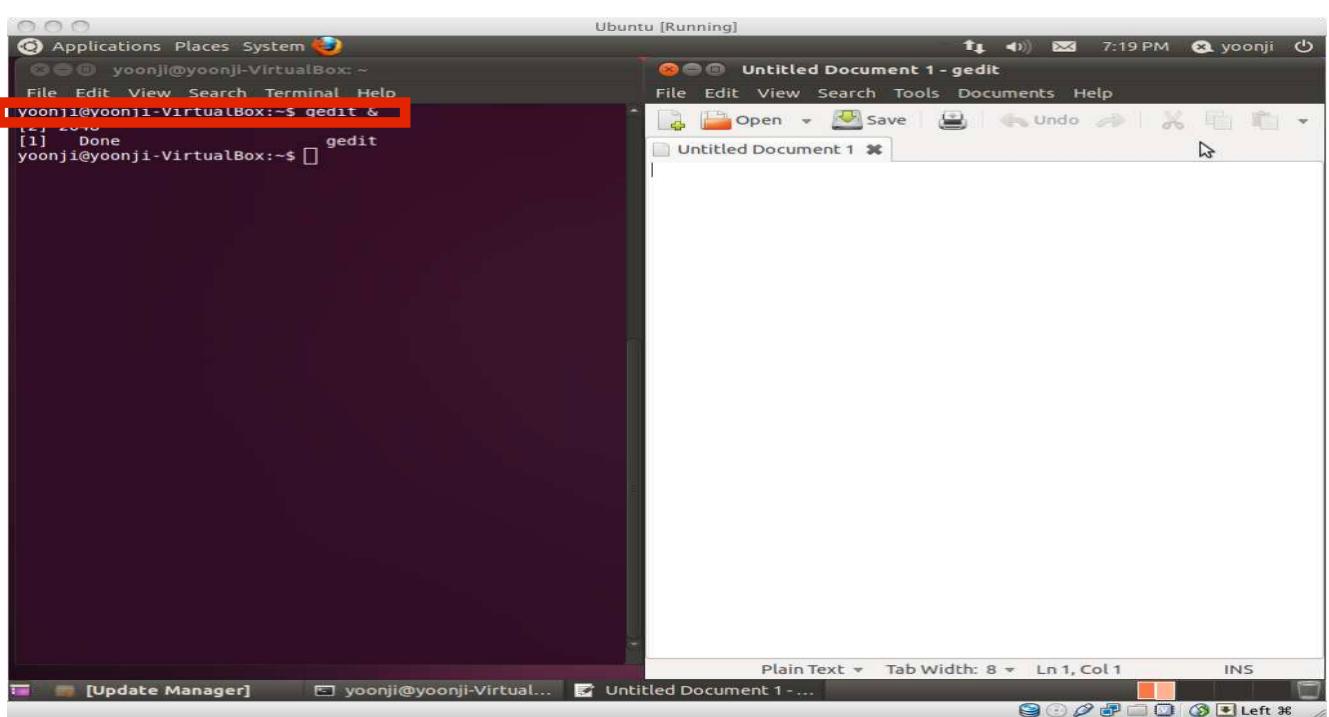


C Programming on Linux

I. Open Terminal (Applications->Accessories->Terminal)

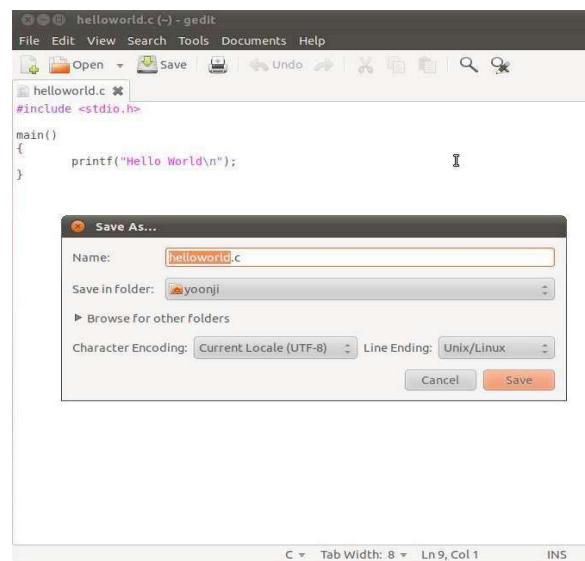


2. Open gedit by typing “gedit &” on terminal or we can use any text editor document available on the System.

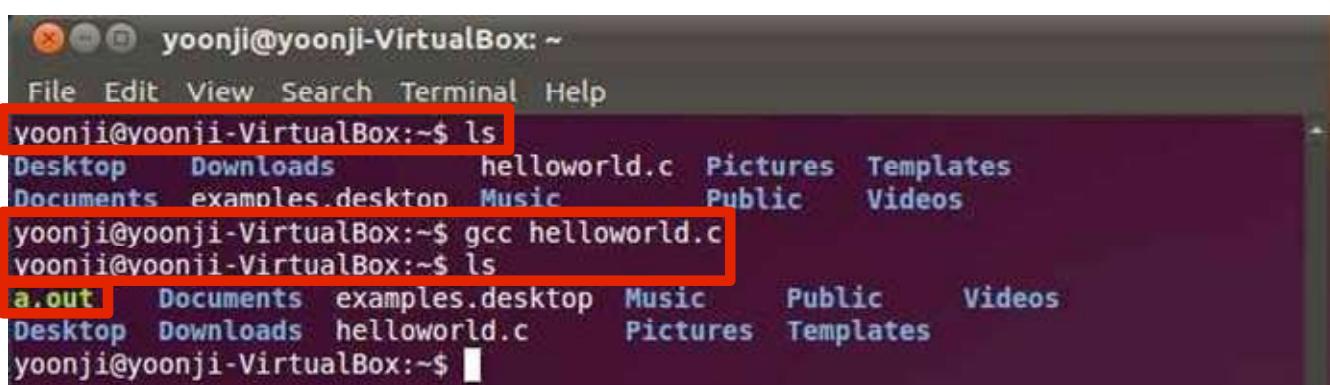


3. Type the following on gedit (or any other text editor)

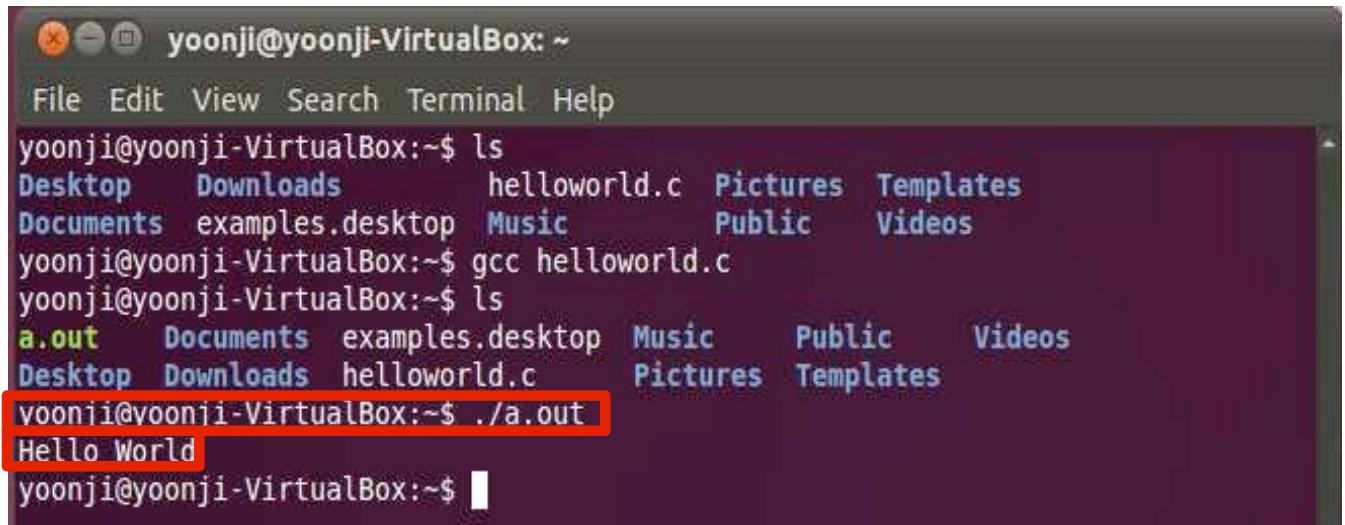
```
#include<stdio.h> main()
{
    printf("Hello    World\n");
}
```



4. Save this file as “helloworld.c”
5. Typing “ls” on Terminal helps to see all files under current folder
6. Confirm that “helloworld.c” is in the current directory. If not, type cd DIRECTORY_PATH to go to the respective directory that has “helloworld.c”
7. Install gcc in the system . Using the code,
- ```
sudo apt install gcc
```
- and press enter. The installation takes place.
8. Type “gcc helloworld.c” to compile, and type “ls” to confirm that a new executable file “a.out” is created
9. Type “./a.out” on Terminal to run the program



10. If you see “Hello World” on the next line, you just successfully ran your first C program!



The screenshot shows a terminal window with the following session:

```
yoonji@yoonji-VirtualBox:~$ ls
Desktop Downloads helloworld.c Pictures Templates
Documents examples.desktop Music Public Videos
yoonji@yoonji-VirtualBox:~$ gcc helloworld.c
yoonji@yoonji-VirtualBox:~$ ls
a.out Documents examples.desktop Music Public Videos
Desktop Downloads helloworld.c Pictures Templates
yoonji@yoonji-VirtualBox:~$./a.out
Hello World
yoonji@yoonji-VirtualBox:~$
```

The command `./a.out` is highlighted with a red rectangle.

- I1. Try other C program codes from online.

### Result:

Thus to install a C compiler in the virtual machine created using virtual box and execute Simple C program is successfully completed and the output is verified.

**EX.No:3****Install Google App Engine****Date: 04/09/2021****Aim:**

To Install and run Google App Engine on Windows 10 and create a “hello world” app and other simple web application using python/java.

**Procedure:**

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

**Pre-Requisites: Python 2.5.4**

If Python2.5.4 is not installed in the computer, then download and Install Python 3.0 or any other latest releases from:

<http://www.python.org/download/releases/3.0/>

**Download and Install**

The download package of Google App Engine SDK can be found by going to:

<http://code.google.com/appengine/downloads.html>

and download the appropriate install package.

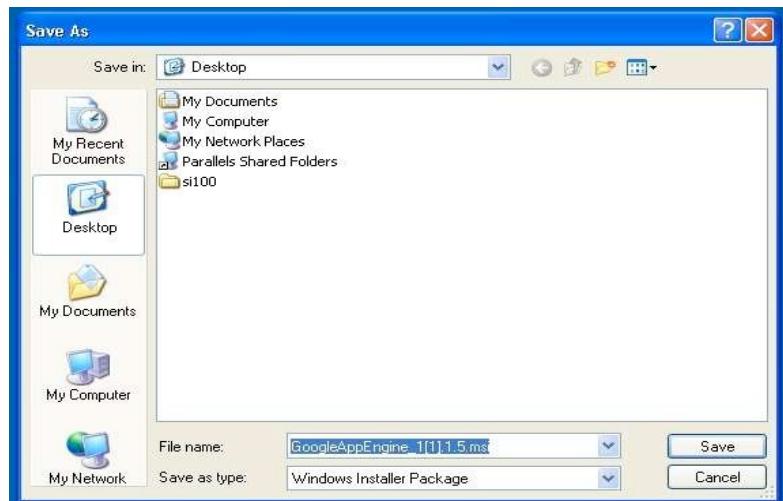
**Download the Google App Engine SDK**

Before downloading, please read the [Terms](#) that govern your use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the [SDK Release Notes](#) for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our [Issue Tracker](#).

| Platform              | Version          | Package                                           | Size   | SHA1 Checksum                            |
|-----------------------|------------------|---------------------------------------------------|--------|------------------------------------------|
| Windows               | 1.1.5 - 10/03/08 | <a href="#">GoogleAppEngine_1.1.5.msi</a>         | 2.5 MB | e974312b4aefc0b3873ff0d93eb4c525d5e88c30 |
| Mac OS X              | 1.1.5 - 10/03/08 | <a href="#">GoogleAppEngineLauncher-1.1.5.dmg</a> | 3.6 MB | f62208ac01c1b3e39796e58100d5f1b2f052d3e7 |
| Linux/Other Platforms | 1.1.5 - 10/03/08 | <a href="#">google_appengine_1.1.5.zip</a>        | 2.6 MB | cbb9ce817bdabf1c4f181d9544864e55ee253de1 |

Download the Windows installer—the simplest thing is to download it to the Desktop



Double Click on the **GoogleApplicationEngine** installer and proceed to the installation.



Click through the installation wizard, and it should install the App Engine. If Python 2.5 is not installed, it will install Python 2.5 as well.



Once the installation is complete can, the downloaded installer can be discarded.

## **Result:**

Thus the Google App Engine is successfully installed and the creation of “hello world app” is executed and the output is verified.

**EX.No:4****GAE launcher to Launch a Web Application****Date: 06/09/2021****Aim:**

To use Google App Engine launcher to launch a web application and study its working.

**Procedure:**

To create a simple application in GAE launcher, we use the "+" option to have the launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.

Make a folder for on the Google App Engine applications. Let's make the Folder on the Desktop called "apps" – the path to this folder is:

**C:\Documents and Settings\user\Desktop\apps**

And then make a sub—folder in within **apps** called "**ae-01-trivial**" - the path to this folder would be:

**C:\ Documents and Settings \user\Desktop\apps\ae-01-trivial**

Using a text editor such as JEdit ([www.jedit.org](http://www.jedit.org)), create a file called **app.yaml** in the **ae-01-trivial** : folder with the following contents:

**application: ae-01-trivial**

**version: 1**

**runtime: python**

**api\_version: 1**

**handlers:**

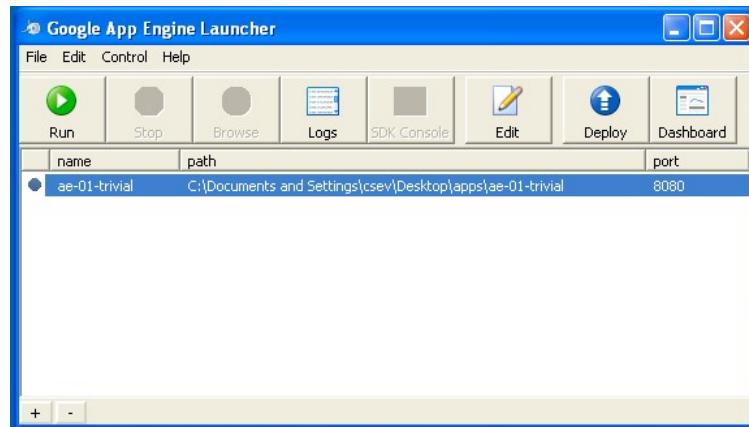
**- url: /.\***

**script: index.py**

Then create a file in the **ae-01-trivial** folder called **index.py** with three lines in it:

```
print 'Content-Type:text/plain'
print ''
print 'Hello there Chuck'
```

Then start the **Google App Engine Launcher** program that can be found under **Applications**. Use the **File -> Add Existing Application** command and navigate into the **apps** directory and select the **ae-01-trivial** folder. Once the application is added, then select it so that we can control the application using the launcher.



Once we have selected the application and press **Run**. After a few moments , the application will start and the launcher will show a little green icon next in the application. Then press **Browse** to open a browser pointing at the application which is running at **<http://localhost:8080>**

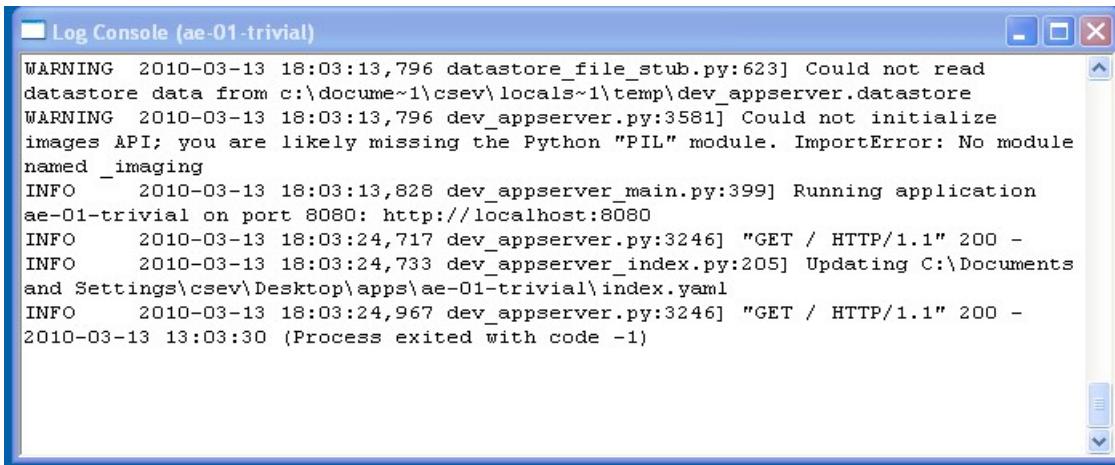
Paste **<http://localhost:8080>** into the browser and you should see your application asfollows:



We can edit the **index.py** to change the name “Chuck” to any nameand then press Refresh in the browser to verify the updates.

### Watching the Log

We can watch the internal log of the actions that the web server is performing when we are interacting with the application in the browser. Select the application in the Launcher and press the Logs button to bring up a log window:



```

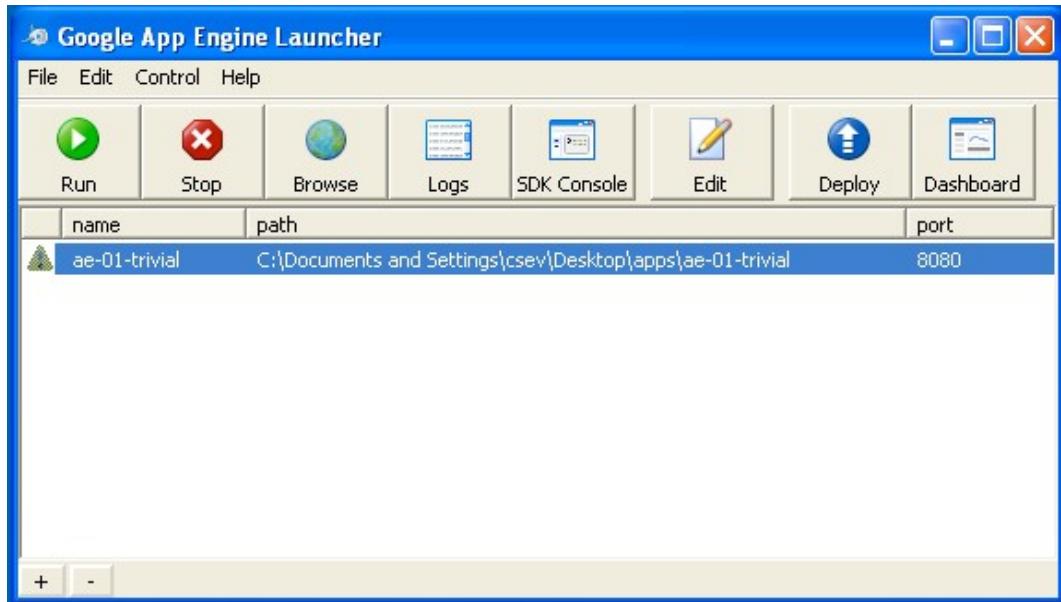
Log Console (ae-01-trivial)
WARNING 2010-03-13 18:03:13,796 datastore_file_stub.py:623] Could not read
datastore data from c:\docume~1\csev\locals~1\temp\dev_appserver.datastore
WARNING 2010-03-13 18:03:13,796 dev_appserver.py:3581] Could not initialize
images API; you are likely missing the Python "PIL" module. ImportError: No module
named _imaging
INFO 2010-03-13 18:03:13,828 dev_appserver_main.py:399] Running application
ae-01-trivial on port 8080: http://localhost:8080
INFO 2010-03-13 18:03:24,717 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO 2010-03-13 18:03:24,733 dev_appserver_index.py:205] Updating C:\Documents
and Settings\csev\Desktop\apps\ae-01-trivial\index.yaml
INFO 2010-03-13 18:03:24,967 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
2010-03-13 13:03:30 (Process exited with code -1)

```

Each time you press **Refresh** button in the browser – we can see it retrieving the output with a **GET** request.

### Dealing With Errors

With two files to edit, there are two general categories of errors that we may encounter. If there is amistake on the **app.yaml**file, the App Engine will notstart and the launcher will show a yellow icon near your application:



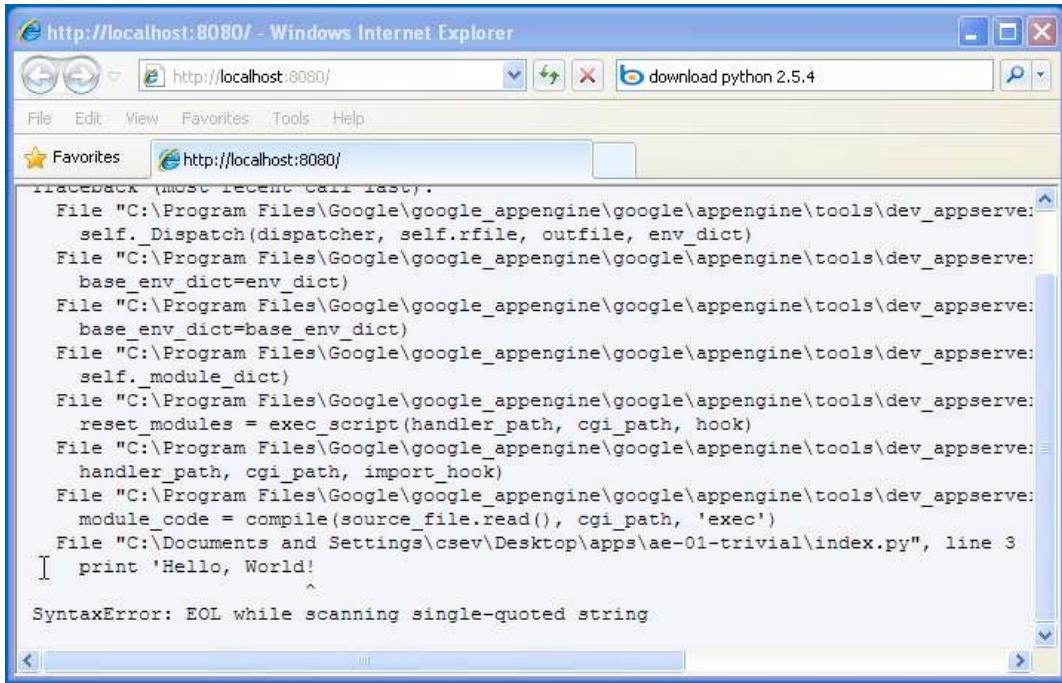
To get more detail on what is going wrong, we can take a look at the log for the application:



```
Log Console (ae-01-trivial)
Unknown url handler type.
<URLMap
 static_dir=None
 secure=default
 script=None
 url='.*'
 static_files=None
 upload=None
 mime_type=None
 login=optional
 require_matching_file=None
 auth_fail_action=redirect
 expiration=None
>
in "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\app.yaml", line 8,
column 1
```

In the instance – the mistake is mis-indenting the last line in the **app.yaml** (line 8).

If you make a syntax error in the **index.py** file, a Python traceback error will appear in your browser.



```
http://localhost:8080/ - Windows Internet Explorer
File Edit View Favorites Tools Help
Favorites http://localhost:8080/
Traceback (most recent call last):
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 self._Dispatch(dispatcher, self.rfile, outfile, env_dict)
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 base_env_dict=env_dict)
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 base_env_dict=base_env_dict)
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 self._module_dict)
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 reset_modules = exec_script(handler_path, cgi_path, hook)
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 handler_path, cgi_path, import_hook)
 File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserver:
 module_code = compile(source_file.read(), cgi_path, 'exec')
 File "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\index.py", line 3
[print 'Hello, World!
 ^
SyntaxError: EOL while scanning single-quoted string
```

The error we need to see is likely to be the last few lines of the output. When we make a mistake in the **app.yaml** file – we must fix the mistake and attempt to start the application again. If we make a mistake in a file like **index.py**, we can simply fix the file and press refresh in the browser – there is no need to restart the server.

## Shutting Down the Server

To shut down the server, using the Launcher, select the application and press the **Stop** button.

## Result:

Thus to use Google App Engine launcher to launch a web application is successfully completed and the output from the web browser is verified.

**EX.No:5****Install CloudSim and run Scheduling Algorithm****Date: 07/09/2021****Aim:**

To simulate a cloud scenario using CloudSim by installing it and run a scheduling algorithm that is not present in CloudSim.

**Procedure:****How to use CloudSim in Eclipse**

CloudSim is written in Java. The knowledge needed 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. It is a library and, hence, CloudSim does not have to be installed. Normally, we can unpack the downloaded package in any directory, add it to the Java classpath(build path) and it is ready to be used. Please verify whether Java is available on your system.

To use CloudSim in Eclipse:

1. Download CloudSim installable files from  
<https://code.google.com/p/cloudsim/downloads/list> and unzip
2. Open Eclipse
3. Create a new Java Project: File -> New
4. Import an unpacked CloudSim project into the new Java Project. Copy all the files from the sfh scheduling folder and paste in the project folder along with the utils.
5. The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows:

```
CloudSim.init(num_user, calendar, trace_flag)
```

6. 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 datacenter9883 = new Datacenter(name, characteristics, new
VmAllocationPolicySimple(hostList), s
```

7. The third step is to create a broker:

```
DatacenterBroker broker = createBroker();
```

8. The fourth step is to create one virtual machine unique ID of the VM, userId 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())
```

- #### 9. Submit the VM list to the broker:

broker.submitVmList(vmlist)

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

```
Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize,
utilizationModel, utilizationMode
```

11. Submit the cloudlet list to the broker:

```
broker.submitCloudletList(cloudletList)
```

- ## 12. Start the simulation:

## CloudSim.startSimulation()

13. After the coding part is complete, we need to build path to the cloudsim directory.

14. Right click on the project's parent directory and choose build path, followed by libraries.

15. Add external jars file named "**CloudSim-3.0.jar**" from the CloudSim->jars folder.

16. Click ok and apply, then run the project to get the output.

## Sample Output from the Existing Example:

Starting CloudSimExample1... Initialising...

Starting CloudSim version 3.0 Datacenter\_0 is starting...

Broker is starting... Entities started.

: Broker: Cloud Resource List received with 1 resource(s) 0.0: Broker: Trying to Create VM #0 in Datacenter\_0

: 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

: 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          |       |      |            |             |   |         |   |   |

\*\*\*\*\*Datacenter: Datacenter\_0\*\*\*\*\* User id      Debt

|   |      |
|---|------|
| 3 | 35.6 |
|---|------|

CloudSimExample1 finished!

This is for the in-built program, here we use short job first algorithm and get the required output.

NOTE: The output differs depending on the algorithm we use.

## **Result:**

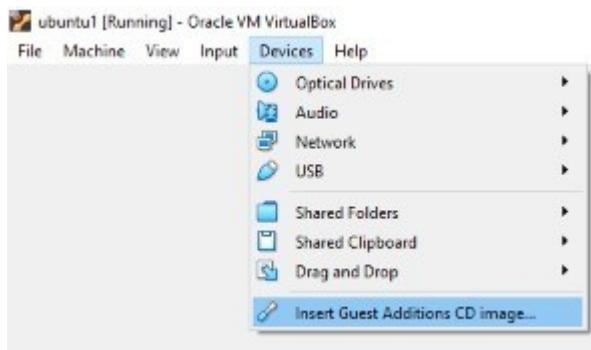
Thus to install and simulate cloud scenario using CloudSim is successfully completed and the scheduling program is executed successfully and the output is verified.

**EX.No:6****Transfer files from One virtual machine to Another****Date: 08/09/2021****Aim:**

To transfer files using various methods from one virtual machine to another virtual machine.

**Procedure:****Installing guest addition**

1. Before copying files the guest additions should be installed to transfers files from one operating system to another. Click on the **MenuBar->Devices->Insert Guest Additions CD image**.
2. After mounting the disk install the program, using the terminal.
3. Restart the virtual machine (ubuntu). Now it is possible to share files and folders between the virtual machines.

**Various file and folder transfer methods:**

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.

4. 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.

5. You can mount part of the file system of a virtual machine via NFS or SSHFS on the other, or you can share file and directory with Samba.

You may find interesting the article Sharing files between guest and host without VirtualBox shared folders with detailed step by step instructions.

You should remember that you are dialling with a little network of machines with different operative systems, and in particular:

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

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). 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. 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.

- It is suggested that to enable Drag & drop to be cosy with the Windows machines and the Shared folders or to be cosy with Linux.

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 to transfer files using various methods from one virtual machine to another virtual machine by guest mode installation is successfully completed.

**EX.No:7****Launch Virtual Machine using OpenStack****Date: 09/09/2021****Aim:**

To launch virtual machine using trystack (Online Openstack Demo Version).

**OpenStack:**

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). 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.

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.

**TryStack.org Homepage**

The Easiest Way To Try Out OpenStack. We've set up a large, growing cluster of hardware running OpenStack on x86. The best part? It's totally free for you to try & test your apps—thanks to our generous individual and corporate contributors.

For A Free Account:

Join Our Facebook Group

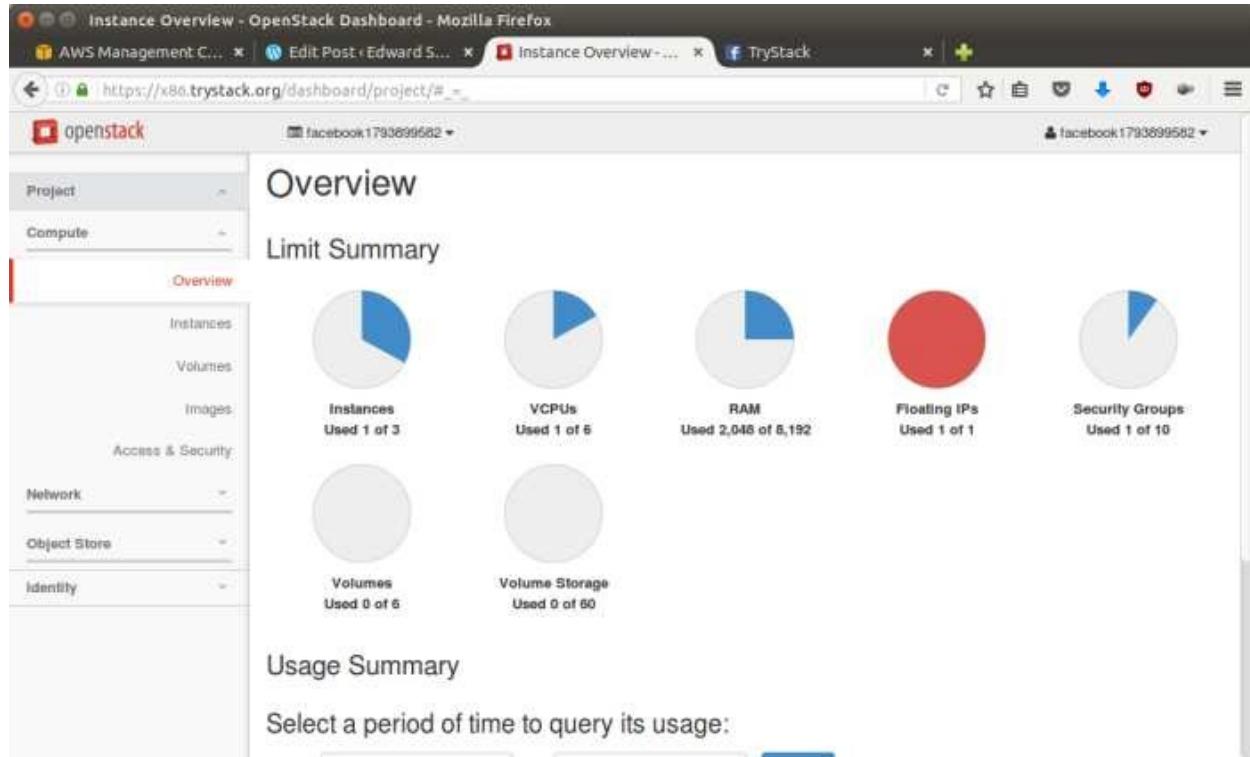
Once we approve your account...  
Try out OpenStack  
OpenStack RDO Liberty on x86/RHEL Login  
Or Learn About Using The API

Testing only, please.  
Rule No. 1: Remember that TryStack is designed exclusively as a testing sandbox. We wanted a fast, easy way for developers to test code against a real OpenStack environment, without having to stand up hardware themselves. It probably goes without saying that this is not the place for production code - you should host only test code and test servers here. In fact, your account on TryStack will be periodically wiped to help make sure no one account tries to rule permanently over our hardware. Play nice in the sandbox!

I assume that you already join to the Facebook Group and login to the dashboard. After you log in to the TryStack, you will see the Compute Dashboard like:

## OpenStack Compute Dashboard

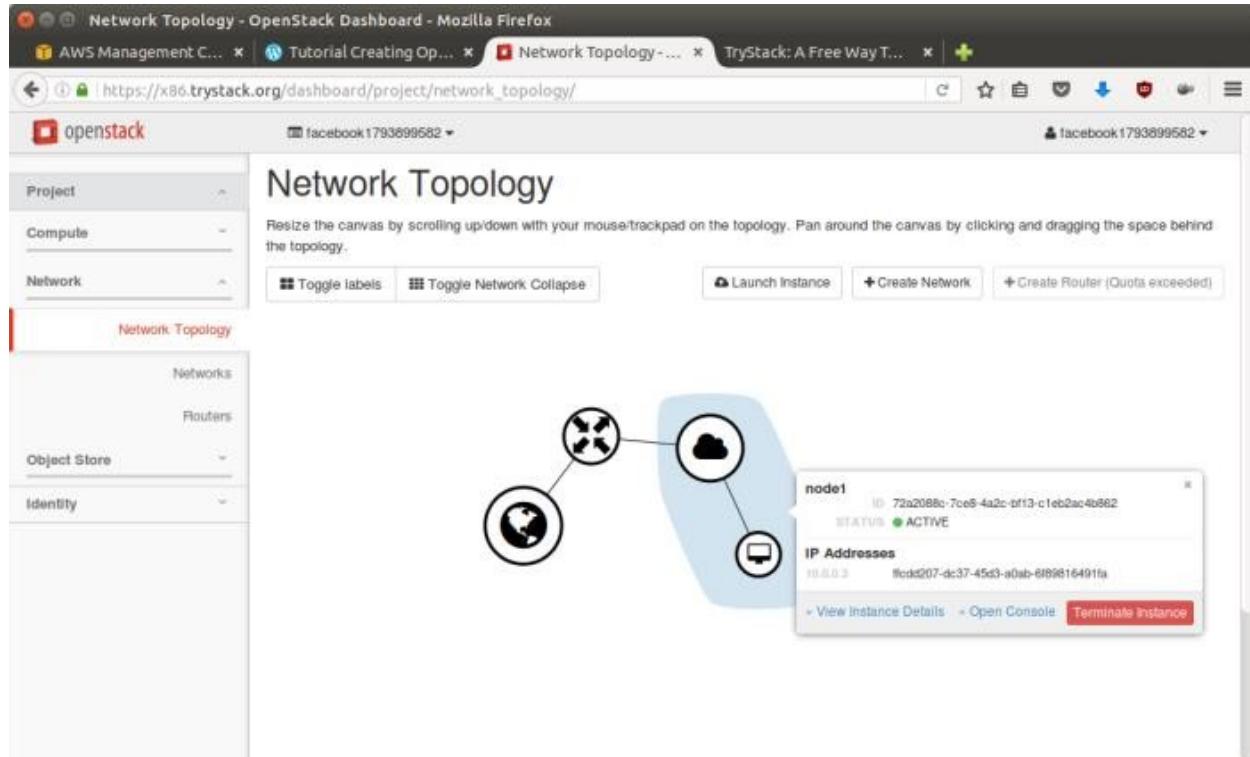
### Overview: What we will do?



In this post, I will show you how to run an OpenStack instance. The instance will be accessible through the internet (have a public IP address). The final topology will like:

### Network topology

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



**Before starting:**

Open firefox (any browser) and launch <https://192.168.10.100/dashboard>

Login with the respective User Credentials.

**Step 1: Create Network**

The network in here is our own local network. So, your instances will be not mixed up with the others. We can imagine this as our 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 we want to create multiple instances, step 1-5 should be repeated. Here we 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 the 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 have been created in Step 1.
  2. Click Add interface.
6. Go to Network > Network Topology. We 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 the instance as 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). We need to request a public (floating) IP address to be assigned to your instance.

1. Go to Compute > Instance.
2. In one of the 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 we 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 our 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, we can SSH the instances to the floating IP address that we got in the step 4. If we are using Ubuntu image, the SSH user will be ubuntu.

### **Result:**

Thus, to launch virtual machine using trystack(Online Openstack Demo Version) is successfully completed and the output is verified.

**Ex.No:8**

## Installing Hadoop

**Date: 13/09/2021****Aim:**

To install Hadoop single node cluster and run simple application like word count on Linux.

**Procedure:**

Install Hadoop

Step 1: Download the hadoop-install.sh and run the file in terminal using the

**command:** *cd ~/Downloads && chmod +x hadoop-install.sh && ./hadoop-install.sh*

The code downloads the Hadoop from internet and installs in the system.

Step 2: To make sure that Java and Hadoop have been properly installed on our system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

**Command:** *java -version*

```
edureka@localhost:~$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Checking Java Version

**Command:** *hadoop version*

```
edureka@localhost:~$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be
5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Checking Hadoop Version

Step 3: Edit the Hadoop Configuration files.

**Command:** *cd hadoop-2.7.3/etc/hadoop/*

**Command:** *ls*

All the Hadoop configuration files are located in *hadoop-2.7.3/etc/hadoop* directory as we can see in the snapshot below:

```
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml httpfs-env.sh mapred-env.sh
configuration.xsl httpfs-log4j.properties mapred-queues.xml.template
container-executor.cfg httpfs-signature.secret mapred-site.xml.template
core-site.xml httpfs-site.xml slaves
hadoop-env.cmd kms-acls.xml ssl-client.xml.example
hadoop-env.sh kms-env.sh ssl-server.xml.example
hadoop-metrics2.properties kms-log4j.properties yarn-env.cmd
hadoop-metrics.properties kms-site.xml yarn-env.sh
hadoop-policy.xml log4j.properties yarn-site.xml
hdfs-site.xml mapred-env.cmd
```

Fig: Hadoop Installation – Hadoop Configuration Files

Step 4: Open *core-site.xml* and edit the property mentioned below inside configuration tag: *core-site.xml* informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

**Command:** *vi core-site.xml/*

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

Fig: Hadoop Installation – Configuring core-site.xml

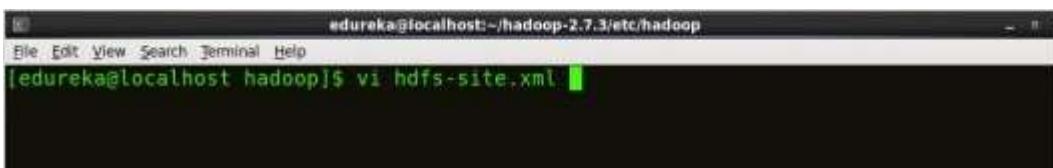
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <configuration>
4 <property>
5 <name>fs.default.name</name>
6 <value>hdfs://localhost:9000</value>
7 </property>
```

#### Step 5:

Edit hdfs-site.xml and edit the property mentioned below inside.

configuration tag: hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

Command: *vi hdfs-site.xml*



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

Fig: Hadoop Installation – Configuring hdfs-site.xml

**Step 6: Edit the mapred-site.xml file and edit the property mentioned below inside configuration tag:**

`mapred-site.xml` contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

Command: `cp mapred-site.xml.template mapred-site.xml`

```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop:
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ cp mapred-site.xml.template mapred-site.xml
[edureka@localhost hadoop]$ █
```

Command: *vi mapred-site.xml*.

```
[edureka@localhost ~]$ vi mapred-site.xml
```

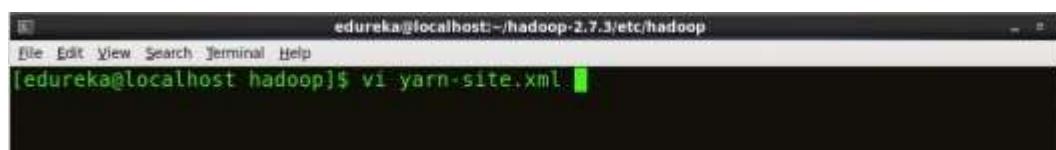
```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Fig: Hadoop Installation – Configuring mapred-site.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?xmlstylesheet type="text/xsl"
4 href="configuration.xsl"?>
5
6 <configuration>
7 <property>
8 <name>mapreduce.framework.name</name>
9 <value>yarn</value>
10 </property>
11 </configuration>
```

Step 7: Edit yarn-site.xml and edit the property mentioned below inside configuration tag: yarn-site.xml contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

Command: *vi yarn-site.xml*



```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

Fig: Hadoop Installation – Configuring yarn-site.xml

```

1
2
3 <?xml version="1.0">
4 <configuration>
5 <property>
6 <name>yarn.nodemanager.aux-services</name>
7 <value>mapreduce_shuffle</value>
8 </property>
9 <property>
10 <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</ name>
11 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
12 </property>
13 </configuration>
14
15
16

```

Now, the hadoop has been configured.

Step 8: Now download the run-cluster.sh file, and run it in terminal.

Command: `cd ~/Downloads && chmod +x run-cluster.sh && ./run-cluster.sh`

During the run-time the script might ask for passwords multiple times.

```

Starting all the processes....
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
v@localhost's password:
localhost: starting namenode, logging to /home/v/Desktop/hadoop/logs/hadoop-v-namenode-v-VirtualBox.out
v@localhost's password:
localhost: starting datanode, logging to /home/v/Desktop/hadoop/logs/hadoop-v-datanode-v-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
v@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /home/v/Desktop/hadoop/logs/hadoop-v-secondarynamenode-v-VirtualBox.out
starting yarn daemons
starting resourcemanager, logging to /home/v/Desktop/hadoop/logs/yarn-v-resourcemanager-v-VirtualBox.out
v@localhost's password:
localhost: starting nodemanager, logging to /home/v/Desktop/hadoop/logs/yarn-v-nodemanager-v-VirtualBox.out
Checking the process

```

The run-cluster.sh script automatically installs and checks all the configurations and at the end Firefox(or any other available browser) will be opened.

The screenshot shows the HDFS Health Overview page for a single-node cluster. The top navigation bar includes tabs for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities.

**Overview 'localhost:9000' (active)**

<b>Started:</b>	Fri Sep 10 02:01:06 IST 2021
<b>Version:</b>	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
<b>Compiled:</b>	2016-08-18T01:41Z by root from branch-2.7.3
<b>Cluster ID:</b>	CID-31a0fac7-eb6c-4210-b7cd-5d3333545a7f
<b>Block Pool ID:</b>	BP-726291907-127.0.1.1-1631219459946

### Summary

Security is off.  
Safemode is off.  
7 files and directories, 0 blocks = 7 total filesystem object(s).  
Heap Memory used 31.72 MB of 65.88 MB Heap Memory. Max Heap Memory is 966.69 MB.  
Non Heap Memory used 39.97 MB of 40.84 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

<b>Configured Capacity:</b>	838.94 GB
<b>DFS Used:</b>	28 KB (0%)
<b>Non DFS Used:</b>	51.31 GB

## Result:

Thus, to install hadoop single node cluster and run simple application like wordcount is successfully completed and the output is verified successfully.