



**JKK MUNIRAJAH COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

T.N PALAYAM-638 506.



Approved by AICTE, New Delhi and Affiliated to Anna University.

RECORD NOTE BOOK

Reg.No

Certified that this is the bonafide record of work done by
Selvan/Selvi..... of the Semester
..... branch during the year,
in the Laboratory
With the laboratory code.....

Staff -in-charge

Head of the Department

Submitted for the University Practical Examination on _____.

Internal Examiner

External Examiner

CCS335 – CLOUD COMPUTING

S.NO	DATE	NAME OF THE EXPERIMENT	PAGE NO.,	MARKS	SIGN
1		Install Virtual box/VMware/ Equivalent open source cloud Workstation with different flavours of Linux or Windows OS on top of windows 8 and above.			
2		Install a C compiler in the virtual machine created using a virtual box and execute Simple Programs			
3		Install Google App Engine. Create a hello world app and other simple web applications using python/java.			
4		Use the GAE launcher to launch the web applications.			
5		Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.			
6		Find a procedure to transfer the files from one virtual machine to another virtual machine.			
7		Install Hadoop single node cluster and run simple applications like word count.			
8		Creating and Executing Your First Container Using Docker.			
9		Run a Container from Docker Hub			

CONTENT BEYOND THE SYLLABUS

10		Creating Virtual Machine Using Open Nebula.			
----	--	---	--	--	--

Ex No : 1	Install Virtual box/VMware/ Equivalent open source cloud Workstation with different flavours of Linux or Windows OS on top of windows 8 and above.
DATE :	

AIM:

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

Procedure:

Step 1- Download Link

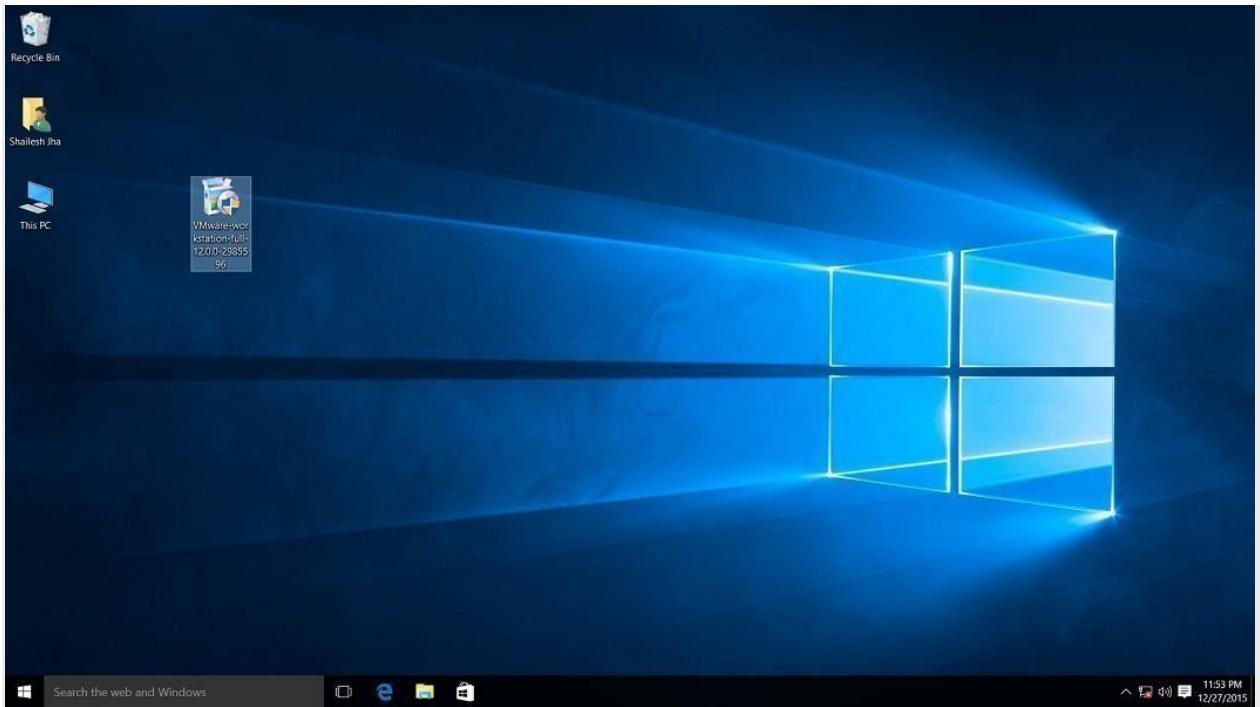
Link for downloading the software is <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>. Download the software for windows. Good thing is that there is no signup process. Click and download begins. Software is around 541 MB.

Step 2- Download the installer file

It should probably be in the download folder by default, if you have not changed the settings in your browser. File name should be something 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. But for now, till the next version is available, they will all be VMware Workstation 15 Pro.

Step 3- Locate the downloaded installer file

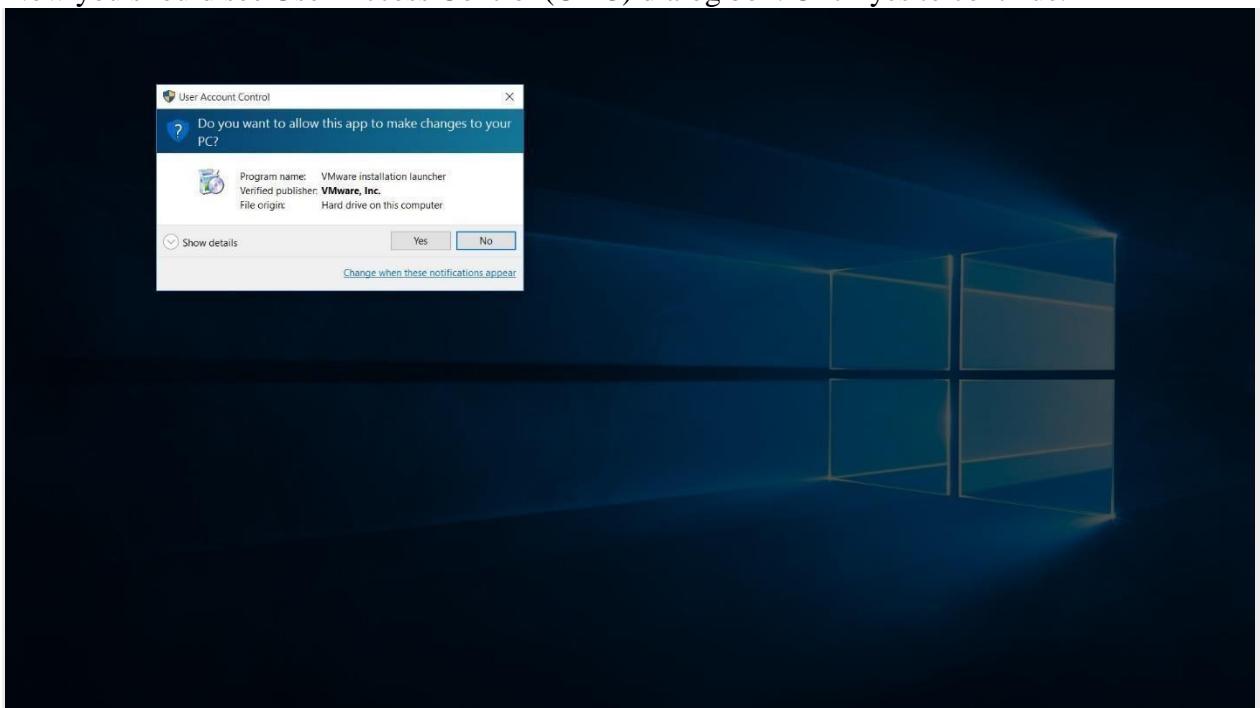
For demonstration purpose, I have placed the downloaded installer on my desktop. Find the installer on your system and double click to launch the application.



VMware workstation 15 pro for windows 10 installer file screenshot.

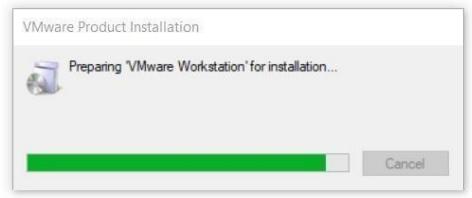
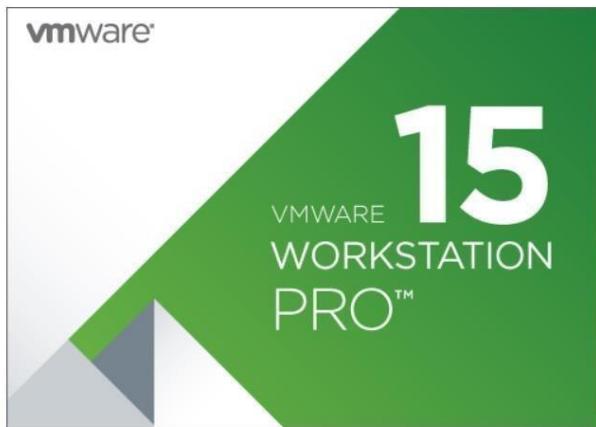
Step 4- User Access Control (UAC) Warning

Now you should see User Access Control (UAC) dialog box. Click yes to continue.



VMware Workstation 12 Pro installer windows 10 UAC screenshot

Initial Splash screen will appear. Wait for the process to complete.



VMware Workstation 15 Installation Splash Screen

Step 5- VMware Workstation Setup wizard

Now you will see VMware Workstation setup wizard dialog box. Click next to continue.



VMware Workstation Pro Setup



Welcome to the VMware Workstation Pro Setup Wizard

The Setup Wizard will upgrade VMware Workstation Pro on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright 1998-2018 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at:

<http://www.vmware.com/go/patents>

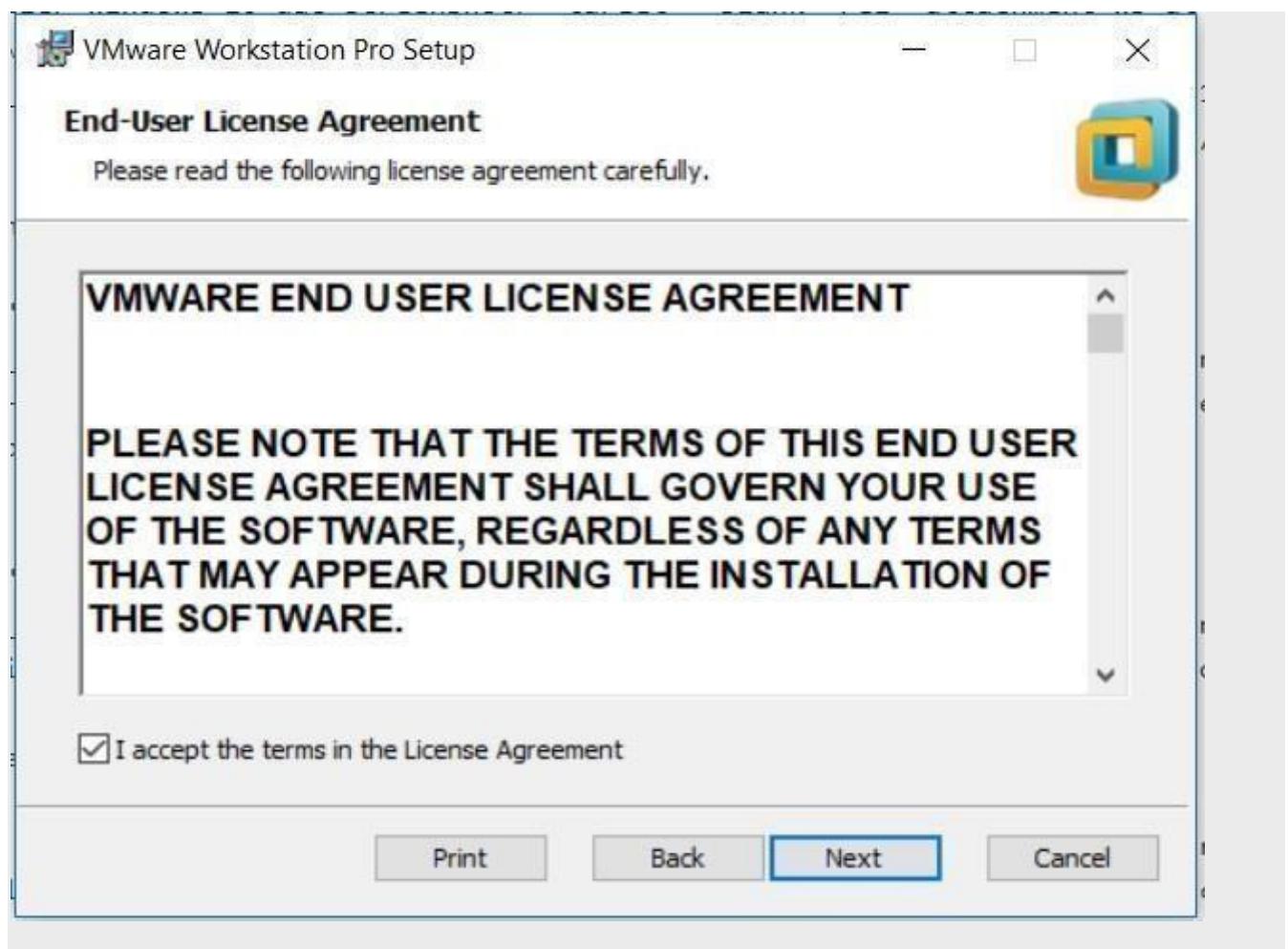
Next

Cancel

VMware Workstation 15 Installation – Setup Wizard

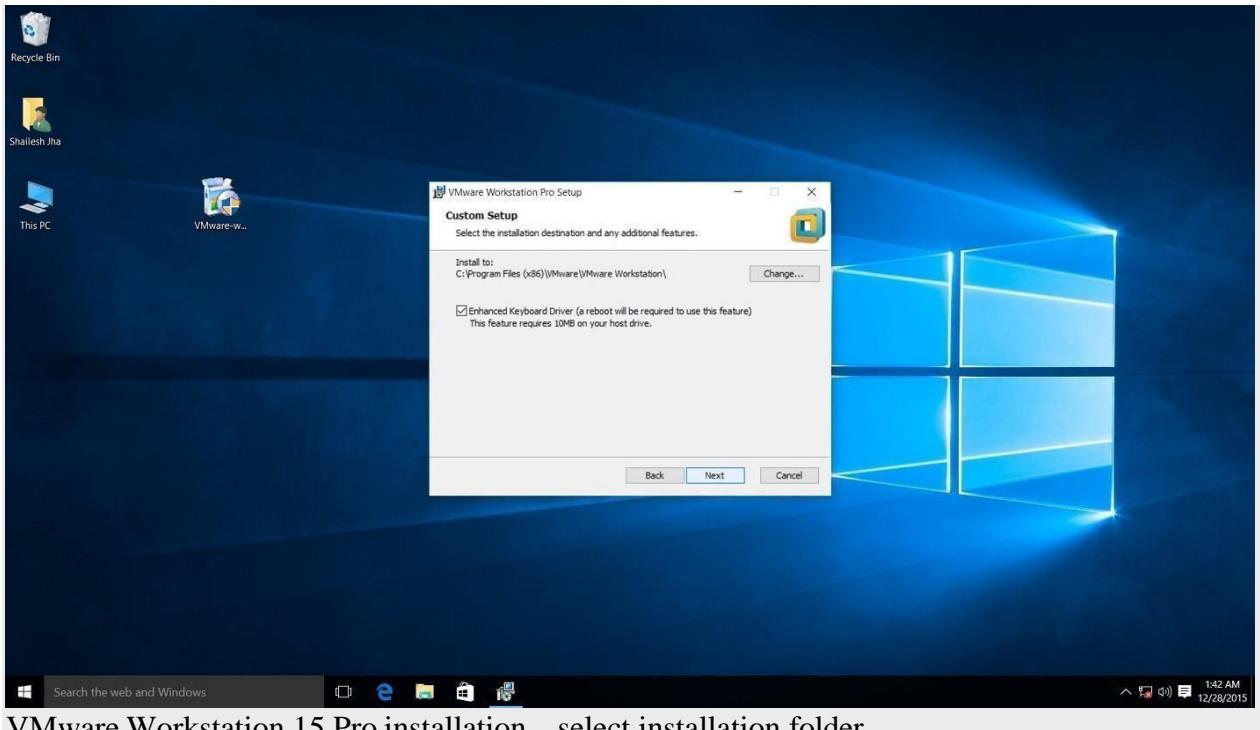
Step 6- End User Licence Agreement

This time you should see End User Licence Agreement dialog box. Check “I accept the terms in the Licence Agreement” box and press next to continue.



Step 7- Custom Setup options

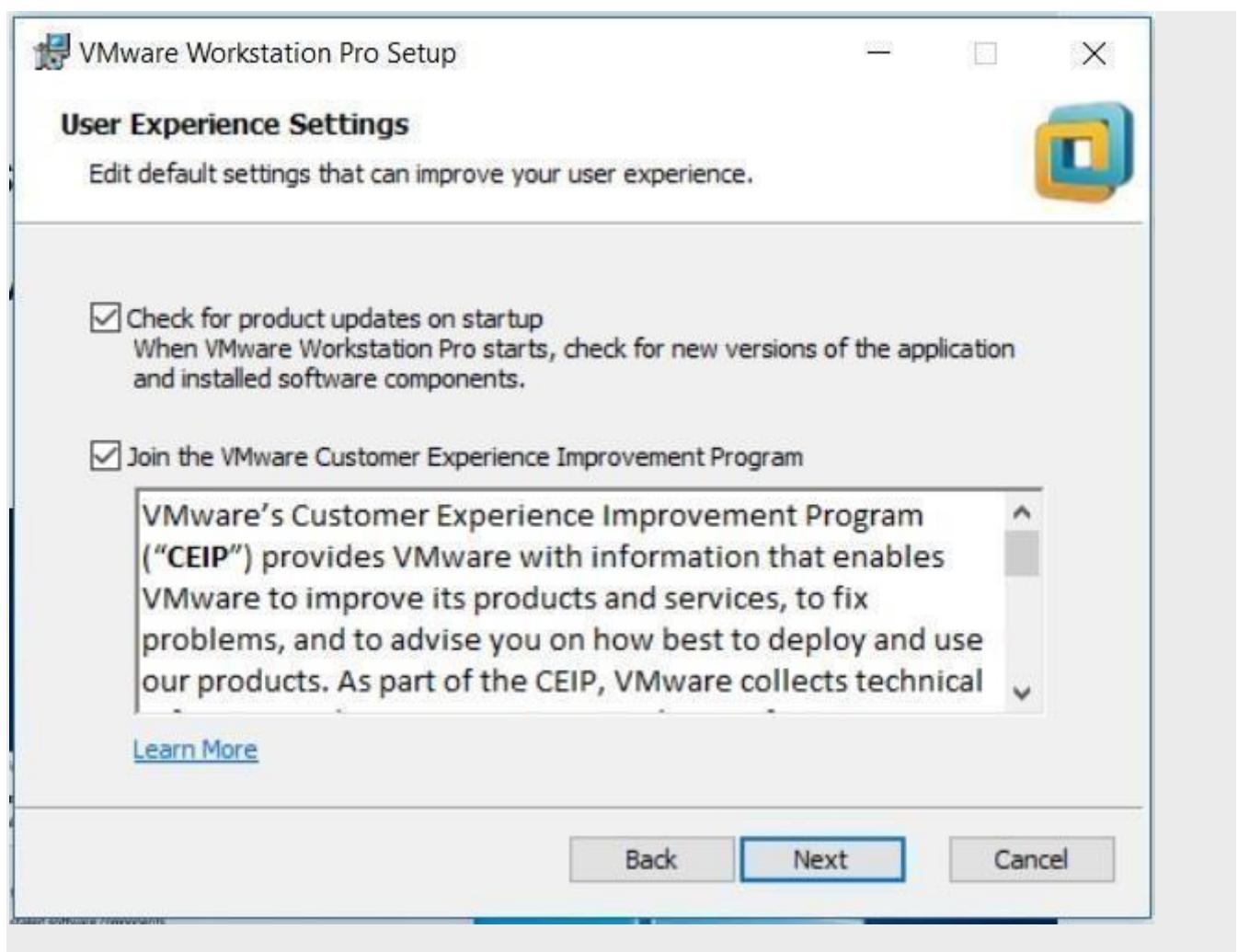
Select the folder in which you would like to install the application. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.



VMware Workstation 15 Pro installation – select installation folder

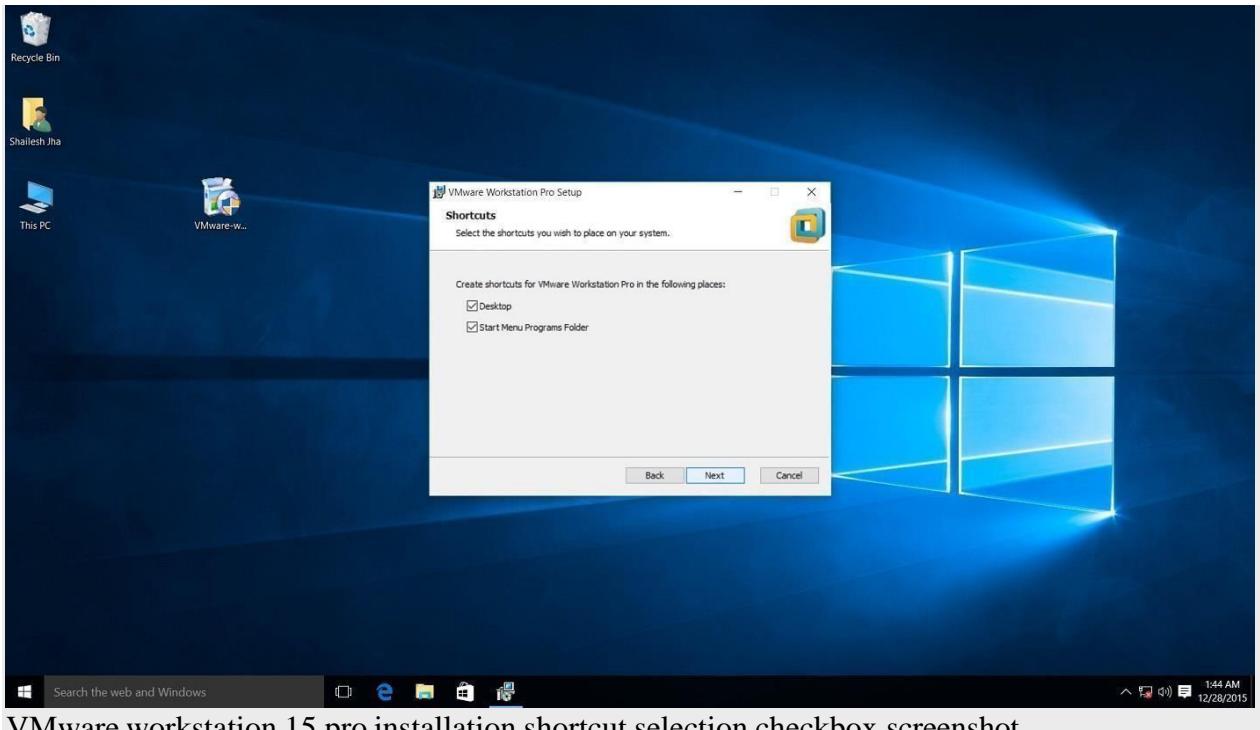
Step 8- User Experience Settings

Next you are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”. Do as You wish. I normally leave it to defaults that are unchecked.



Step 9- Application Shortcuts preference

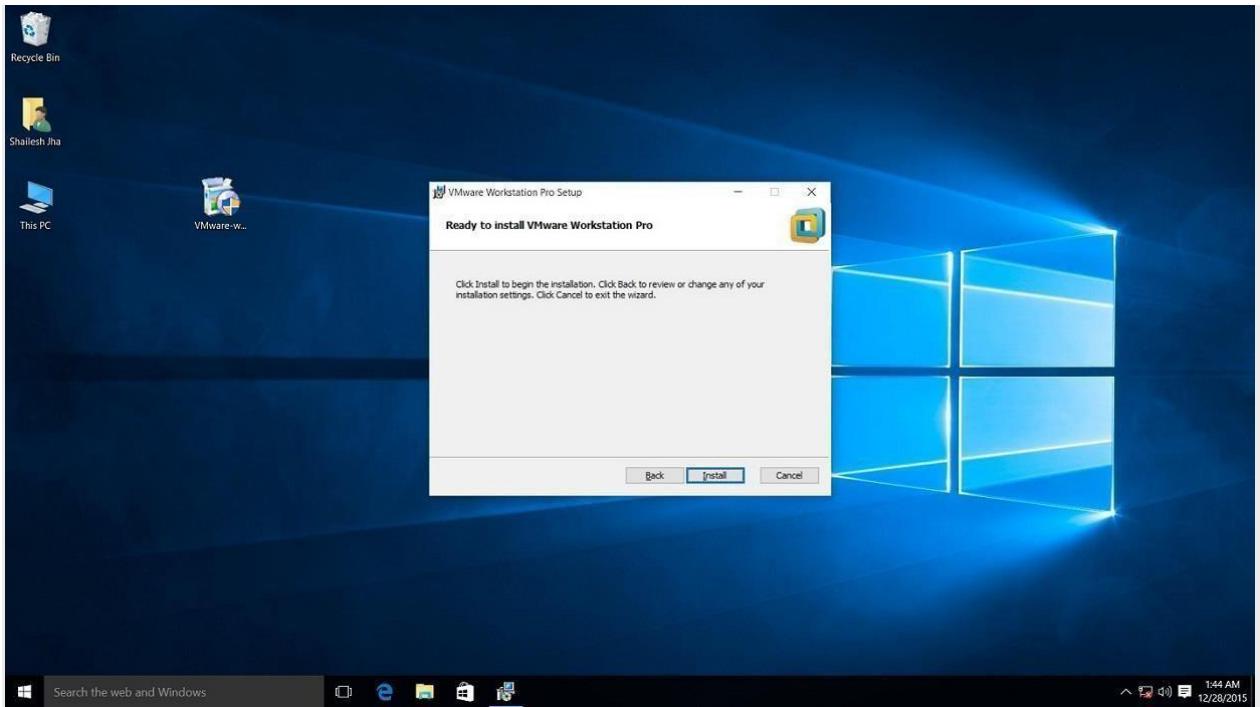
Next step is to select the place you want the shortcut icons to be placed on your system to launch the application. Please select both the options, desktop and start menu and click next.



VMware workstation 15 pro installation shortcut selection checkbox screenshot.

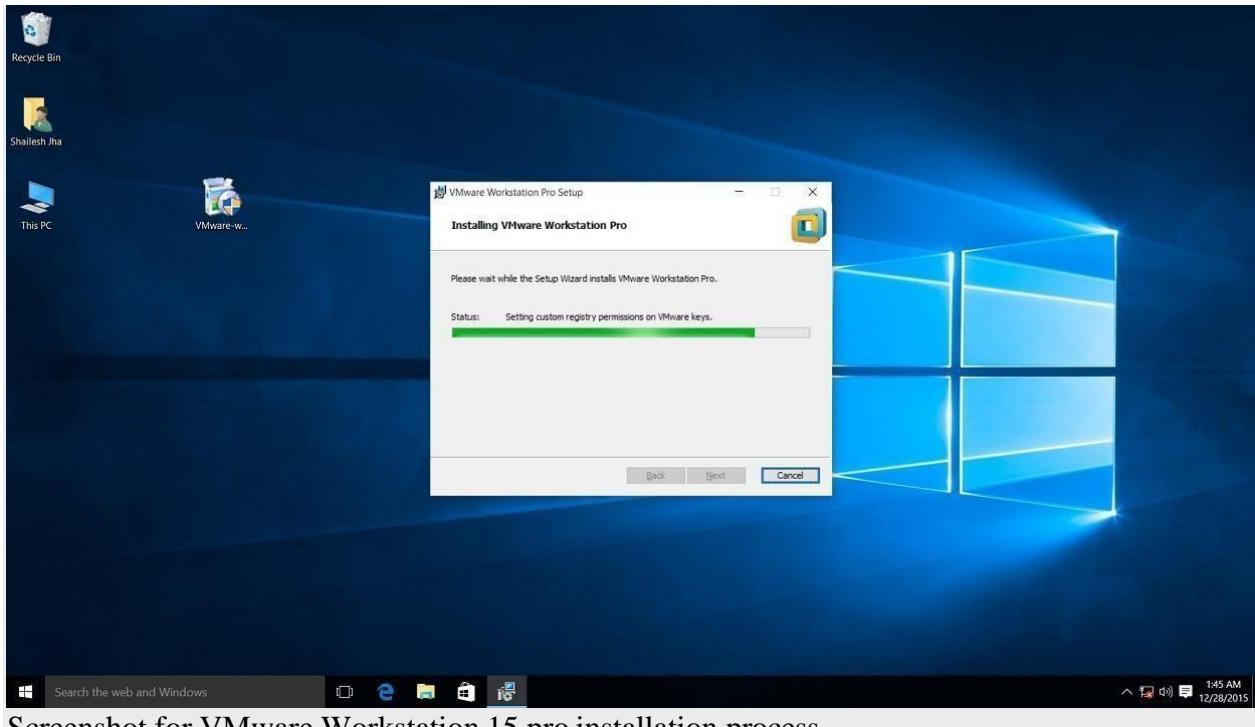
Step 10- Installation begins

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



Screenshot for VMware Workstation 15 pro installation begin confirmation dialog box on windows 10.

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



Screenshot for VMware Workstation 15 pro installation process.

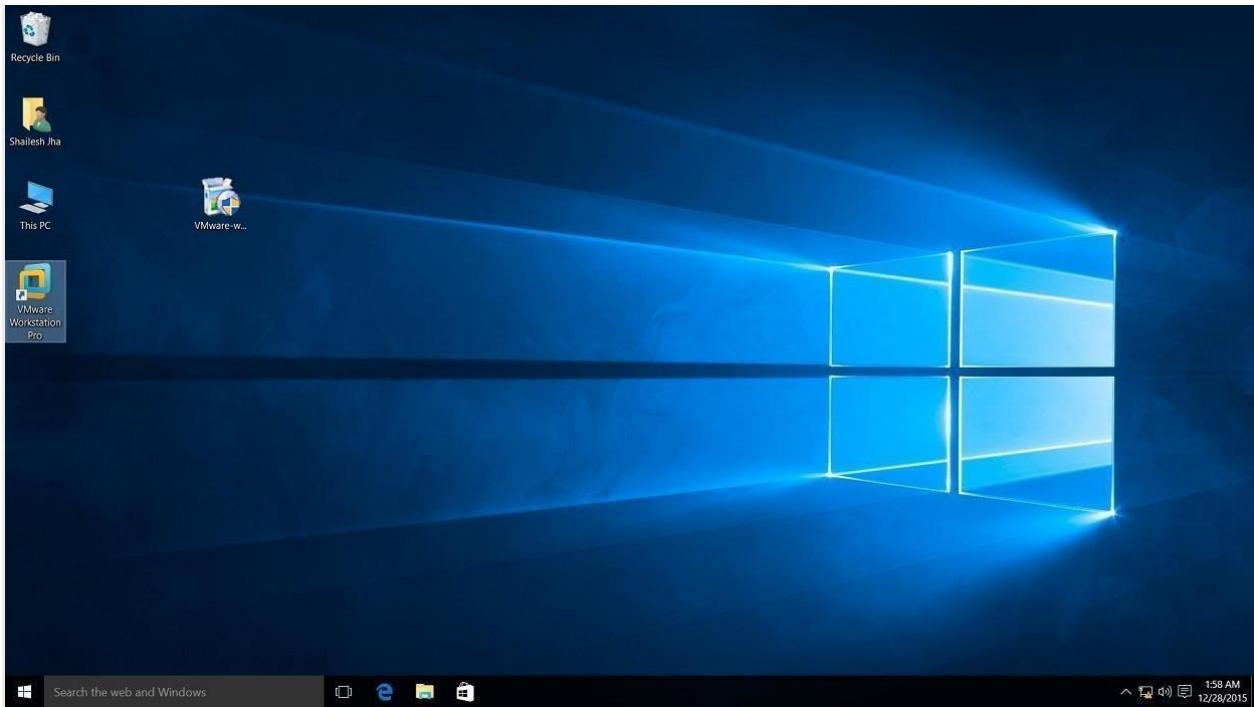
At the end you will see installation complete dialog box. Click finish and you are done with the Installation process. You may be asked to restart your computer. Click on Yes to restart.



VMware Workstation 15 Installation – Installation Complete

Step 11- Launch VMware Workstation

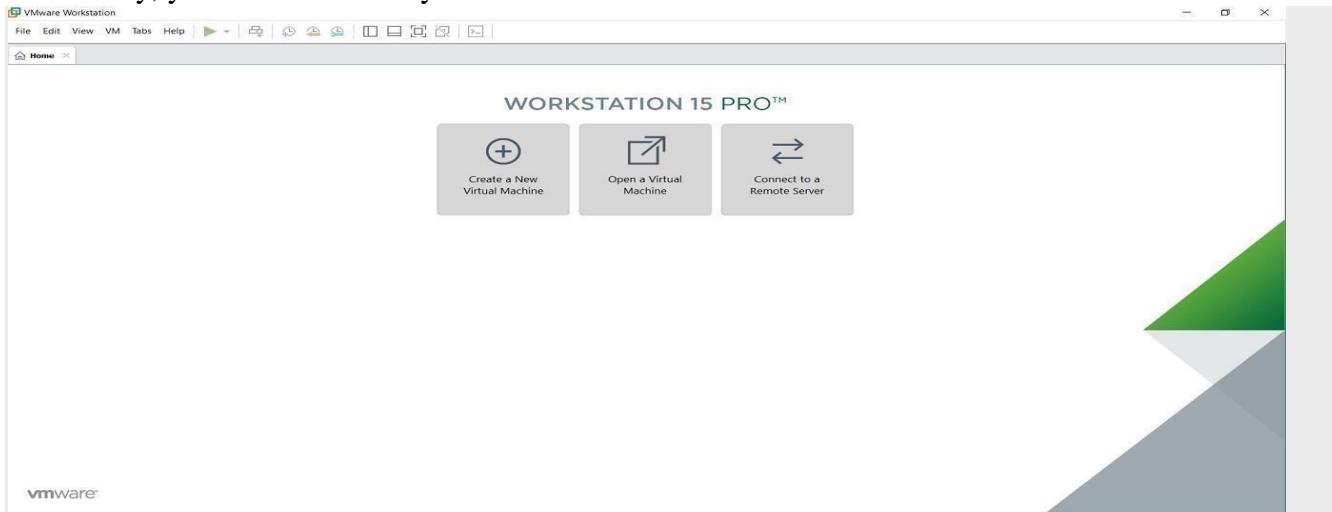
After the installation completes, you should see VMware Workstation icon on the desktop. Double click on it to launch the application.



Screenshot for VMware Workstation 15 Pro icon on windows 10 desktop.

Step 12- License Key

If you see the dialog box asking for license key, click on trial or enter the license key. Then what you have is the VMware Workstation 15 Pro running on your windows 10 desktop. If don't have the license key, you will have 30 days trial.



VMware Workstation 15 Pro home screen

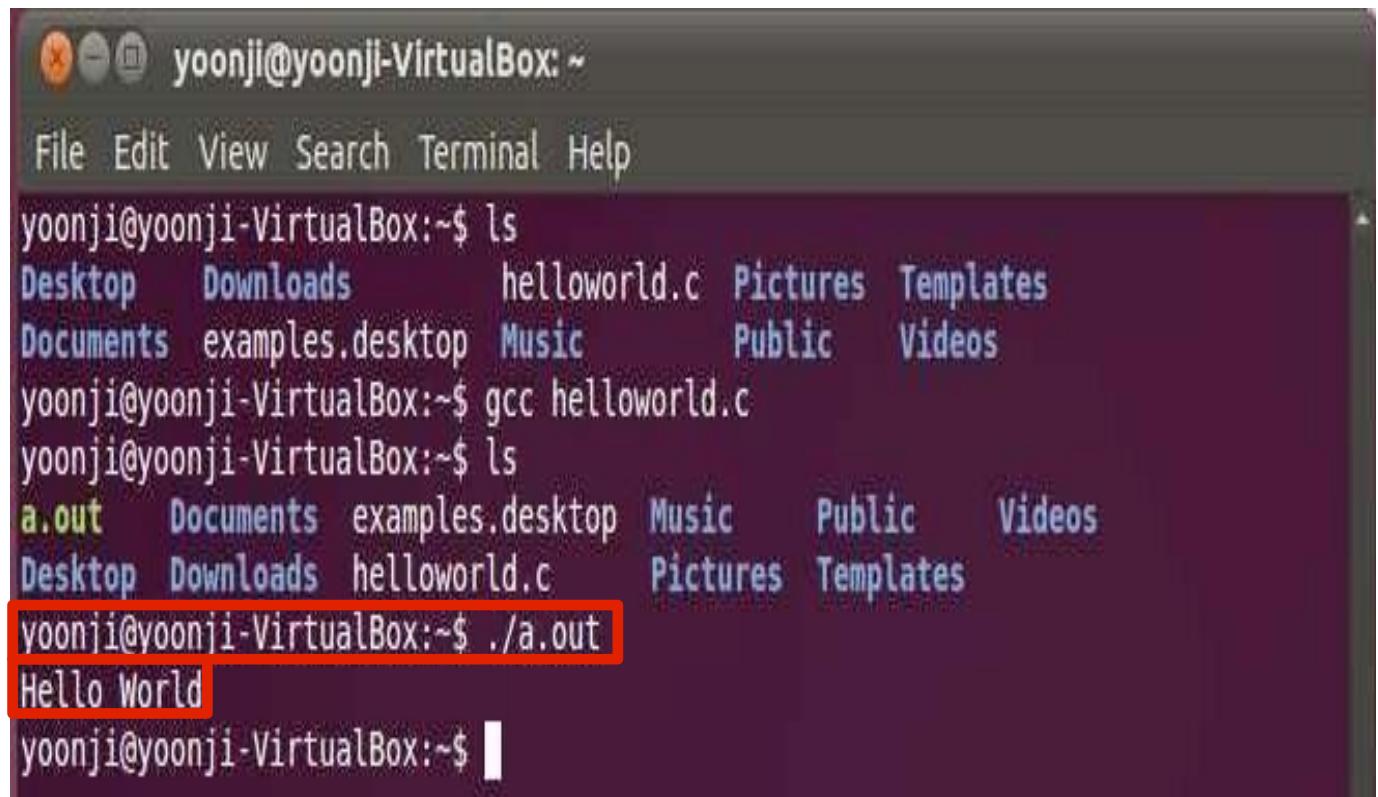
Step 13- At some point if you decide to buy

At some point of time if you decide to buy the License key, you can enter the License key by going to

Help->Enter a License Key You can enter the 25 character license key in the dialog box shown below and click OK. Now you have the license version of the software.

C Programming on Linux

1. Type “./a.out” on Terminal to run the program
2. If you see “Hello World” on the next line, you just successfully ran your first C program!
3. Try other codes from “A Shotgun Introduction to C” on professor Edwards’s webpage. You can also find many C programming guides online.(just Google it!) Enjoy :)



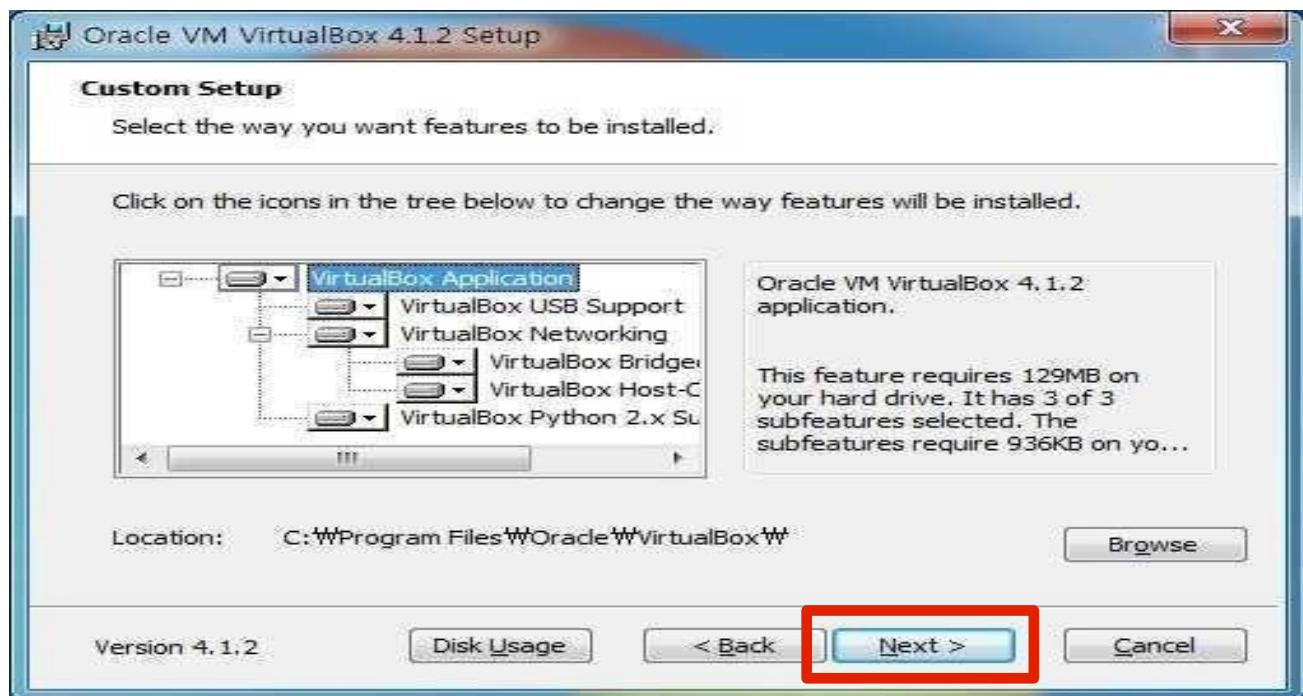
The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are window control icons (close, minimize, maximize) followed by the text "yoonji@yoonji-VirtualBox: ~". Below this is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the following command-line 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:~$
```

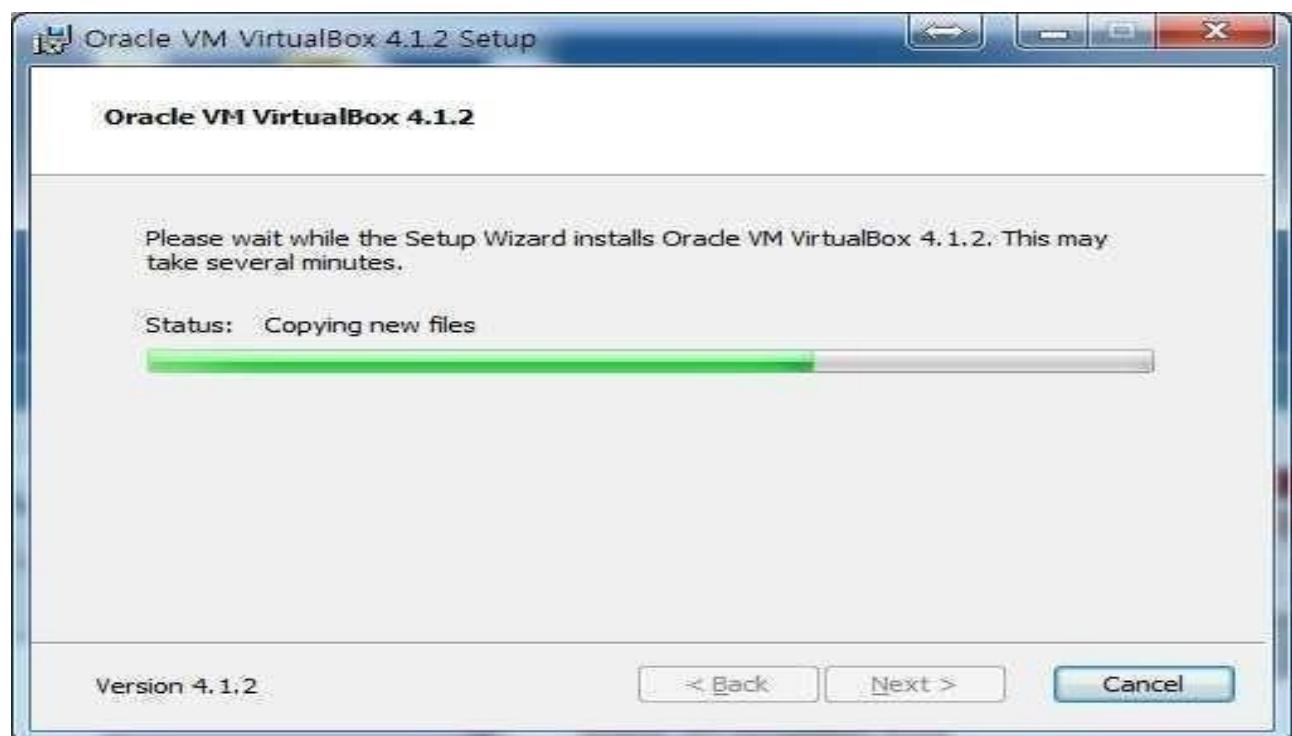
The command `./a.out` is highlighted with a red rectangle. The output "Hello World" is also highlighted with a red rectangle.

InstallVirtual Box

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



4. Click continue and finish installing Virtual Box



5. When finished installation,close the window

Download Linux

I. Visit the page <http://www.ubuntu.com/download/ubuntu/download>

2. Choose the Latest version of Ubuntu and 32-bit and click "Start Download"

The screenshot shows the Ubuntu download page. At the top, there are navigation links: Download, Windows Installer, Alternative downloads, CDs, Upgrade, and 下载 Ubuntu. Below these, a large orange button labeled '1 Download Ubuntu' is visible. To the right of the button, a red box highlights the 'Download options' section. This section contains two dropdown menus: one for 'Ubuntu 11.04 - Latest version' and another for '32-bit (recommended)'. To the right of this section is a large orange button with the text 'Download started' and 'Ubuntu 11.04 32-bit'. An arrow labeled 'CLI' points from the text 'Choose the Latest version of Ubuntu and 32-bit and click "Start Download"' to the 'Download started' button. Below the main section, there are three columns: 'Additional options', 'If you're running Windows', and 'Other ways to get Ubuntu'. The 'Additional options' column includes a link to 'previous versions and alternative downloads'. The 'If you're running Windows' column includes links to 'Ubuntu Windows installer' and 'Order CDs'. The 'Other ways to get Ubuntu' column includes links to 'Ubuntu Server'.

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

If you're running Windows

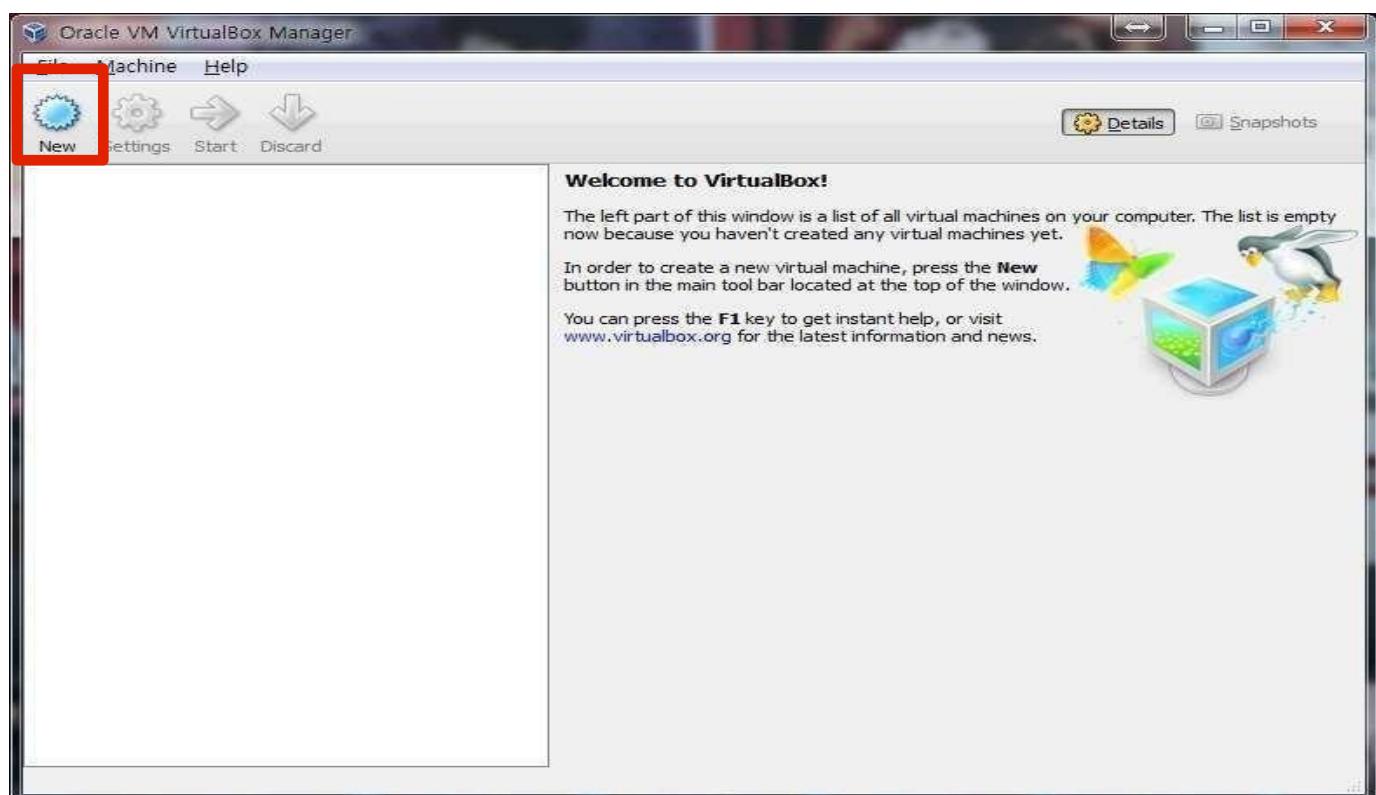
Other ways to get Ubuntu

Order CDs >

Ubuntu Server >

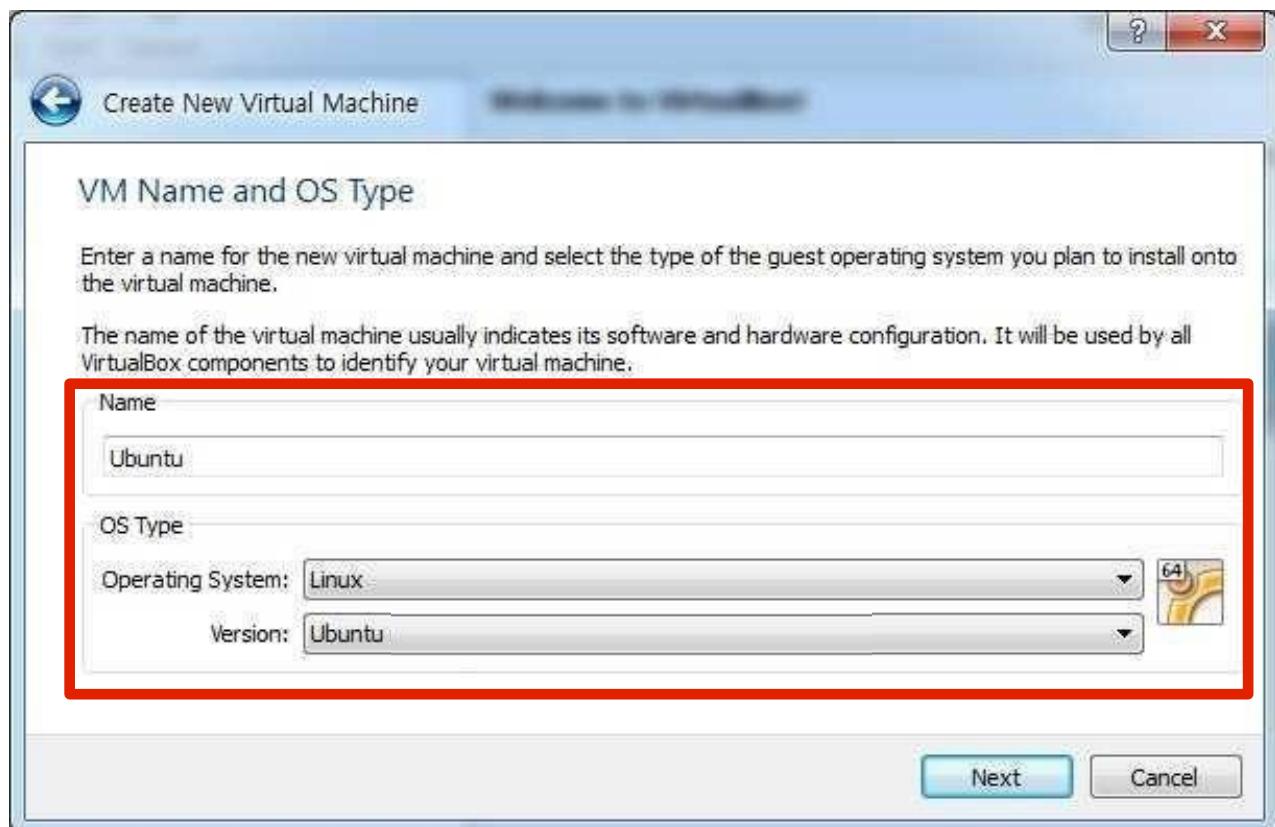
Install Linux using Virtual Box

1. Run Virtual Box by double-clicking the icon
2. Click “New” button on the top left corner



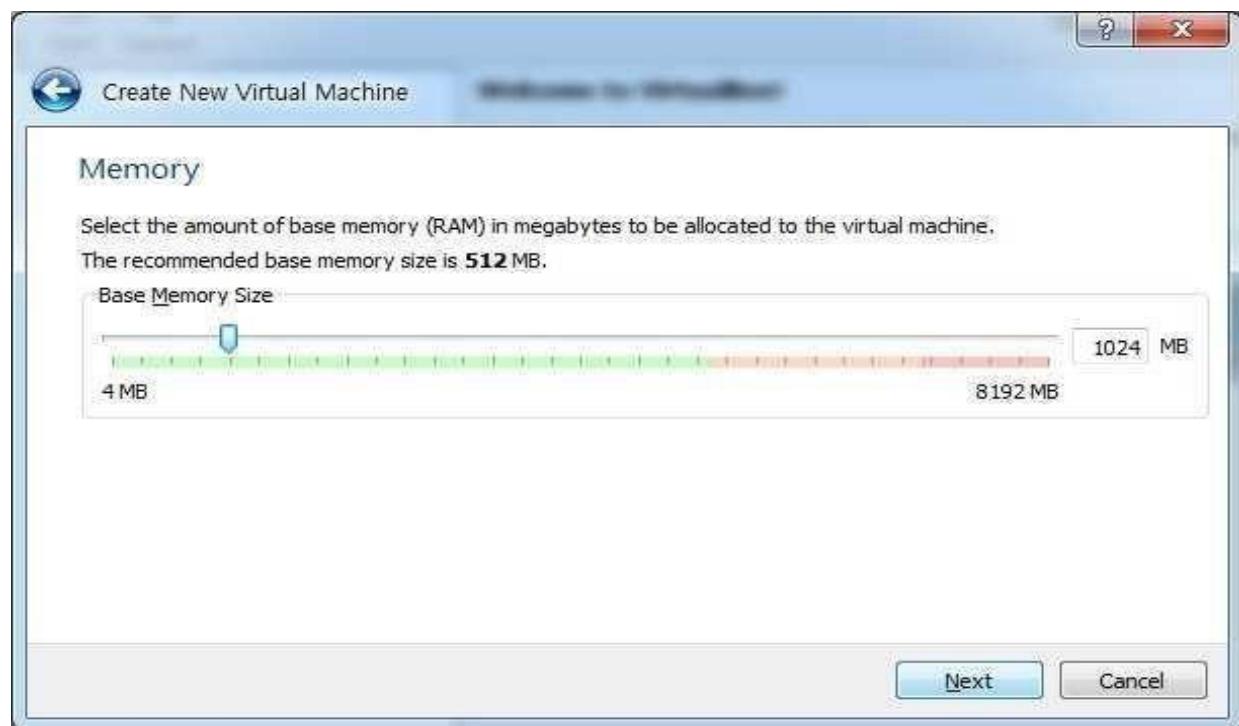
3. Click “Continue” on the pop-up window

4. Type VM name, select “Linux” for the OS and choose “Ubuntu” for the version.



5. Choose the amount of memory to allocate (I suggest Choosing between 512 MB to 1024 MB)

6. Click Continue or Next



7. Choose create a new virtual hard disk

8. Click Continue or Next



9. Choose VDI (Virtual Box Disk Image)

10. Click Continue or Next

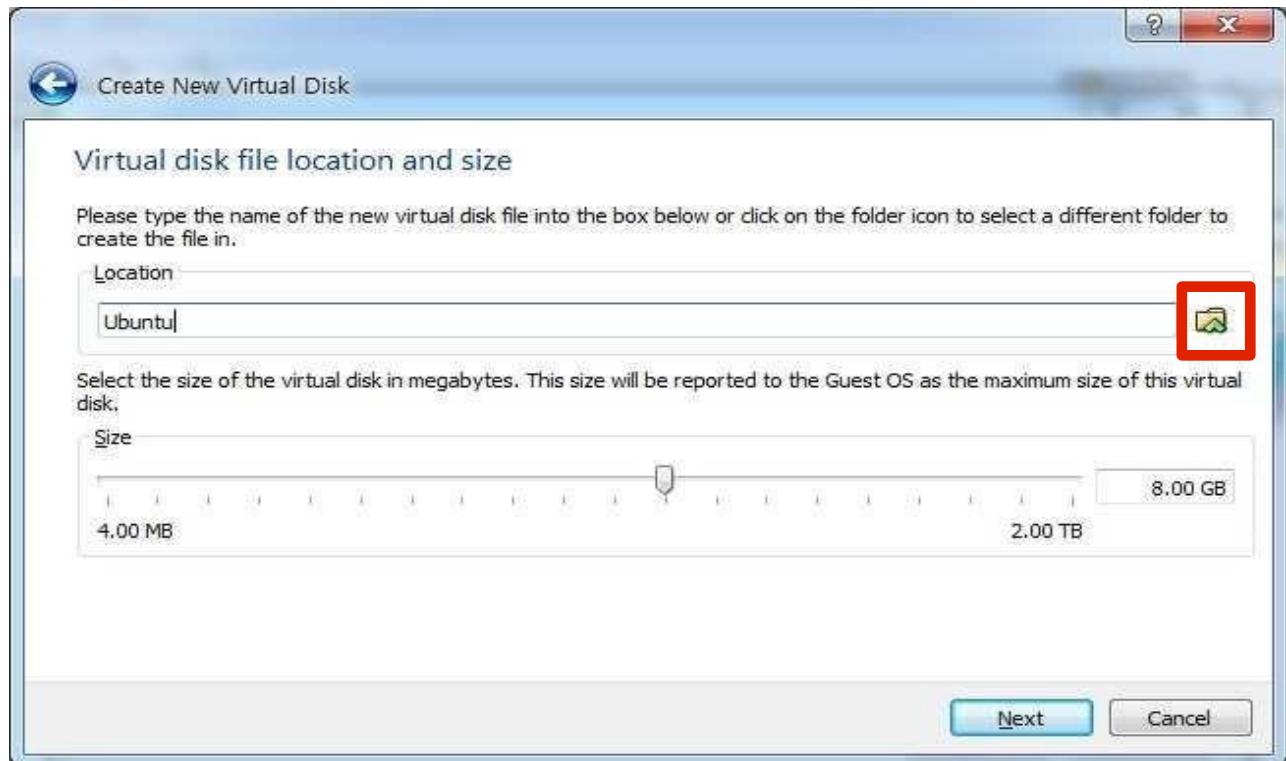


11. Choose “Dynamically Allocated” click continue. This way, the size of your Virtual Hard Disk will grow as you use.



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

I3. Select the size of the Virtual Disk (I recommend choosing 8GB) and click continue



I4. Click Create

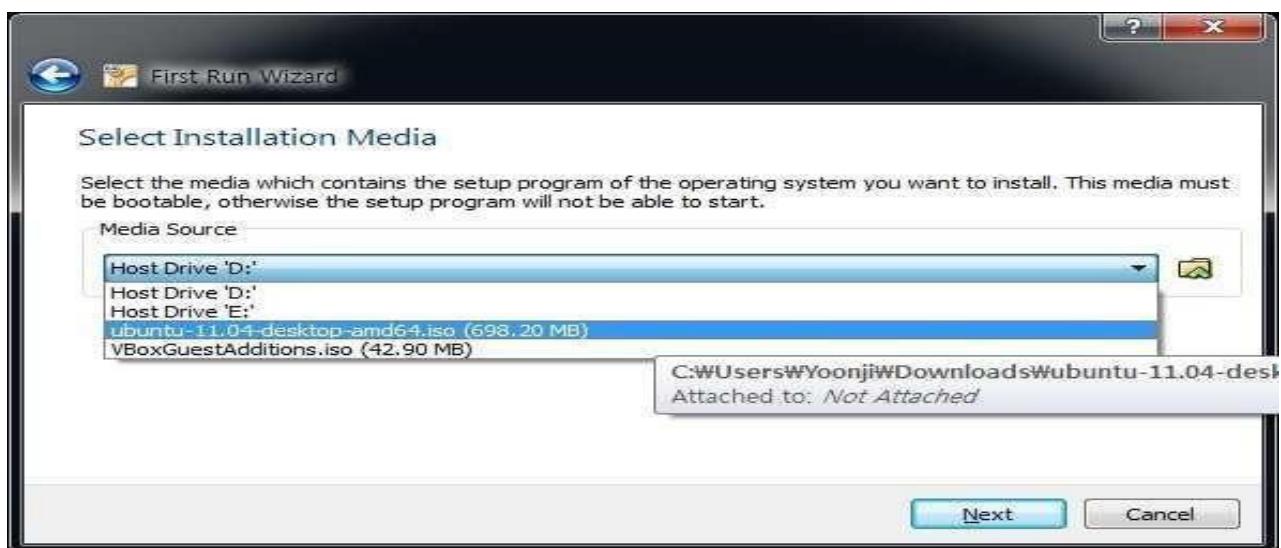


Running Linux

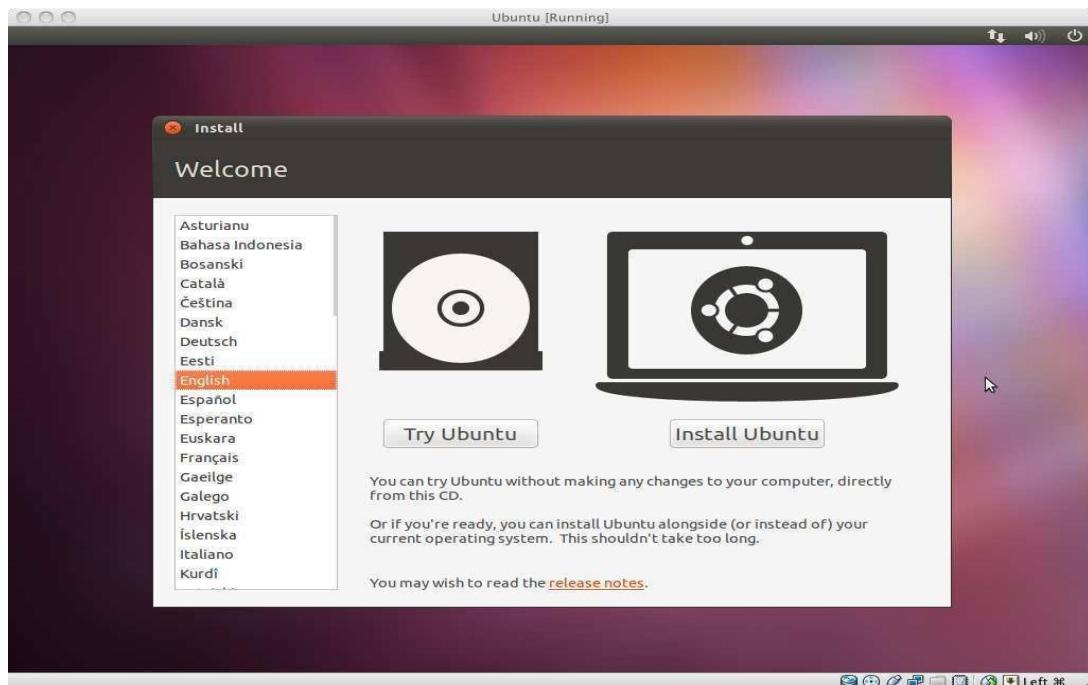
1. Choose Ubuntu from left column and click Start



2. Click the folder icon and choose the ubuntu I so file you downloaded and click continue and start



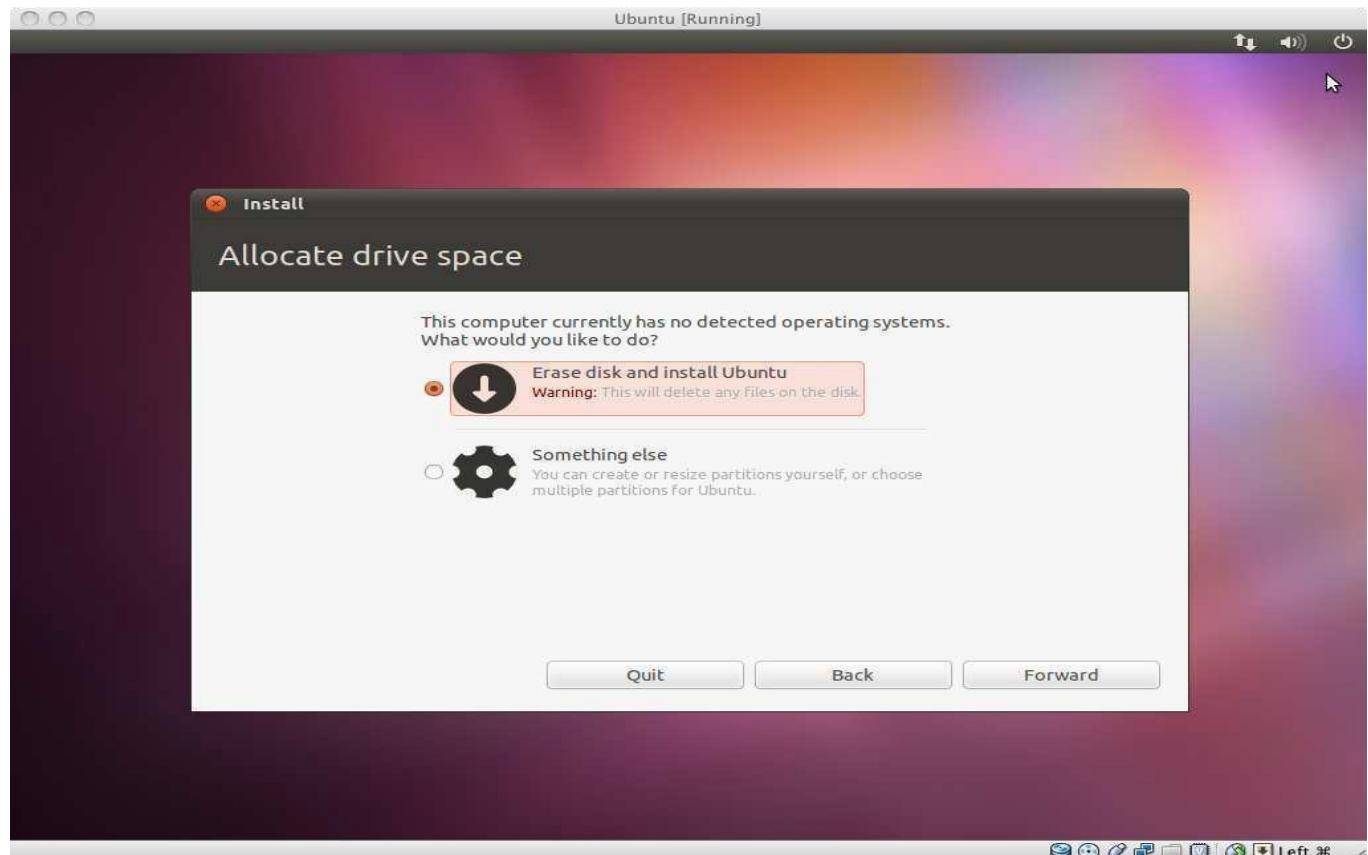
1. 1.Click Install Ubuntu



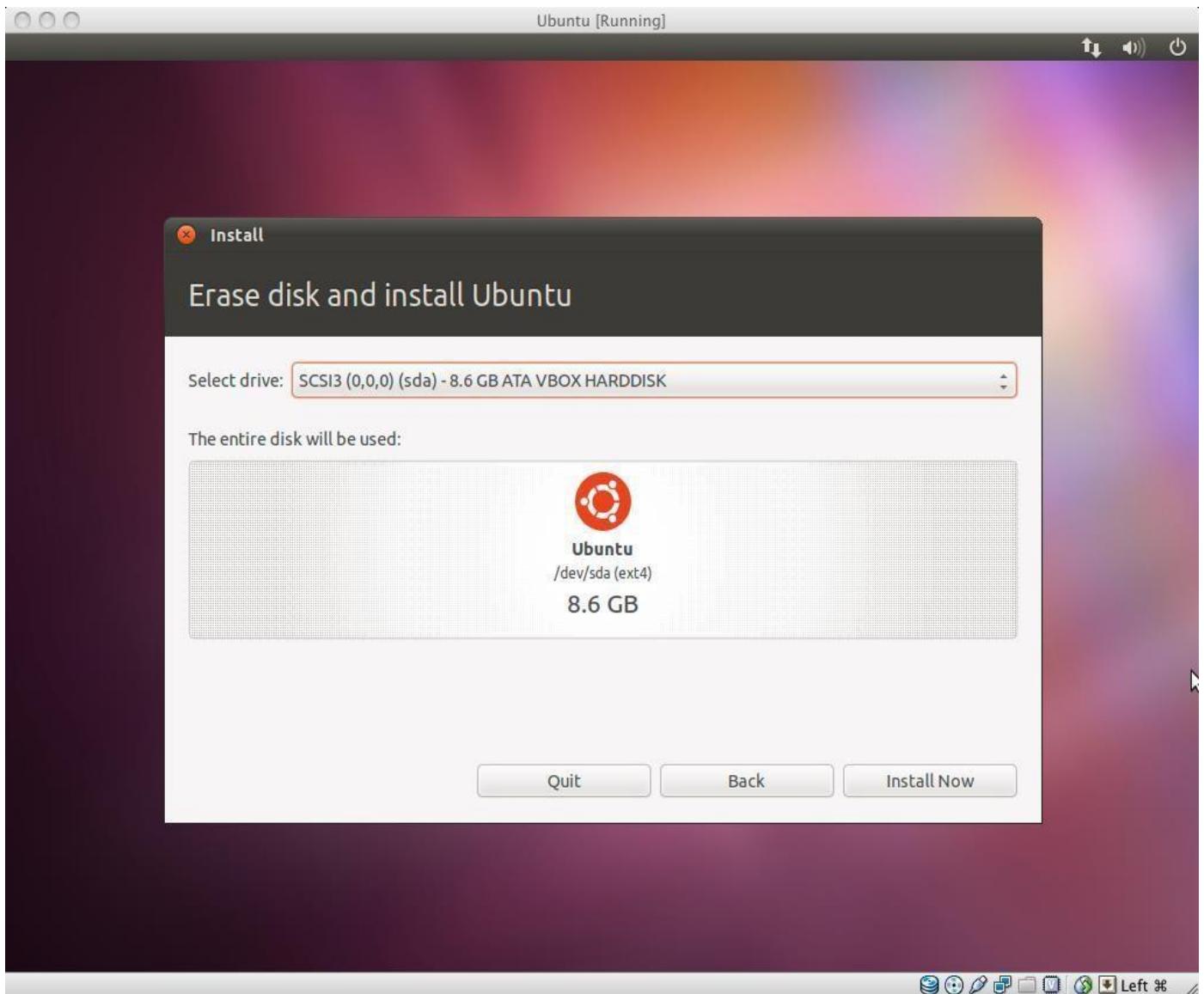
2. Check “Download updates” and click Forward



3. Choose “Erase disk and install Ubuntu” and click Forward (Don’t worry ,it won’t wipe your computer

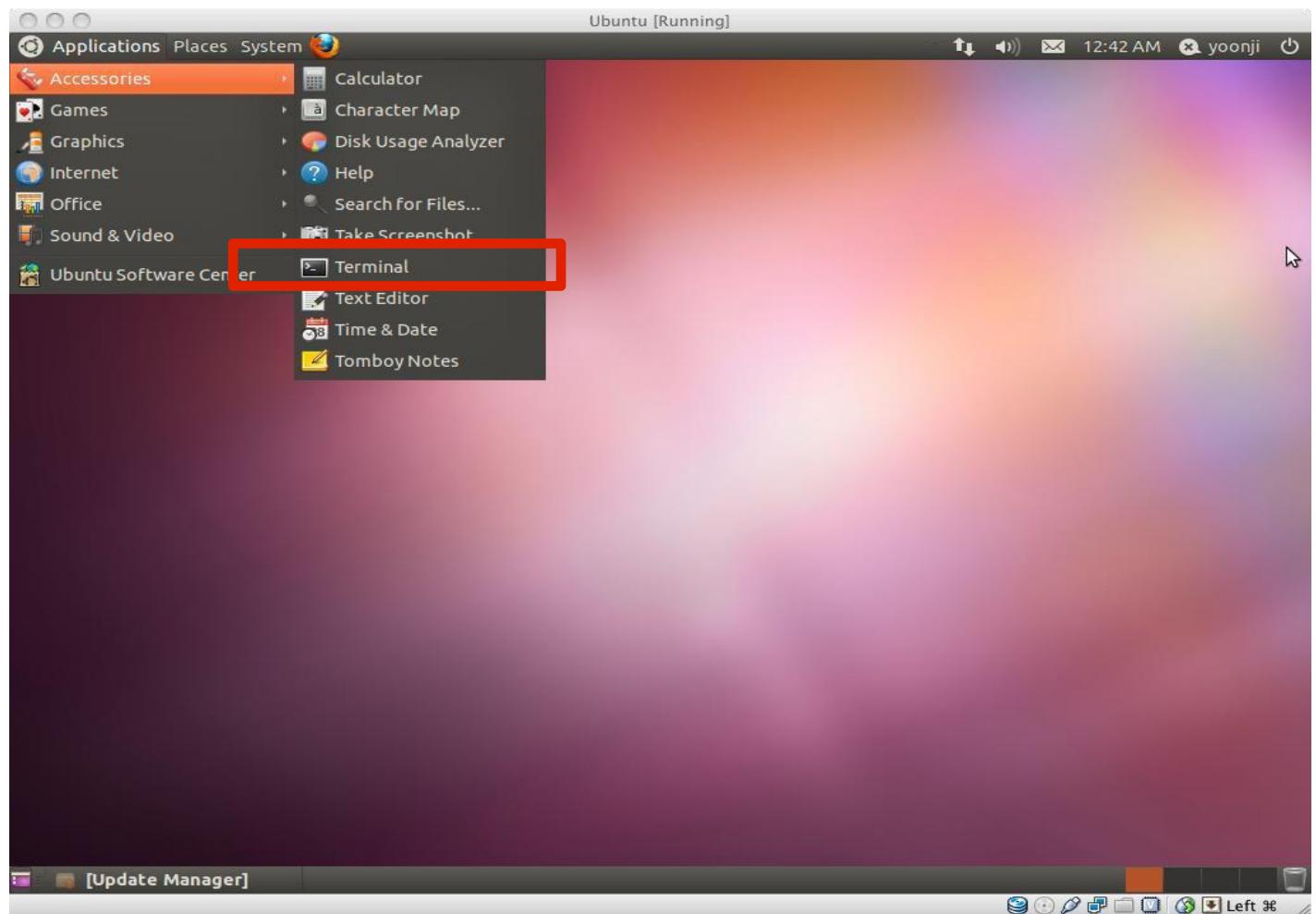


4. Click "Install Now"



5. When finished, click Restart and press Enter.

1. Open Terminal (Applications-Accessories-Terminal)



RESULT:

Thus the Installation of Virtual box/VMware Workstation with different flavors of Linux or windows OS On virtual machine is created.

Ex No : 2	Install a C compiler in the virtual machine created using a virtual box and execute Simple Programs
DATE :	

AIM:

To install a C compiler in the virtual machine and execute a sample program.

Procedure:

Step1:

Install the centos or Ubuntu in the open nebula as per previous commands.

Step 2:

Login into the VM of installed OS.

Step 3:

If it is ubuntu then, for gcc installation

```
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install gcc-6 gcc-6-base
```

Step 4:

Write a sample program

```
likeWelcome.cpp
#include<iostream.h>
using namespace
std;int main()
{
cout<<"Hello
world";return 0;
}
```

Step 5:

First we need to compile and link our program. Assuming the source code is saved in a file welcome.cpp, we can do that using GNU C++ compiler g++, for example
g++ -Wall -o welcome welcome.cpp

And output can be executed by ./welcome

<https://www.iitk.ac.in/nt/faq/vbox.htm>

Result:

Thus the GCC compiler has been successfully installed and executed a sample program.

Ex No: 3	Install Google App Engine. Create a hello world app and other simple web applications using python/java.
DATE :	

Aim:

To Install Google App Engine and Create hello world app and other simple web application using python.

Procedure:

Use **Notepad** to create a **Google App Engine (GAE) Python** project (hello world example), run it locally, and deploy it to Google App Engine account.

Tools used:

1. Python 2.7
2. Google Cloud SDK

Note

GAE supports Python 2.7 version to Current version

P.S Assume Python 3.7 are installed.

1. Install Google Cloud SDK

The Google Cloud CLI works on Windows 8.1 and later and Windows Server 2012 and later. Download the [Google Cloud CLI installer](#).

Alternatively, open a Power Shell terminal and run the following Power Shell commands:

```
(New-Object Net.WebClient).DownloadFile
("https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe",
"$env:Temp\GoogleCloudSDKInstaller.exe")
```

```
& $env:Temp\GoogleCloudSDKInstaller.exe
```

2. Write Python & app.yaml code using Notepad ++

Figure – Open Notepad ++ File >> New (Ctrl+N)

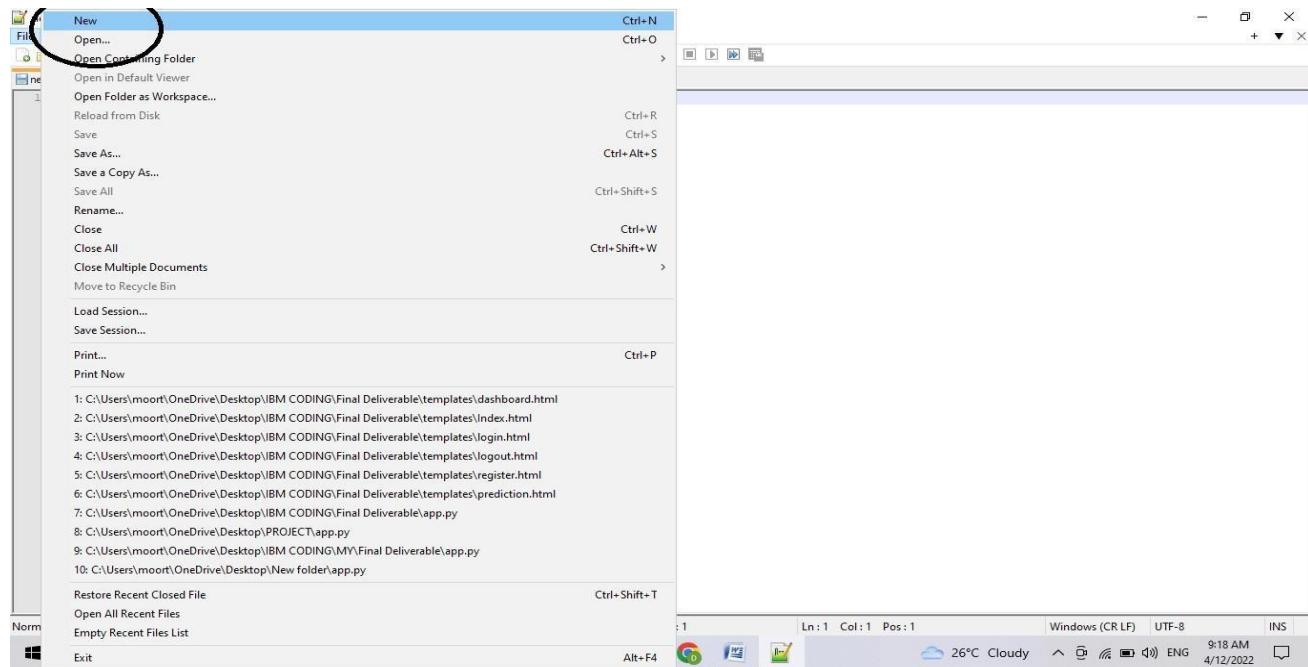


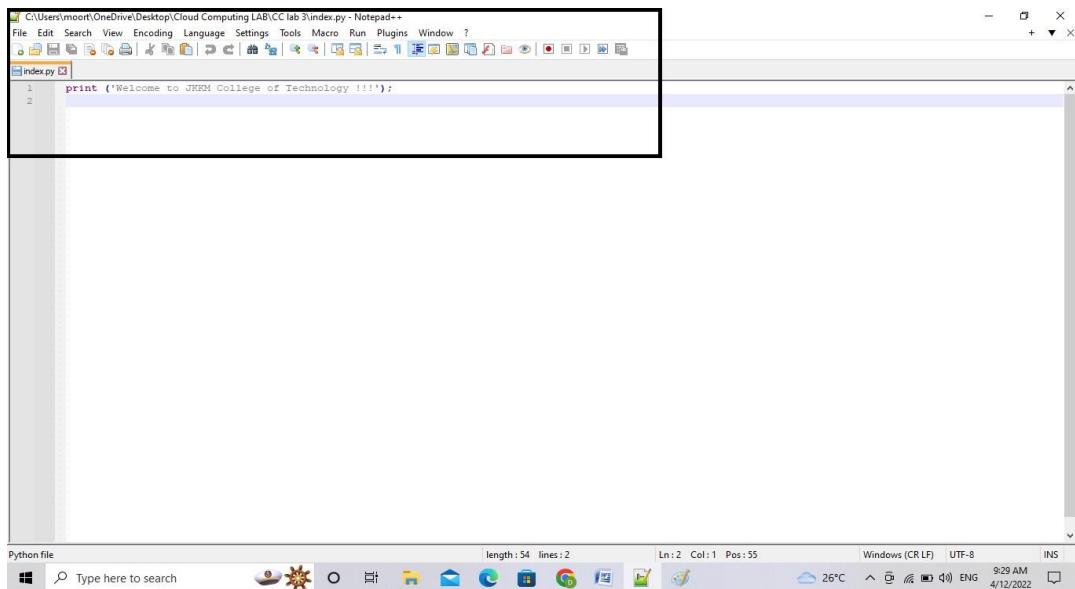
Figure – app.yaml code

```
version: 1
runtime: python27
api_version: 1
threadsafe: false

handlers:
- url: /.*
  script: index.py
```

A screenshot of the Notepad++ application window showing the contents of the 'app.yaml' file. The code defines a Python application with version 1, runtime 'python27', and threadsafe 'false'. It includes a single handler for all URLs ('/.*') that points to the 'index.py' script. The status bar at the bottom indicates the file has 111 lines and 9 characters per line.

Figure – Python code



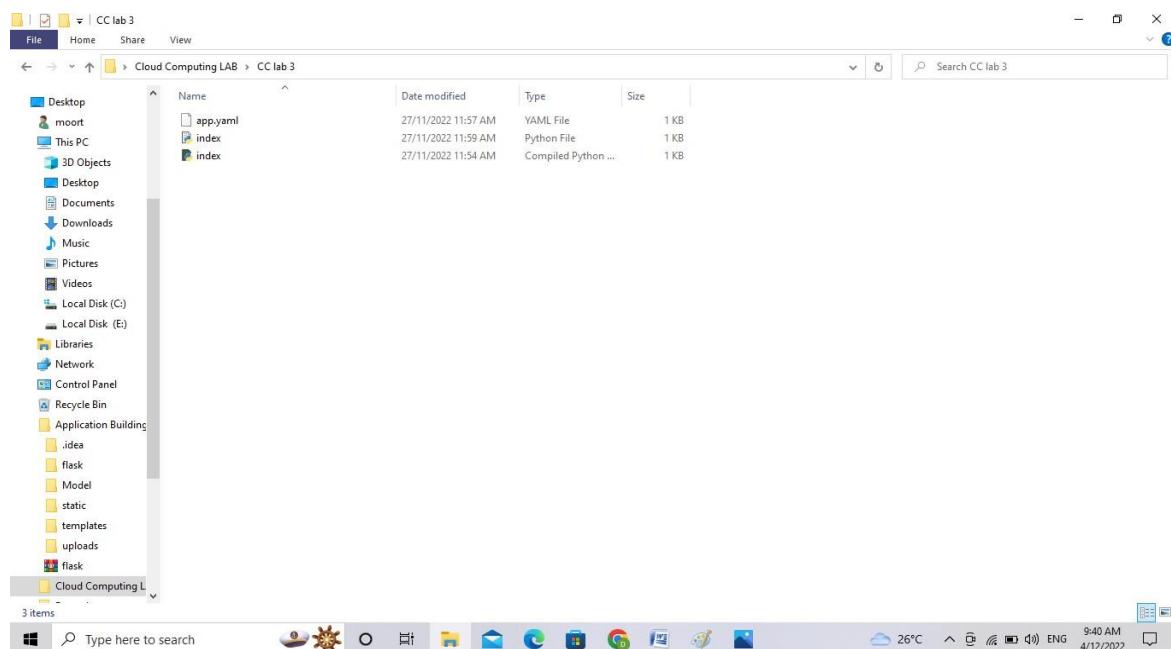
A screenshot of the Notepad++ application window. The title bar reads "C:\Users\moort\Desktop\Cloud Computing LAB\CC lab 3\index.py - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?.

The main editor area contains the following Python code:

```
1 print ('Welcome to JKEM College of Technology !!!');
2
```

The status bar at the bottom shows "Python file", "length: 54 lines: 2", "Ln: 2 Col: 1 Pos: 55", "Windows (CR LF)", "UTF-8", "INS", "26°C", "9:29 AM", and "ENG 4/12/2022".

3. New Folder >> Python & App.yaml codes



4. app.yaml :

```
runtime: python27  
api_version: 1  
threadsafe: false
```

```
handlers:
```

```
  - url: /.*
```

```
    script: index.py
```

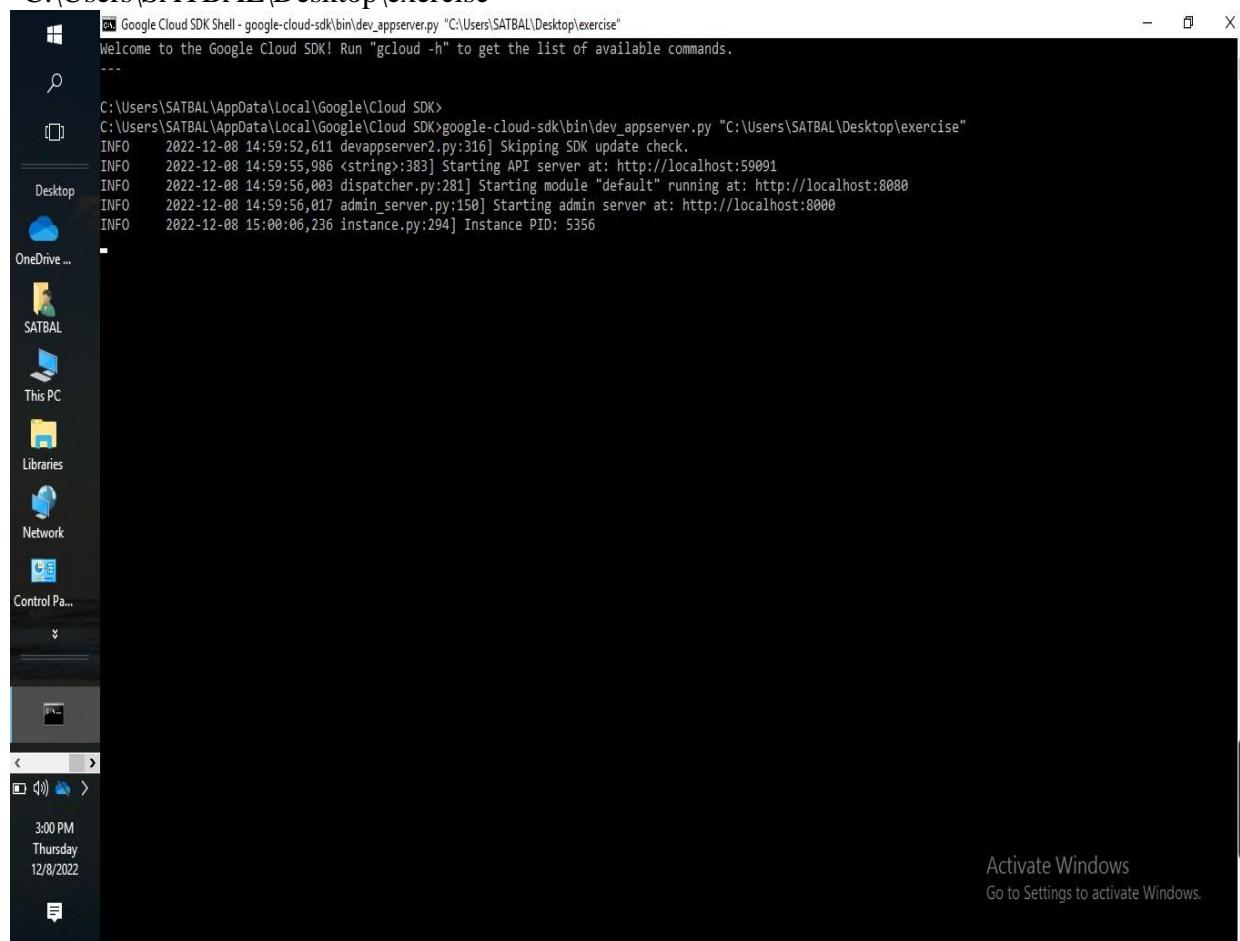
5. Open the Google cloud SDK or command prompt current Working folder:

6. Deploy The app.yaml code in Google cloud SDK Command line interface :

After opening the command line interface, type `google-cloud-sdk\bin\dev_appserver.py "location of your app.yaml"`

`C:\Users\SATBAL\AppData\Local\Google\Cloud SDK>`

`C:\Users\SATBAL\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\SATBAL\Desktop\exercise"`

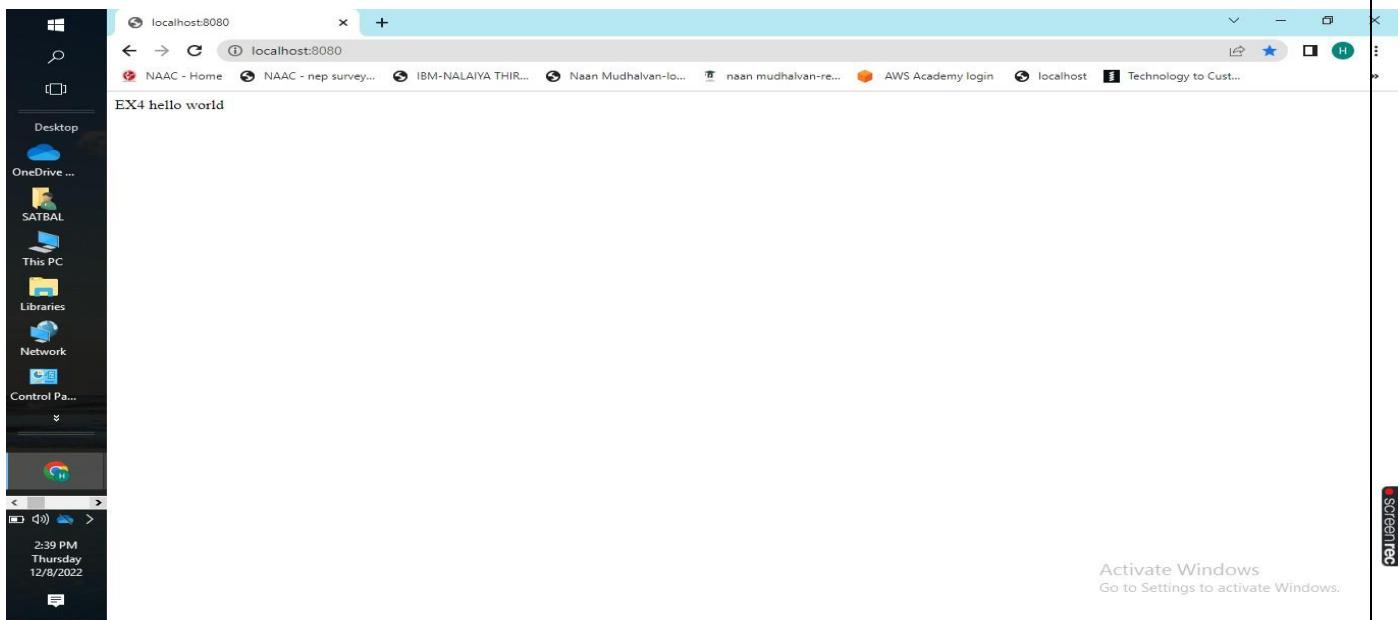


```
Google Cloud Shell - google-cloud-sdk\bin\dev_appserver.py "C:\Users\SATBAL\Desktop\exercise"  
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.  
---  
C:\Users\SATBAL\AppData\Local\Google\Cloud SDK>  
C:\Users\SATBAL\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\SATBAL\Desktop\exercise"  
INFO 2022-12-08 14:59:52,611 devappserver2.py:316] Skipping SDK update check.  
INFO 2022-12-08 14:59:55,986 <string>:383] Starting API server at: http://localhost:8080  
INFO 2022-12-08 14:59:56,003 dispatcher.py:281] Starting module "default" running at: http://localhost:8080  
INFO 2022-12-08 14:59:56,017 admin_server.py:158] Starting admin server at: http://localhost:8000  
INFO 2022-12-08 15:00:06,236 instance.py:294] Instance PID: 5356
```

The above command starts the API server and creates an instance with id .

7. Type the following url in web browser:

http://localhost:8080/



Result:

Thus the Google App Engine is installed and hello world application is created using python and executed

Ex No: 4	
DATE:	Use the GAE launcher to launch the web applications.

Aim:

To use GAE launcher to launch the web applications

Procedure:

You can use Google App Engine to host a static website. Static web pages can contain client-side technologies such as HTML, CSS, and JavaScript. Hosting your static site on App Engine can cost less than using a traditional hosting provider, as App Engine provides a free tier.

Sites hosted on App Engine are hosted on the REGION_ID.r.appspot.com sub domain, such as [my-project-id].uc.r.appspot.com. After you deploy your site, you can map your own domain name to your App Engine-hosted website.

Before you begin

1. Download google app engine 1.9.62 or use google cloud sdk

2. **Install and then initialize the Google app engine launchers:**

3. Download the SDK

Creating a website to host on Google App

Engine Basic structure for the project

This guide uses the following structure for the project:

- app.yaml: Configure the settings of your App Engine application.
- www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
- css/: Directory to store style sheets.

- style.css: Basic style sheet that formats the look and feel of your site.
- images/: Optional directory to store images.
- index.html: An HTML file that displays content for your website.
- js/: Optional directory to store JavaScript files.
- Other asset directories.

Creating the app.yaml file

The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files. In the following steps, you will add handlers that will load www/index.html when some one visits your website, and all static files will be stored in and called from the www directory.

Create the app.yaml file in your application's root directory:

1. Create a directory that has the same name as your project ID. You can find your project ID in the Console.
2. In directory that you just created, create a file named **app.yaml**.
3. Edit the **app.yaml** file and add the following code to the file:

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

handlers:
- url: /
  static_files: www/index.html
  upload: www/index.html

- url: /(.*) static_files: www/\1
  upload: www/(.*)
```

More reference information about the app.yaml file can be found in the app.yaml reference documentation.

Creating the index.html file

Create an **HTML file** that will be served when someone navigates to the root page of your website. Store this file in your `www` directory.

Deploying your application to App Engine

When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the `app.yaml` file is located:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}

button {
background-color: #04AA6D;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
}
button:hover {
opacity: 0.8;
}

.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}
```

```

/* Center the image and position the close button */
.imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
    position: relative;
}

img.avatar {
    width: 40%;
    border-radius: 50%;
}

.container {
    padding: 16px;
}

span.psw {
    float: right;
    padding-top: 16px;
}

/* The Modal (background) */
.modal {
    display: none; /* Hidden by default */
    position: fixed; /* Stay in place */
    z-index: 1; /* Sit on top */
    /* left: 0;
    top: 0;
    width: 100%; /* Full width */
    height: 100%; /* Full height */
    overflow: auto; /* Enable scroll if needed */
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
    padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
    background-color: #fefefe;
    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
    border: 1px solid #888;
    width: 80%; /* Could be more or less, depending on screen size */
}

```

```

/* The Close Button (x) */
.close {
    position: absolute;
    right: 25px;
    top: 0;
    color: #000;
    font-size: 35px;
    font-weight: bold;
}

.close:hover,
.close:focus {
    color: red;
    cursor: pointer;
}

/* Add Zoom Animation */
.animate {
    -webkit-animation: animatezoom 0.6s;
    animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
    from {-webkit-transform: scale(0)}
    to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
    from {transform: scale(0)}
    to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
    span.psw {
        display: block;
        float: none;
    }
    .cancelbtn {
        width: 100%;
    }
}
</style>
</head>
<body>
<h2>Login Form For CC</h2>

```

```

<button onclick="document.getElementById('id01').style.display='block'" style="width:auto;">Login</button>

<div id="id01" class="modal">

  <form class="modal-content animate" action="/image.html" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
      
    </div>

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="uname" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>

      <button type="submit">Login</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button" onclick="document.getElementById('id01').style.display='none'" class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
</div>

<script>
// Get the modal
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>

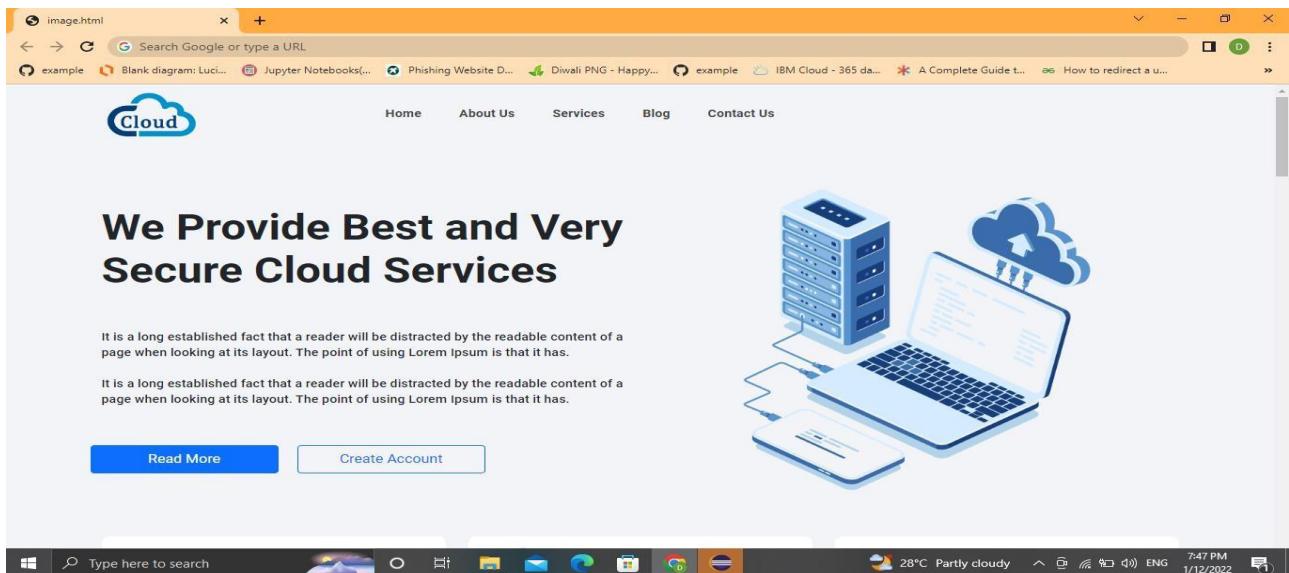
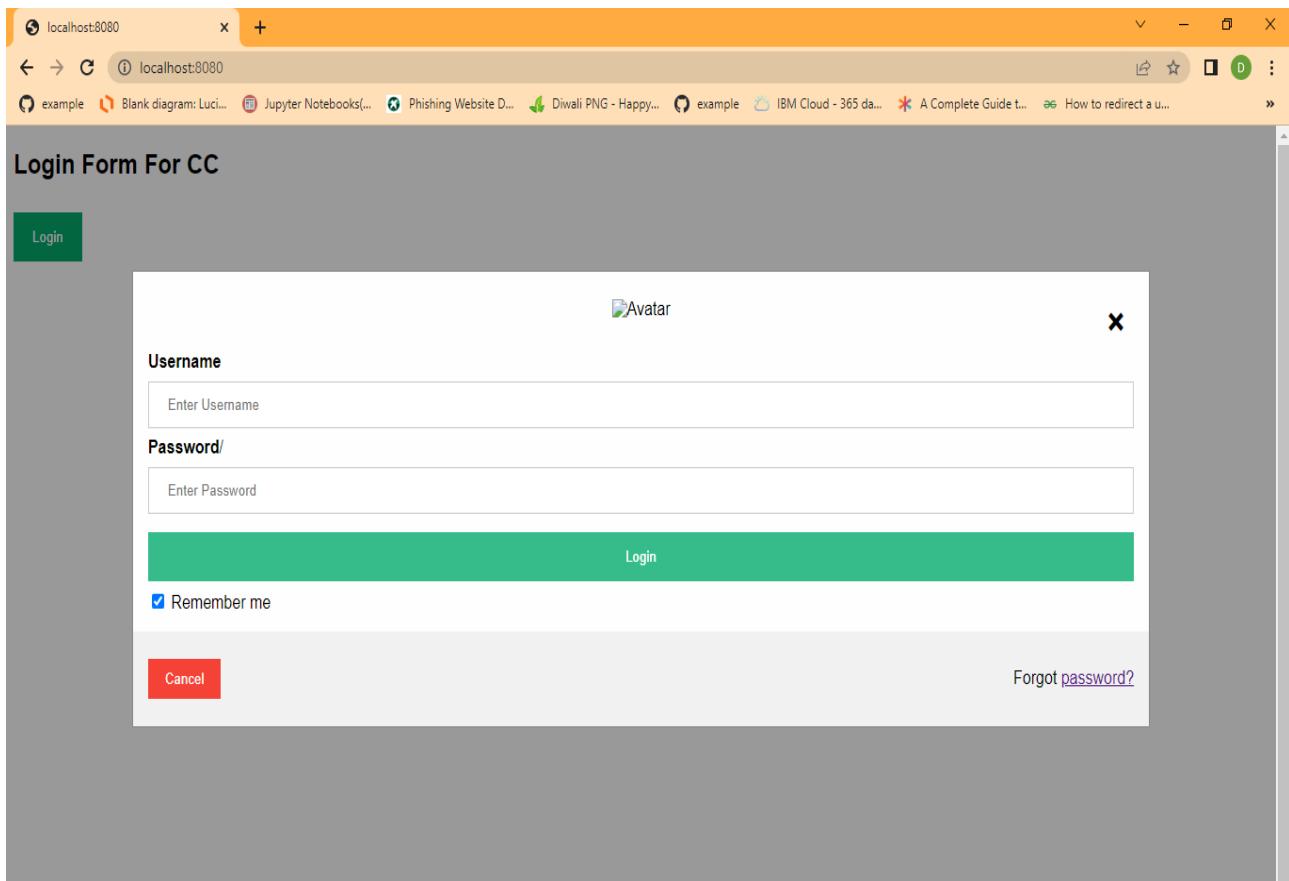
</body>
</html>

```

In google cloud sdk command line interface typ^{4e0} C:\Users\SATBAL\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\SATBAL\Desktop\exercise"

OUTPUT:





RESULT:

Thus the use GAE launcher is used to launch the python web application.

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

Aim:

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

Procedure:

Basics of Scheduling

In computers, Scheduling is a process of arranging the submitted jobs/task into a very specific sequence of execution. It is an essential characteristic of any software operating environment, which is handled by a very special program known as a scheduler.

Scheduler's main objective is to keep the underlined hardware resources(primarily processor) to be used effectively as well as efficient. In general, the scheduler may prefer to have any of the following scheduling approaches:

- **Space-shared:** In this, the requested resources are allocated dedicatedly to the requesting workload for execution and will be released only on completion. **Space-shared is also known as a batch process scheduling.**
- **Time-shared:** In this, the requested resources would be shared among more than one workload(task). The sharing is done based on time-sliced allocation where each workload is allocated with a required resource for a defined time(e.g., 200 milliseconds). Once the defined time slice is over, the current workload execution paused, and the resource is released. The released resource gets allocated to the next workload for the same defined time slice, and this cycle goes on till the time all the workloads execution is over. Time- shared is also known as round-robin **scheduling**.

Scheduling in Cloud

As cloud computing is the virtualized operating environment, and the virtual machines are the primary computing component which is responsible for the execution of the workloads(tasks). The virtual machine(s) are powered by a physical server host machine (i.e.) hardware. Depending on the requirement of the Virtual Machine(VM) there could be ‘one to one’ or ‘many to one’ mapping between the VM and host machine. That means in cloud computing the scheduling is done at both the mapping levels that are:

- Virtual Machine to Host Machines
- Tasks to Virtual Machines

Both of VM to Host as well as Workload(task) to VM mappings may utilize space-share or time-shared or any other specialized scheduling algorithm.

Scheduling in Cloudsim

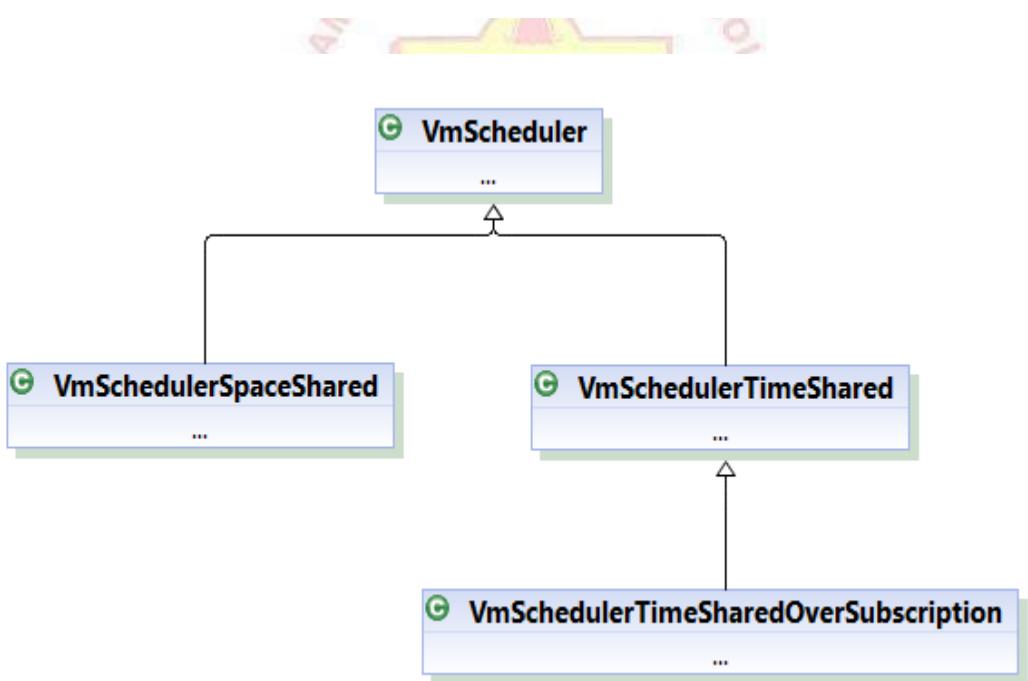
The Cloudsim simulation toolkit framework has effectively addressed the Scheduling scenario and implemented it as a set of the programmable class hierarchies with parent class as:

1. VmScheduler
2. CloudletScheduler

Also, Virtual Machine(VM) and Task(Cloudlet) scheduling are one of the most important and the popular use case to be simulated by researchers using the CloudSim simulation toolkit.

Note: In cloudsim, the task is called as cloudlet, therefore in the following text instead of ‘task’ we will be using the ‘cloudlet’.

Cloudsim Virtual Machine Scheduling



Cloudsim Virtual Machine Scheduler Class Hierarchy

These classes can be located in “*org.cloudbus.cloudsim*” package of cloudsim. The definition of this abstract class is extended to the following types of policies implemented as classes:

- **Vm Scheduler Time Shared:** This class implements the VM scheduling policy that allocates one or more processing elements to a single Virtual machine and allows the sharing of processing elements by multiple virtual machines with a specified time slice. This class also considers the overhead of VM allocation switching(similar to context switching) in policy definition. Here, the VM allocation will fail if the number of processing elements requested is not available. for example, if the VM request for quad- core processor whereas the allocated host has an only dual-core the allocation will fail.
- **Vm Scheduler Space Shared:** This class implements the VM scheduling policy that allocates one or more processing elements to a single virtual machine, but this policy implementation does not support sharing of processing elements (i.e.) all the requested resources will be used by the allocated VM till the time the VM is not destroyed. Also, Under this allocation policy, if any virtual machine requests a processing element and is not available at that time, the allocation fails.
- **Vm Scheduler Time Shared Over Subscription:** This is an extended implementation of VMScheduler Time Shared VM scheduling policy, which allows over-subscription of processing elements by the virtual machine(s) (i.e.) the scheduler still allows the allocation of VMs that require more CPU capacity than is available. And this oversubscription results in performance degradation.

The application of the VmScheduler classes is while instantiating the host model. Following is the code snippet used in CloudsimExample1.java from line number 160 to 174:

```

int hostId = 0;
int ram = 2048; // host memory
(MB) long storage = 1000000; //
host storageint bw = 10000;
hostList.add
(new Host(hostId,
new RamProvisionerSimple(ram),new
BwProvisionerSimple(bw), storage,
peList,
new VmSchedulerTimeShared(peList)
)
); // This is our machine

```

This is where the processing element list is passed as a parameter to the VmSchedulerTimeShared() class call and during the simulation, the cloudsim will simulate the timeshare behavior for the virtual machines. Also, in case you want to test other VmScheduler you may replace it with VmScheduler Time Shared() call with appropriate parameters, this includes your own designed custom virtual machine scheduler.

Cloudsim Cloudlet Scheduling

The “***CloudletScheduler***” is an abstract class that defines the basic skeleton to implement the policy to be used for cloudlet scheduling to be performed by a virtual machine. The hierarchy of the cloudsim Cloudlet scheduler classes is as:

Cloudlet Scheduler Class Hierarchy

These classes again exist in “*org.cloudbus.cloudsim*” package of cloudsim. The definition of this abstract class is extended as the following types of policies implemented as three individual classes in cloudsim:

- **Cloudlet Scheduler Space Shared:** This class implements a policy of scheduling for Virtual machine to execute cloudlet(s) in space shared environment (i.e.) only one cloudlet will be executed on a virtual machine at a time. It means cloudlets share the same queue and requests are processed one at a time per computing core. *Space-sharing* is similar to batch processing.
- **Cloudlet Scheduler Time Shared:** This class implements a policy of cloudlet scheduling for Virtual machines to execute cloudlets in a time-shared environment (i.e.) more than one cloudlet will be submitted to the virtual machine and each will get its specified share of time. It means several requests (cloudlets) are processed at once but they must share the computing power of that virtual machine(by simulating context switching), so they will affect each other's processing time. It basically influences the completion time of a cloudlet in CloudSim. Time-sharing is probably referring to the concept of sharing executing power (such as CPU, logical processor, GPU) and is commonly known as the round-robin scheduling.
- **Cloudlet Scheduler Dynamic Workload:** This implements a special policy of scheduling for virtual machine assuming that there is just one cloudlet which is working as an online service with a different requirement of workload as per the need of peak/offpeak user load at a specified period of time.

The application of the CloudletScheduler classes is while instantiating the Vm model. Following is the code snippet used in CloudsimExample1.java from line number 82 to 91:

```

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 cpusString
vmm = "Xen"; // VMM name

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

```

By instantiating the CloudletSchedulerTimeShared() class, the Virtual machine is decided to follow the timeshare(round-robin) approach while simulation for scheduling & executing theCloudlets. Also, in case you want to test other CloudletScheduler you may replace it with CloudletScheduler TimeS hared() call with appropriate parameters, this includes your own designed custom cloudlet scheduler.

Now in case you want to implement your own scheduling policies with respect to Virtual Machine or Cloudlet(s), you may simply extend the VmScheduler or CloudletScheduler class to implement all the abstract methods as specified. This gives you the flexibility to design and implement your own set of algorithms and then later test & optimize during the repetitive simulation runs.

Learn More

In case you missed reading our popular article on Detailed introduction to Cloudsim you may follow CloudSim Simulation Toolkit: An Introduction.

You may subscribe to an online self-paced course named Learn Basics of Cloudsim, The contentfor this course which will be updated weekly till August 2021.

CLOUD SCHEDULING ALGORITHM

```

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
     */
    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 machinevmlist = 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);

// Print the debt of each user to each datacenter
datacenter0.printDebts();

```

```

Log.printLine("CloudSimExample1 finished!");
} catch (Exception e) {
e.printStackTrace(); Log.printLine("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 machinesint hostId = 0;
int ram = 2048; // host memory (MB) long storage = 1000000; // host storageint 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 systemString 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 resourcedouble 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 SA // devices by

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.println();
Log.println("===== OUTPUT =====");
Log.println("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);

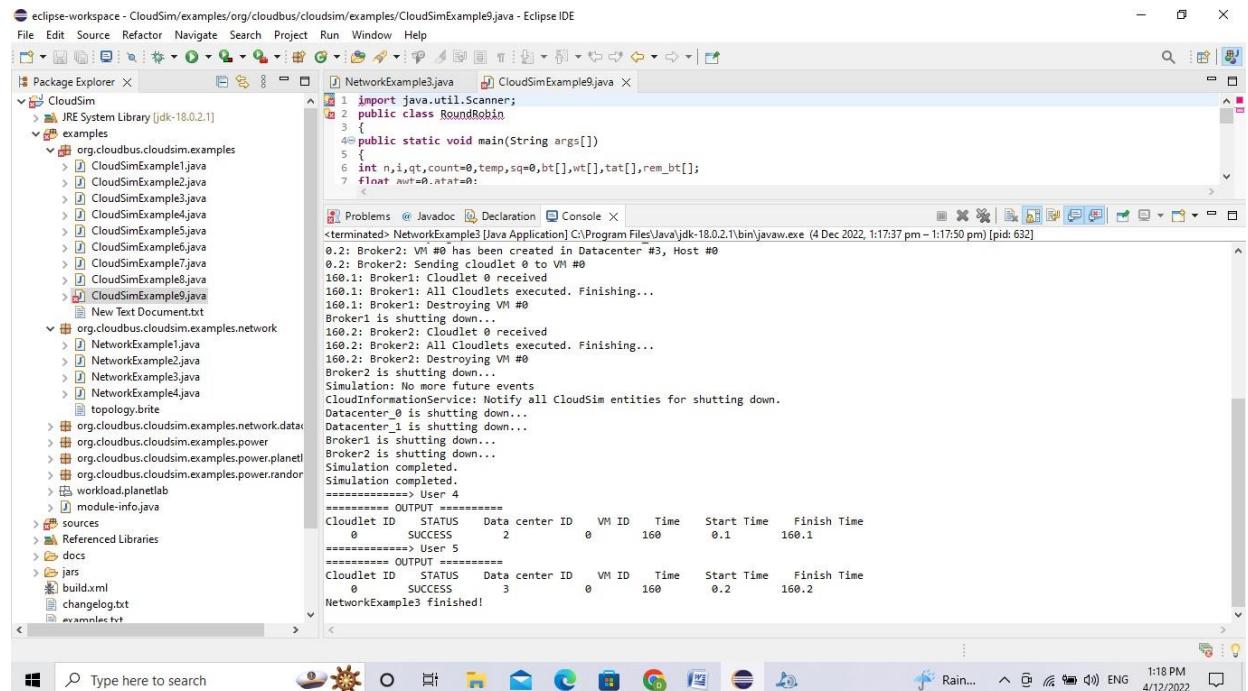
if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {Log.print("SUCCESS");

Log.println(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 :



The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and run.
- Package Explorer:** Shows the project structure under the CloudSim package, including JRE System Library [jdk-18.0.2.1] and examples sub-package containing NetworkExample1.java through NetworkExample9.java, plus New Text Document.txt.
- Editor:** Displays the CloudSimExample9.java code, which contains a main method that creates a RoundRobin scheduler and executes a series of cloudlets across two brokers.
- Console:** Shows the terminal output of the application execution. It includes log messages from brokers (Broker1 and Broker2) indicating the creation of VMs, sending cloudlets, receiving them, and destroying VMs. It also shows the simulation shutting down and the completion of the execution.
- Bottom Status Bar:** Shows the system tray, network status, battery level, and system information including the date and time (4/12/2022, 1:18 PM).

Key log entries from the console output:

```

<terminated> NetworkExample3 Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (4 Dec 2022, 1:17:37 pm - 1:17:50 pm) [pid: 632]
0.2: Broker2: VM #0 has been created in Datacenter #3, Host #0
0.2: Broker2: Sending cloudlet 0 to VM #0
160.1: Broker1: Cloudlet 0 received
160.1: Broker1: All Cloudlets executed. Finishing...
160.1: Broker1: Destroying VM #0
Broker1 is shutting down...
160.2: Broker2: Cloudlet 0 received
160.2: Broker2: All Cloudlets executed. Finishing...
160.2: Broker2: Destroying VM #0
Broker2 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker1 is shutting down...
Broker2 is shutting down...
Simulation completed.
Simulation completed.
=====
=> User 4
=====
===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
0      SUCCESS    2            0       160     0.1        160.1
=====
=> User 5
=====
===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
0      SUCCESS    3            0       160     0.2        160.2
NetworkExample3 finished!

```

Result:

Thus the cloud scenario using CloudSim and scheduling algorithm in CloudSim is executed.

Ex No: 6

DATE:

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

Aim:

To move the files between virtual machine.

You can move files between virtual machines in several ways:

- You can copy files using network utilities as you would between physical computers on your network. To do this between two virtual machine:
 - Both virtual machines must be configured to allow access to your network. Any of the networking methods (host-only, bridged and NAT) are appropriate.
 - With host-only networking, you copy files from the virtual machines to the host and vice-versa, since host-only networking only allows the virtual machines see your host computer.
 - With bridged networking or NAT enabled, you can copy files across your network between the virtual machines.
- You can create a shared drive, either a virtual disk or a raw partition, and mount the drive in each of the virtual machines.

How to Enable File sharing in Virtual Box.

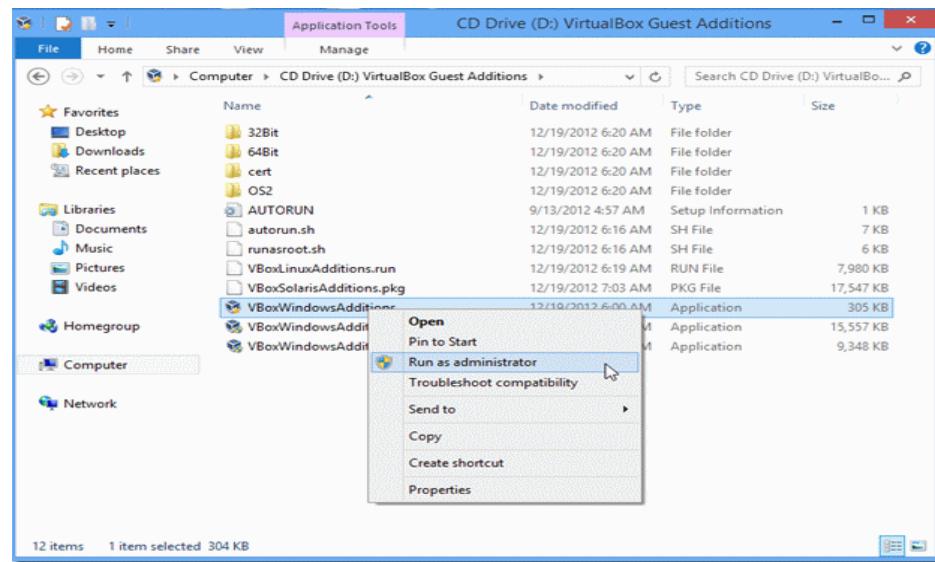
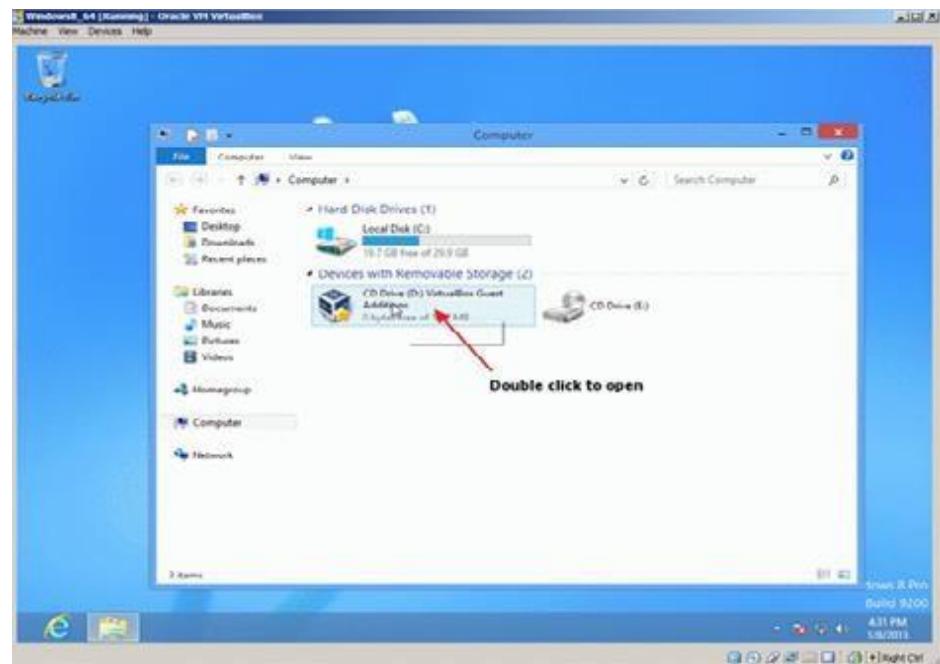
Step 1. Install Guest Additions on the Guest machine.
Step 2. Configure File Sharing on VirtualBox.

Step 1. Install Guest Additions on the Guest machine.

1. Start the Virtuabox Guest Machine (OS).
2. From Oracle's VM VirtualBox main menu, select **Devices > Install Guest Additions ***



- a. . Open Windows Explorer.
- b. Double click at the "CD Drive (X:) VirtualBox Guest additions" to explore its contents.



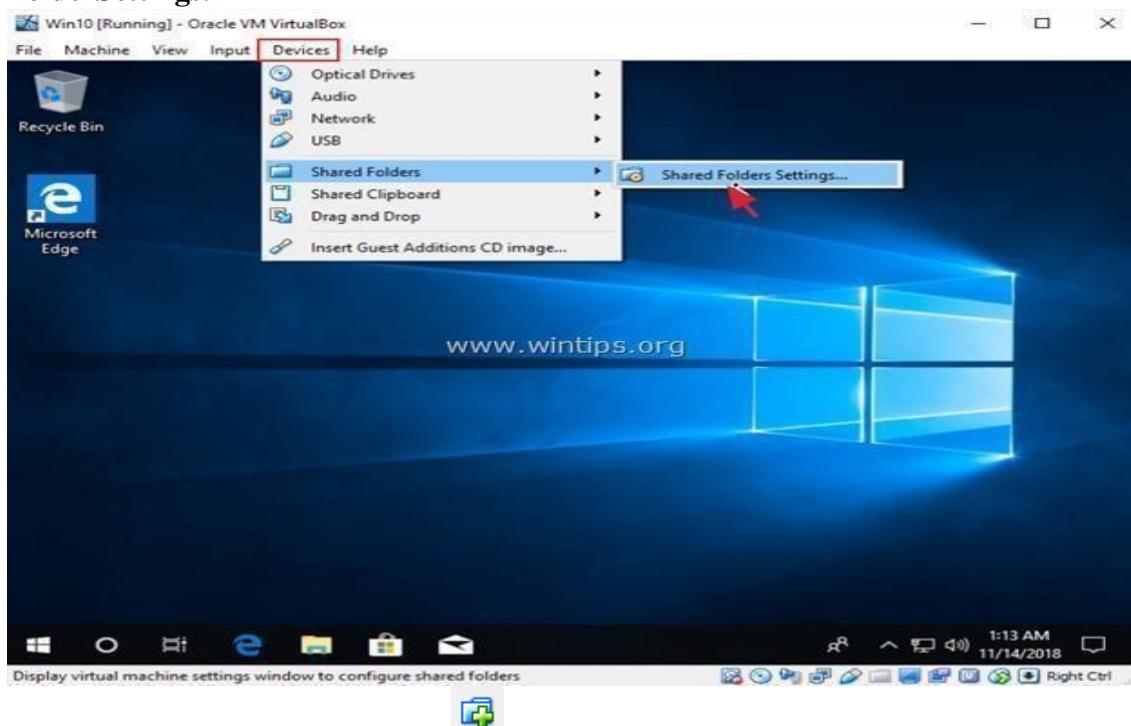
3. Press **Next** and then follow the on screen instructions to complete the Guest Additions installation.



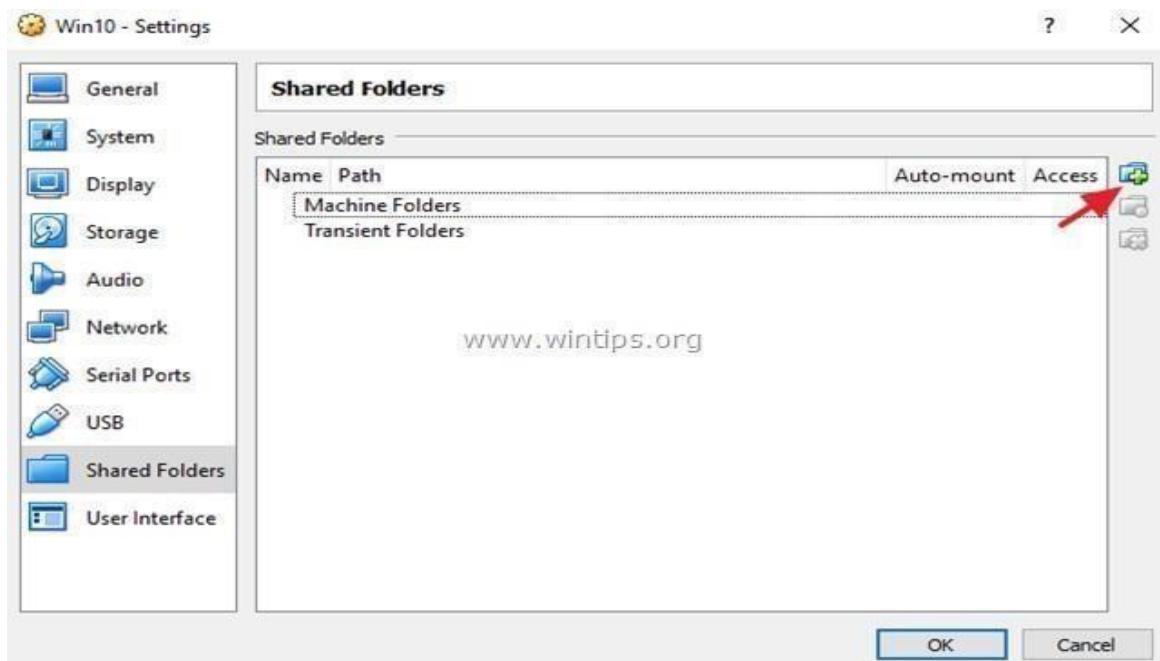
4. When the setup is completed, choose **Finish** and **restart** the Virtuabox guest machine.

Step 2. Setup File Sharing on VirtualBox Guest Machine.

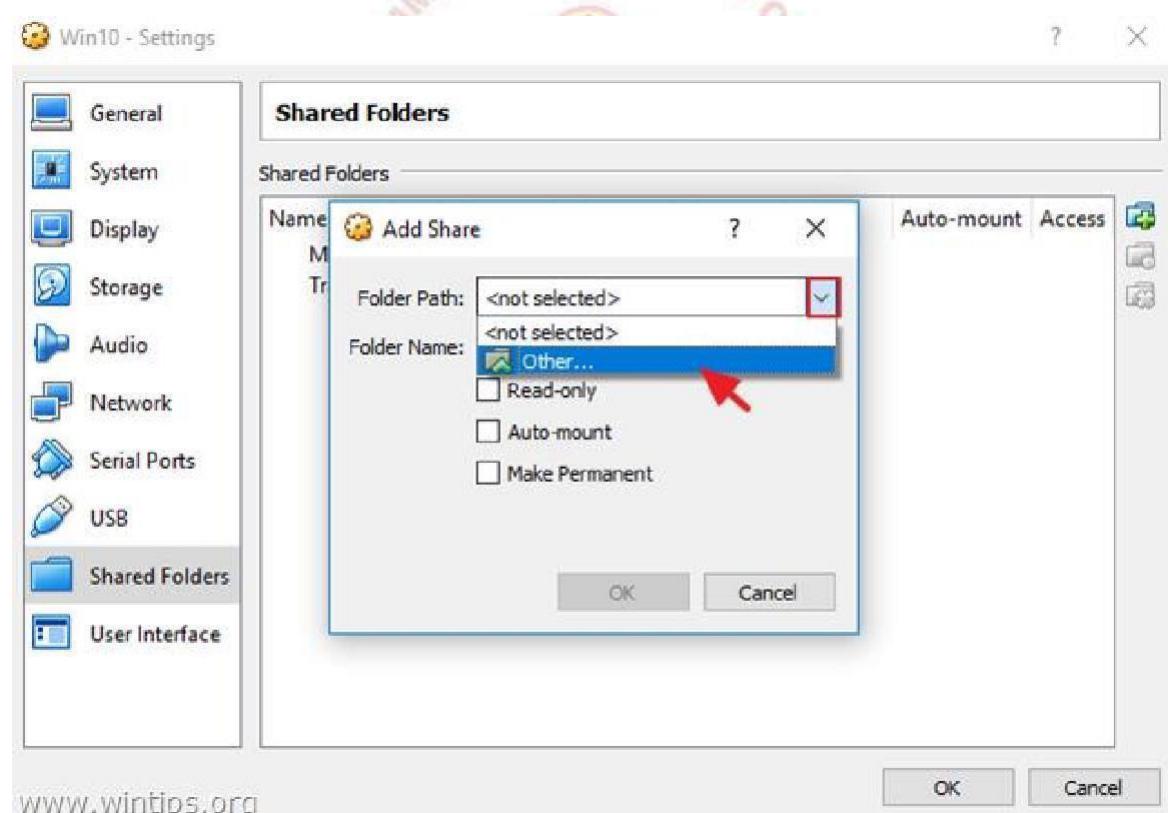
1. From VirtualBox menu click **Devices** and choose **Shared Folders -> Shared Folder Settings**.



1. Click the Add new shared folder icon.

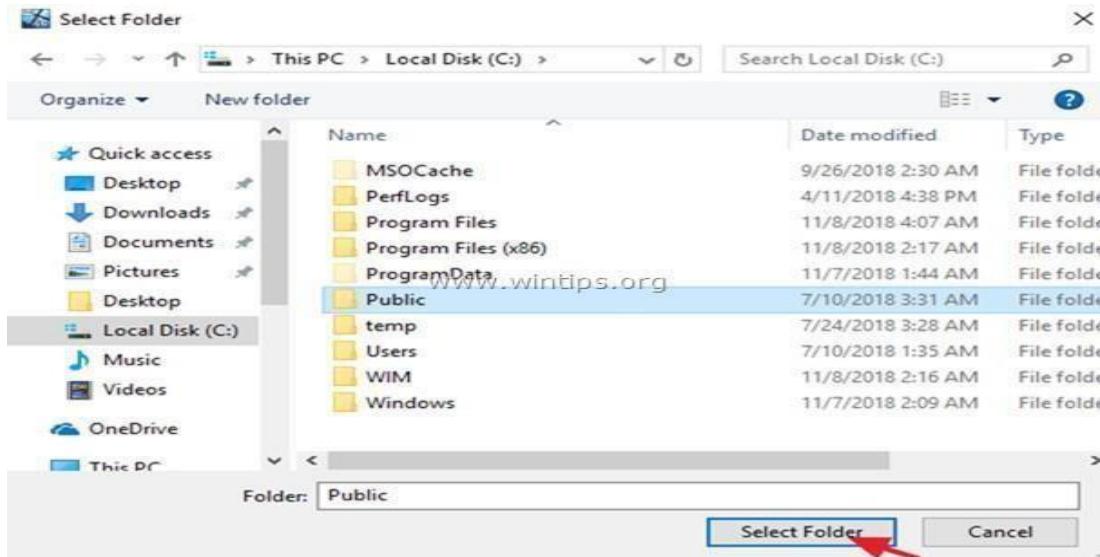


2. Click the drop-down arrow and select Other.

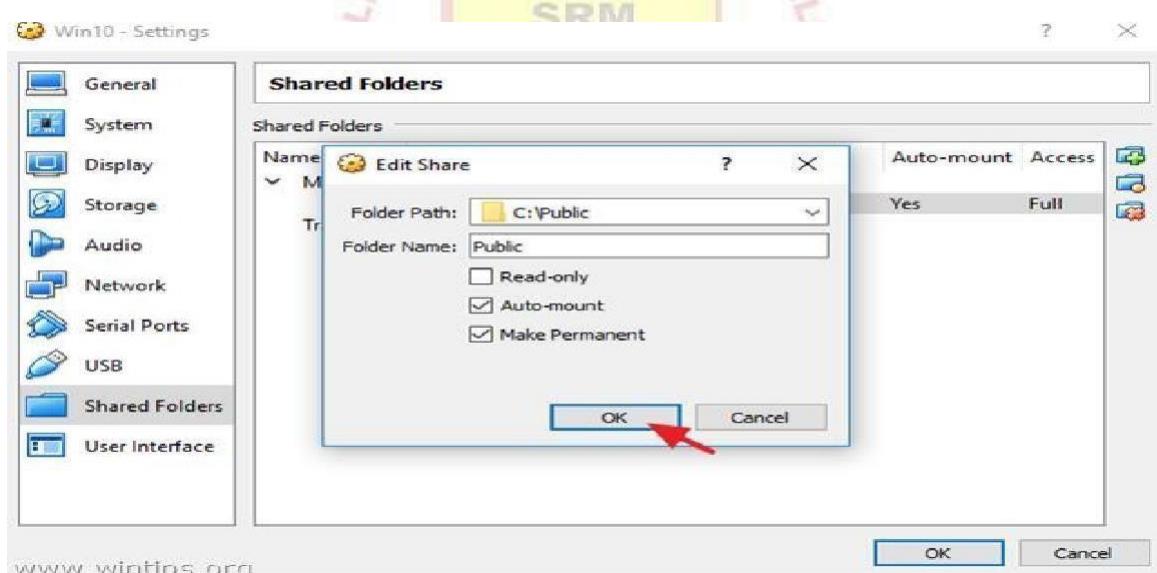


3. Locate and highlight (from the Host OS) the folder that you want to share between the Virtual Box Guest machine and the Host and click Select Folder. *

* Note: To make your life easier, create a new folder for the file sharing, on the Host OS and give it with a recognizable name. (e.g. "Public")



- Now, in the 'Add Share' options, type a name (if you want) at the 'Folder Name' box, click the **Auto Mount** and the **Make Permanent** checkboxes and click **OK** twice to close the Shared Folder Settings.



- You're done! To access the shared folder from the Guest OS, open Windows Explorer and under the 'Network locations' you should see a new network drive that corresponds to the shared folder on the Host OS.

Result:

Thus the files are transferred from one VM to another VM.

Ex No:07	
DATE:	Install Hadoop single node cluster and run simple applications like word count.

Aim:

To Set up the one node Hadoop Cluster and execute a word count program

Procedure:**1) Installing Java:**

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware. Hadoop needs Java 6 or above to work.

Step 1: Download Jdk tar.gz file for linux-62 bit, extract it into “/usr/local”
boss@solaiv[]# cd /opt

boss@solaiv[]# sudo tar xvzf /home/itadmin/Downloads/jdk-8u5-linux-x64.tar.gz
boss@solaiv[]# cd /opt/jdk1.8.0_05

Step 2:

Open the “/etc/profile” file and Add the following line as per the version set a environment for Java

Use the root user to save the /etc/proflie or use gedit instead of vi .

The 'profile' file contains commands that ought to be run for login shells

boss@solaiv[]# sudovi /etc/profile#--insert JAVA_HOME

JAVA_HOME=/opt/jdk1.8.0_05

#--in PATH variable just append at the end of the line PATH=\$PATH:\$JAVA_HOME/bin

#--Append JAVA_HOME at end of the export statement export PATH JAVA_HOME
save the file using by pressing “Esc” key followed by :wq!

Step 3: Source the /etc/profile
boss@solaiv[]# source

/etc/profile
Step 4: Update the java alternatives

By default OS will have a open jdk. Check by “java -version”. You will be prompt “openJDK”

If you also have openjdk installed then you'll need to update the java alternatives:

If your system has more than one version of Java, configure which one your system causes by entering the following command in a terminal window

By default OS will have a open jdk. Check by “java -version”. You will be prompt “Java HotSpot(TM) 64-Bit Server”

```
boss@solaiv[]# update-alternatives --install "/usr/bin/java" java  
"/opt/jdk1.8.0_05/bin/java" 1
```

```
boss@solaiv[]# update-alternatives --config java --type selection number:  
boss@solaiv[]# java -version
```

1) configure ssh

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this short tutorial). For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost

The need to create a Password-less SSH Key generation based authentication is so

that the master node can then login to slave nodes (and the secondary node) to start/stop them easily without any delays for authentication

If you skip this step, then have to provide password

Generate an SSH key for the user. Then Enable password-less SSH access to yo sudo apt-get install openssh-server

--You will be asked to enter password, root@solaiv[]# sshlocalhost root@solaiv[]# ssh-

```
keygenroot@solaiv[]# ssh-copy-id -i localhost  
--After above 2 steps, You will be connected without password, root@solaiv[]#  
sshlocalhost
```

```
root@solaiv[]# exit
```

2) Hadoop installation

Now Download Hadoop from the official Apache, preferably a stable release version of Hadoop 2.7.x and extract the contents of the Hadoop package to a location of your choice.

We chose location as “/opt/”

- Step 1: Download the tar.gz file of latest version Hadoop(hadoop-2.7.x) from the officialsite .
- Step 2: Extract(untar) the downloaded file from this commands to /opt/bigdata

```
root@solaiv[]# cd /opt  
root@solaiv[/opt]# sudo tar xvzf /home/itadmin/Downloads/hadoop-  
2.7.0.tar.gzroot@solaiv[/opt]# cd hadoop-2.7.0/
```

Like java, update Hadop environment variable in /etc/profile

```
boss@solaiv[]# sudovi /etc/profile
```

```
#--insert HADOOP_PREFIX HADOOP_PREFIX=/opt/hadoop-2.7.0  
  
##in PATH variable just append at the end of the  
linePATH=$PATH:$HADOOP_PREFIX/bin  
  
##Append HADOOP_PREFIX at end of the export statement export  
PATHJAVA_HOME HADOOP_PREFIX
```

save the file using by pressing “Esc” key followed by :wq!

Step 3: Source the /etc/profile
boss@solaiv[]# source /etc/profile Verify Hadoop installation

```
boss@solaiv[]# cd $HADOOP_PREFIX boss@solaiv[]# bin/hadoop version Modify the Hadoop Configuration Files
```

```
Add the following properties in the various hadoop configuration files which is available under $HADOOP_PREFIX/etc/hadoop/  
core-site.xml, hdfs-site.xml, mapred-site.xml & yarn-site.xml
```

Update Java, hadoop path to the Hadoop environment file

```
boss@solaiv[]# cd $HADOOP_PREFIX/etc/hadoop
```

```
boss@solaiv[]# vi hadoop-env.sh
```

Paste following line at beginning of the file export

```
JAVA_HOME=/usr/local/jdk1.8.0_05 export
```

```
HADOOP_PREFIX=/opt/hadoop-2.7.0Modify the core-site.xml
```

```
boss@solaiv[]# cd
```

```
$HADOOP_PREFIX/etc/hadoopboss@solaiv[]# vi core-site.xmlPaste following between <configuration> tags
```

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

Modify the hdfs-site.xml boss@solaiv[]# vi hdfs-site.xml

Paste following between <configuration> tags

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

YARN configuration - Single Node modify the mapred-site.xml boss@solaiv[]# cp mapred-site.xml.template

```
mapred-site.xml boss@solaiv[]# vi mapred-site.xml
```

Paste following between <configuration> tags

```
<configuration>  
<property>
```

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

Modiy yarn-site.xml boss@solaiv[]# vi yarn-site.xml

Paste following between <configuration> tags

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

Formatting the HDFS file-system via the NameNode

The first step to starting up your Hadoop installation is formatting the Hadoop files system which is implemented on top of the local file system of our “cluster” which includes only our local machine. We need to do this the first time you set up a Hadoop cluster.

Do not format a running Hadoop file system as you will lose all the data currently in the cluster (in HDFS)

```
root@solaiv[]# cd $HADOOP_PREFIX root@solaiv[]# bin/hadoopnamenode -format Start
```

NameNode daemon and DataNode daemon: (port 50070)

```
root@solaiv[]# sbin/start-dfs.sh
```

To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps Start ResourceManager daemon and NodeManager daemon: (port 8088) root@solaiv[]# sbin/start-yarn.sh

To stop the running process root@solaiv[]# sbin/stop-dfs.sh

To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps Start ResourceManager daemon and NodeManager daemon: (port 8088) root@solaiv[]# sbin/stop-yarn.sh

PROCEDURE:

Update APT

```
test@cs-88:~$ sudo apt-get update E: Unable to lock directory /var/lib/apt/lists/ Check Backgroud  
Process :
```

```
test@cs-88:~$ ps -ef |grep apt
```

```
root  2292          2285 0 11:11 ?    00:00:00 /bin/sh /etc/cron.daily/apt
root  2612          2292 0 11:30 ?    00:00:00 apt-get -qq -y update
root  2615          2612 0 11:30 ?    00:00:00 /usr/lib/apt/methods/http
root  2616          2612 0 11:30 ?    00:00:00 /usr/lib/apt/methods/http
root  2617          2612 0 11:30 ?    00:00:00 /usr/lib/apt/methods/http
root  2619          2612 0 11:30 ?    00:00:00 /usr/lib/apt/methods/gpgv
root  2627          2612 0 11:30 ?    00:00:01 /usr/lib/apt/methods/bzip2

test  2829 2813 0 11:36 pts/0  00:00:00 grep --color=auto apt
```

Kill Backgroud Process :

```
test@cs-88:~$ sudo kill -9 2292 2612 2615 2616 2617 2619 2627 2829
```

Updaet apt :

```
test@cs-88:~$ sudo apt-get update
```

git installation :

```
root@cs-88:~# sudo apt-get install git
```

Clone :

```
root@cs-88:~# git clone https://git.openstack.org/openstack-dev/devstack
```

```
root@cs-
88:~# ls
devstack
root@cs-88:~# cd devstack
root@cs-88:~/devstack# nano
```

```
local.conf[[local|localre]]
```

```
HOST_IP=192.168.4.88
//FLOATING_RANGE=192.168.1.224/27
FIXED_RANGE=10.11.12.0/24
FIXED_NETWORK_SIZE=256
FLAT_RANGE=eth0
ADMIN_PASSWORD=linux
DATABASE_PASSWORD=linux
RABBIT_PASSWORD=linux
SERVICE-TOKEN=linux
```

Save Nono file:

control+x

Hadoop Installation;

```
stack@cs-88:~/Downloads$ sudo scp -r *
/opt/stack@cs-88:~/Downloads$ ls /opt/
test@cs-88:/opt$ ls
hadoop-2.7.0.tar.gz Hadoop Pseudo-Node.pdf HDFSCCommands.pdf jdk-8u60-
linux-x64

test@cs-
88:/opt$ lstatotal
383412
drwxr-xr-x 2 root root 4096 May 6 15:41 .
drwxr-xr-x 23 root root 4096 May 6 16:34 ../
-rw-r--r-- 1 root root 210343364 May 6 15:41 hadoop-2.7.0.tar.gz
-rw-r--r-- 1 root root 159315 May 6 15:41 Hadoop Pseudo-Node.pdf
-rw-r--r-- 1 root root 43496 May 6 15:41 HDFSCCommands.pdf
-rw-r--r-- 1 root root 181238643 May 6 15:41 jdk-8u60-linux-x64.gz
-rw-r--r-- 1 root root 402723 May 6 15:41 mrsampledatal(1).tar.gz
-rw-r--r-- 1 root root 402723 May 6 15:41
```

mrsampledatal(1).tar.gz Change root user to test user:

```
test@cs-88:/opt$ sudo chown -Rh test:test /opt/
test@cs-88:/opt$ ll /* display list file with
permissiontotal 383412
drwxr-xr-x 2 test test 4096 May 6 15:41 .
drwxr-xr-x 23 root root 4096 May 6 16:34 ../
-rw-r--r-- 1 test test 210343364 May 6 15:41 hadoop-2.7.0.tar.gz
-rw-r--r-- 1 test test 159315 May 6 15:41 Hadoop Pseudo-Node.pdf
-rw-r--r-- 1 test test 43496 May 6 15:41 HDFSCCommands.pdf
-rw-r--r-- 1 test test 181238643 May 6 15:41 jdk-8u60-linux-x64.gz
-rw-r--r-- 1 test test 402723 May 6 15:41 mrsampledatal(1).tar.gz
-rw-r--r-- 1 test test 402723 May 6 15:41
mrsampledatal(1).tar.gztest@cs-88:/opt$
```

Unzip JAVA

```
test@cs-88:/opt$ tar -zvxf jdk-8u60-linux-x64.gz
test@cs-88:/opt$ cd
jdk1.8.0_60 test@cs-
88:/opt/jdk1.8.0_60$ pwd
```

```
/opt/jdk1.8.0
_60test@cs-
88:/
Set profiloe for JAVA
test@cs-88:/opt/jdk1.8.0_60$ sudo nano /etc/profile

JAVA_HOME=/opt/jdk1.8.0_60
HADOOP_PREFIX=/opt/hadoop-
2.7.0

PATH=$PATH:$JAVA_HOME/bin
PATH=$PATH:$HADOOP_PREFIX
/bin

export PATH JAVA_HOME HADOOP_PREFIX
```

Save:

Control +x

Press y

Press Enterkey

```
test@cs-88:/opt/jdk1.8.0_60$
cd ..test@cs-88:/opt$ pwd
/opt test@cs-88:/opt$ Unzip hadoop file
test@cs-88:/opt$tar -zxvf hadoop-
```

```
2.7.0.tar.gztest@cs-88:/opt$ source
```

/etc/profile

Java Version

```
test@cs-88:/opt$ java -version
```

```
test@cs-88:$ source
/etc/profiletest@cs-88:$
java -version
java version "1.8.0_60"
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)
```

```
SSH keygeneration
test@cs-88:/opt$ ssh-keygen
Generating public/private rsa key
pair.
Enter file in which to save the key (/home/test/.ssh/id_rsa):
Created directory '/home/test/.ssh'.
Enter passphrase (empty for no
passphrase):Enter same passphrase
again:
Your identification has been saved in /home/test/.ssh/id_rsa.
```

```
Your public key has been saved in
/home/test/.ssh/id_rsa.pub. The key fingerprint is:
c6:f4:33:42:4d:87:fb:3a:72:29:e9:5b:ce:ee:e9:e4 test@cs-
88 The key's randomart image is:
+--[ RSA 2048] --- +
|      ...
|      o..
|      o ..
|      + ..
|      S +.
|      ..o.
|      .oo
|      +*=.
|      .oOE.
+-----+test@cs-88:/opt$
```

```
configure ssh
test@cs-88:/opt$ ssh-copy-id -i localhost
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
anythat are already installed
```

```
test@cs-88:/opt$ sudo apt-get install openssh-
```

```
server
test@cs-88:/opt$ ssh-copy-id -i localhost
```

```
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is
67:12:a1:69:99:ea:b7:b7:96:b1:f5:4a:29:b5:d0:29. Are you sure you want
to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
```

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any  
that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted  
now it is to install the new keys  
test@localhost's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.

```
test@cs-88:/opt$  
Configure Hadoop  
Verify Hadoop installation
```

```
test@cs-88:/opt$ cd  
$HADOOP_PREFIXtest@cs-  
88:/opt/hadoop-2.7.0$ pwd  
/opt/hadoop-2.7.0  
test@cs-88:/opt/hadoop-2.7.0$
```

```
test@cs-88:/opt/hadoop-2.7.0$ ls  
bin include libexec NOTICE.txt  
sbin etc lib LICENSE.txt  
README.txt share
```

```
test@cs-88:/opt/hadoop-2.7.0$ bin/hadoop  
version  
Hadoop 2.7.0  
Subversion https://git-wip-  
us.apache.org/repos/asf/hadoop.git -r  
d4c8d4d4d203c934e8074b31289a28724c0842cf  
Compiled by jenkins on 2015-04-  
10T18:40ZCompiled with protoc 2.5.0  
From source with checksum a9e90912c37a35c3195d23951fd18f  
This command was run using /opt/hadoop-2.7.0/share/hadoop/common/hadoop-  
common-2.7.0.jar  
test@cs-88:/opt/hadoop-2.7.0$
```

Update Java, hadoop path to the Hadoop environment file
test@cs-88:/opt/hadoop-2.7.0\$ cd
\$HADOOP_PREFIX/etc/hadooptest@cs-88:/opt/hadoop-

```
2.7.0/etc/hadoop$ pwd  
/opt/hadoop-2.7.0/etc/hadoop  
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$  
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano hadoop-  
env.shtype at last  
export JAVA_HOME=/opt/jdk1.8.0_60 export HADOOP_PREFIX=/opt/hadoop-2.7.0
```

```
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano core-site.xml
```

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

```
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano hdfs-site.xml
```

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

```
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ cp mapred-site.xml.template mapred-
```

```
site.xmltest@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano mapred-site.xml
```

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

yet another resource negotiated : yarn

```
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano yarn-site.xml
```

```
<configuration>
```

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

```
test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ cd $HADOOP_PREFIX
test@cs-88:/opt/hadoop-2.7.0$ bin/hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is
deprecated.Instead use the hdfs command for it.
```

```
16/05/07 09:24:13 INFO namenode.NameNode: STARTUP_MSG:
```

```
*****
```

```
***images with txid >= 0
```

```
16/05/07 09:24:14 INFO util.ExitUtil: Exiting with status 0
```

```
16/05/07 09:24:14 INFO namenode.NameNode: SHUTDOWN_MSG:
```

```
*****
```

```
***SHUTDOWN_MSG: Shutting down NameNode at cs-88/127.0.1.1
```

```
*****
```

```
***/test@cs-88:/opt/hadoop-2.7.0$
```

```
test@cs-88:/opt/hadoop-2.7.0$ sbin/start-
```

```
dfs.shStarting namenodes on [localhost]
```

```
localhost: starting namenode, logging to /opt/hadoop-2.7.0/logs/hadoop-test-
namenode-cs-88.out
```

```
localhost: starting datanode, logging to /opt/hadoop-2.7.0/logs/hadoop-test-
datanode-cs-88.out
```

```
Starting secondary namenodes [0.0.0.0]
```

```
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
```

```
ECDSA key fingerprint is
```

```
67:12:a1:69:99:ea:b7:b7:96:b1:f5:4a:29:b5:d0:29.Are you sure you want
to continue connecting (yes/no)? y
```

```
Please type 'yes' or 'no': yes
```

```

0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known
hosts. 0.0.0.0: starting secondarynamenode, logging to /opt/hadoop-
2.7.0/logs/hadoop-test-secondarynamenode-cs-88.out
test@cs-88:/opt/hadoop-2.7.0$ 
jps- java machines proces
status test@cs-88:/opt/hadoop-
2.7.0$ jps5667 DataNode

```

CREATING DIRECTORY

```

test@cs-88:/opt/hadoop-2.7.0$ bin/hdfs dfs -mkdir /user/test@cs-88:/opt/hadoop-2.7.0$ 
bin/hdfs dfs -mkdir /exs test@cs-88:/opt/hadoop-2.7.0$ cd ..test@cs-88:/opt$ ls
hadoop-2.7.0 HDFSCCommands.pdf mrsampledadata(1).tar.gz
hadoop-2.7.0.tar.gz
jdk1.8.0_60 mrsampledadata.tar.gzHadoop Pseudo-Node.pdf jdk-8u60-linux-x64.gz
test@cs-88:/opt$ tar zxvf mrsampledadata.tar.gzfile2.txt
file5.txtfile1.txtfile4.txtfile3.txt
test@cs-88:/opt$
test@cs-88:/opt$ bin/hadoop jar share/hadoop/mapreduce/hadoop-
mapreduce-examples-2.7.0.jar grep /user/ /op '(CSE)'

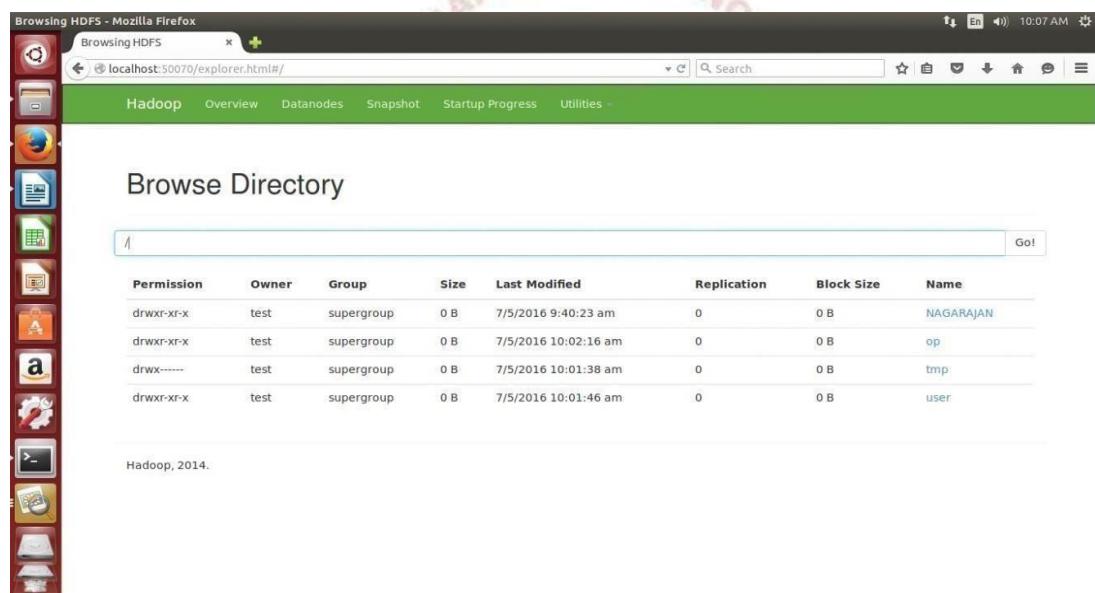
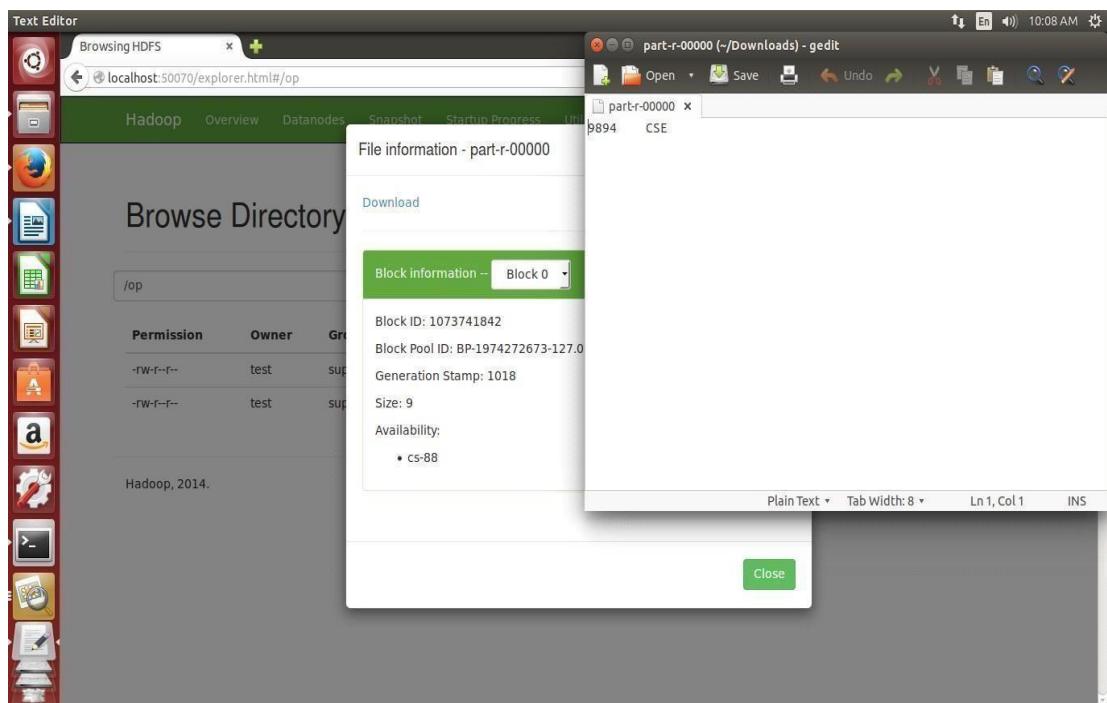
```

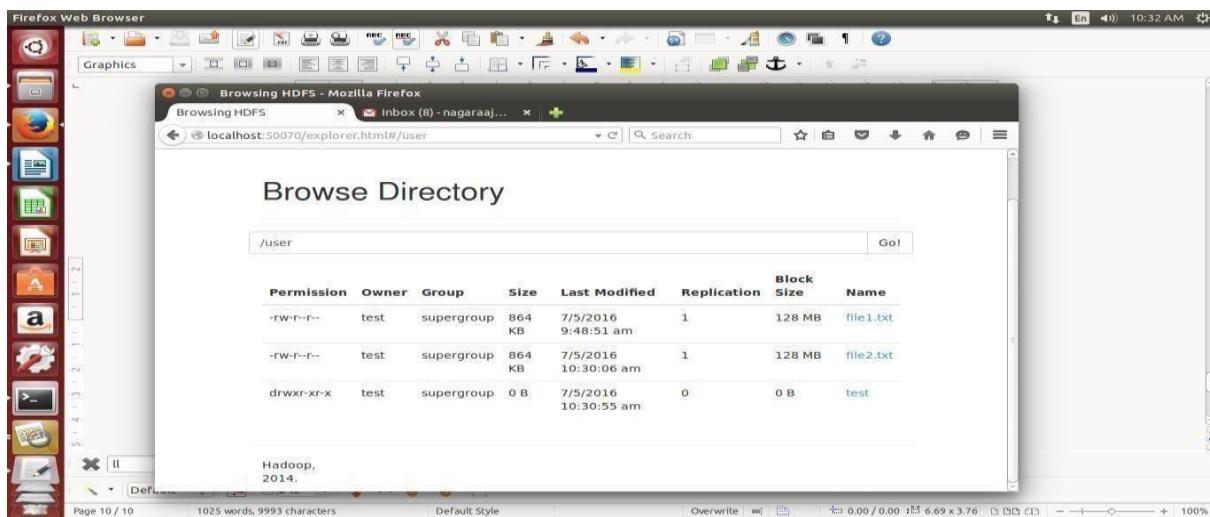
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	test	supergroup	0 B	7/5/2016 9:40:23 am	0	0 B	NAGARAJAN
drwxr-xr-x	test	supergroup	0 B	7/5/2016 10:02:16 am	0	0 B	op
drwx-----	test	supergroup	0 B	7/5/2016 10:01:38 am	0	0 B	tmp
drwxr-xr-x	test	supergroup	0 B	7/5/2016 10:01:46 am	0	0 B	user

 The status bar at the bottom left shows 'Page 9 / 9 1025 words, 9993 characters'. The status bar at the bottom right shows 'Default Style English (India) Overwrite 10:03 AM 100%'.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	test	supergroup	0 B	7/5/2016 10:02:16 am	1	128 MB	_SUCCESS
-rw-r--r--	test	supergroup	9 B	7/5/2016 10:02:16 am	1	128 MB	part-r-00000

 Below the table, it says 'Hadoop, 2014.' The status bar at the bottom left shows 'Page 9 / 9 1025 words, 9993 characters'. The status bar at the bottom right shows 'Default Style English (India) Overwrite 10:05 AM 100%'.





Word count program to demonstrate the use of Map and Reduce

tasksProcedure:

1. Format the path.
2. Start the dfs and check the no. of nodes running.
3. Start the yarn and check the no. of nodes running.
4. Open the browser and check whether the hadoop is installed correctly.
5. Add a file and check whether we can view the file.
6. Implement the grep command for the file added and see the result.
7. Implement the wordcount command for the file added and see the result.
8. After completing the process stop dfs and yarn properly.

Commands:

Install the hadoop cluster by using the commands

1. \$sudochown -Rh gee.gee/opt/
2. \$nano yarn –site.xml
<configuration>
<property>
<name>yarn.nodemanager.aux-services </name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
3. \$cd \$Hadoop _prefix
4. \$bin / Hadoopnamenode_format
5. \$s.bin /start_dfs.sh
6. jps

PROGRAM:

```
packagehadoo
p; import
java.util.*;
importjava.io.IOException;
importjava.io.IOException;
importorg.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*
; import org.apache.hadoop.io. * ;
importorg.apache.hadoop.m
apred.* ;import
org.apache.hadoop.util.* ; public
class ProcessUnits
{
public static class E_EMapper extends
MapReduceBaseimplementsMapper<LongWritable Text,
Text,
IntWrita
ble>
{
public void m ap(LongWritable key, Text
value,OutputCollector<Text, IntWritable>
output, Reporter reporter) throws
IOException{
String line =
value.toString();String
lasttoken = null;
StringTokenizer s = new
StringTokenizer(line, "\t");String year =
s.nextToken(); while(s.hasMoreTokens()){
lasttoken=s.nextToken();
}
intavgprice = Integer.parseInt(lasttoken);
output.collect(new Text(year), new
IntWritable(avgprice));
}
}

public static class E_EReduce extends
MapReduceBaseimplementsReducer< Text, IntWritable, Text,
IntWritable>
{
```

```
public void reduce( Text key, Iterator <IntWritable> values, OutputCollector<Text, IntWritable>
output, Reporter reporter)
throws IOException{
int m axavg=30;
intval=Integer.MIN_VALU;
while (values.hasNext())
{
if((val=values.next().get())>m axavg)
{
output.collect(key, new IntWritable(val));
}
}
}
}

public static void m ain(String args[])throws Exception
{
JobConfconf = new JobConf(Eleunits.class);
conf.setJobNam e("m ax_eletricityunits");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass(E_EMapper.class);
conf.setCombinerClass(E_EReduce.class);
conf.setReducerClass(E_EReduce.class);
conf.setInputForm at(TextInputFormat.class);
conf.setOutputForm at(TextOutputFormat.class);
FileInputFormat.setInputPaths(conf, new
Path(args[0])); FileOutputFormat.setOutputPath(conf,
new Path(args[1]));JobClient.runJob(conf);
}
}
```

OUTPUT

The screenshot shows a Mozilla Firefox browser window titled "Browsing HDFS". The address bar displays "localhost:50070/explorer.html#/output". The main content area is titled "Browse Directory" and shows a list of files in the "/output" directory. A file named "part-r-00000" is selected, and a modal dialog box titled "File information - part-r-00000" is displayed. The dialog box contains the following details:

Block information	Block 0
Block ID:	1073741842
Block Pool ID:	BP-1899221130-127.0.0.1-1471411683839
Generation Stamp:	1018
Size:	10
Availability:	localhost

On the right side of the dialog, there is a table showing block sizes and names:

Block Size	Name
128 MB	SUCCESS
128 MB	part-r-00000

A "Close" button is located at the bottom right of the dialog.

Result:

Thus the mapping & reducing is successfully done using word count program using Hadoop single node cluster.

EXP : 8	Creating and Executing Your First Container Using Docker.
DATE:	

Step 1: Install Docker (If Not Installed)

Before running Docker containers, ensure that Docker is installed on your system.

- **Windows & Mac:** Download Docker Desktop from Docker's official website.
- **Linux:** Install using the command:

```
sudo apt update
sudo apt install docker.io -y
sudo systemctl start docker
sudo systemctl enable docker
```

- Verify installation:

```
docker --version
```

Step 2: Pull a Docker Image

Docker containers run from pre-built images. Let's pull the official Ubuntu image.

```
docker pull ubuntu
```

- This command downloads the latest **Ubuntu** image from **Docker Hub**.

Step 3: Run a Docker Container

Now, create and start a container using the downloaded Ubuntu image.

```
docker run -it ubuntu
```

- **-it:** Runs the container in **interactive mode**, allowing you to execute commands inside.
- **ubuntu:** The image name.

Once inside the container, you'll see a command prompt like:

```
root@<container_id>:/#
```

Step 4: Execute Commands inside the Container

Try running basic Linux commands inside your container:

```
ls  
pwd  
echo "Hello from Docker!"
```

Step 5: Exit and List Running Containers

- Exit the container by typing:

```
exit
```

- List all running containers:

```
docker ps
```

- List all containers (including stopped ones):

```
docker ps -a
```

Step 6: Restart and Remove a Container

- Restart the container using its **Container ID**:

```
docker start <container_id>  
docker attach <container_id>
```

- Stop and remove the container:

```
docker stop <container_id>  
docker rm <container_id>
```

Step 7: Remove Docker Image

To clean up, remove the Ubuntu image:

```
docker rmi ubuntu
```

EXP : 09	
DATE:	Run a Container from Docker Hub

1. Install Docker (if not installed)

- Download and install Docker from Docker's official site.
- Ensure Docker is running.

2. Pull an Image from Docker Hub

- You can directly pull an image using:

```
docker pull <image-name>
```

- Example: Pull the latest Ubuntu image:

```
docker pull ubuntu:latest
```

3. Run the Container

- Use the docker run command:

```
docker run -d --name <container-name> <image-name>
```

- Example: Run an Nginx container in detached mode:

```
docker run -d --name my-nginx -p 8080:80 nginx
```

- -d: Runs the container in the background.
- --name my-nginx: Names the container "my-nginx".
- -p 8080:80: Maps port 80 inside the container to port 8080 on your machine.

4. Check Running Containers

- View running containers:

```
docker ps
```

- View all containers (including stopped ones):

```
docker ps -a
```

5. Access the Running Container

- Open a shell inside the container:

```
docker exec -it <container-name> /bin/bash
```

- Example:

```
docker exec -it my-nginx /bin/sh
```

6. Stop and Remove the Container

- Stop the container:

```
docker stop <container-name>
```

- Remove the container:

```
docker rm <container-name>
```

EXP : 10	
DATE:	

Creating Virtual Machine Using Open Nebula.

Aim:

To Install Virtualbox / VMware Workstation with different flavours of linux or windowsOS on top of windows 10 through open Nebula.

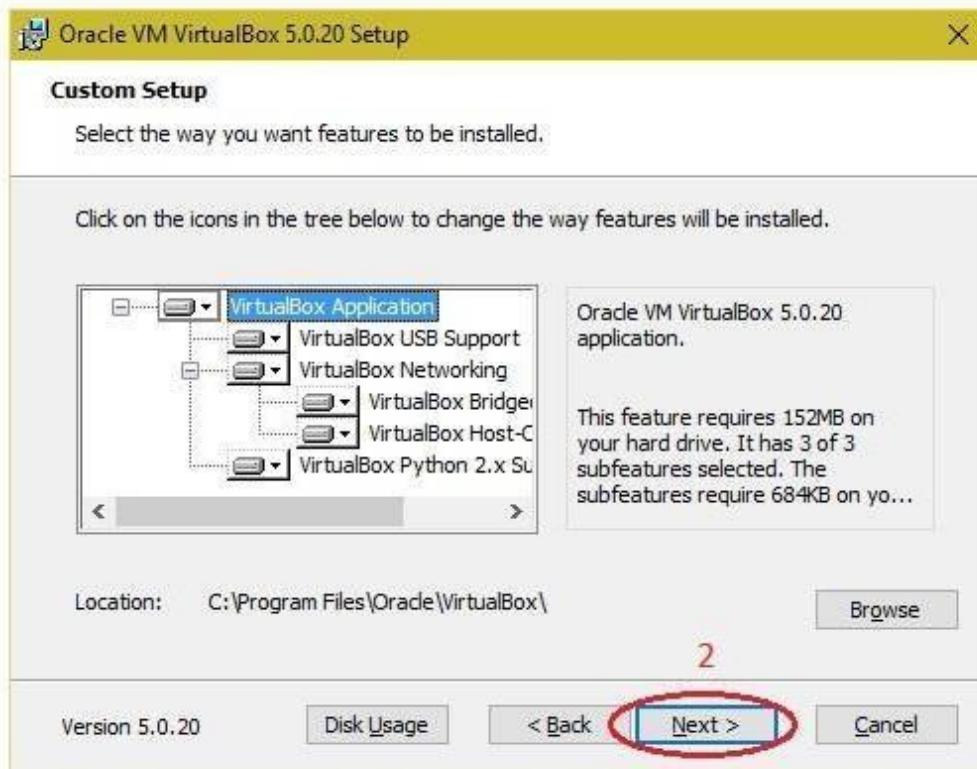
PROCEDURE:

Steps to install Virtual Box:

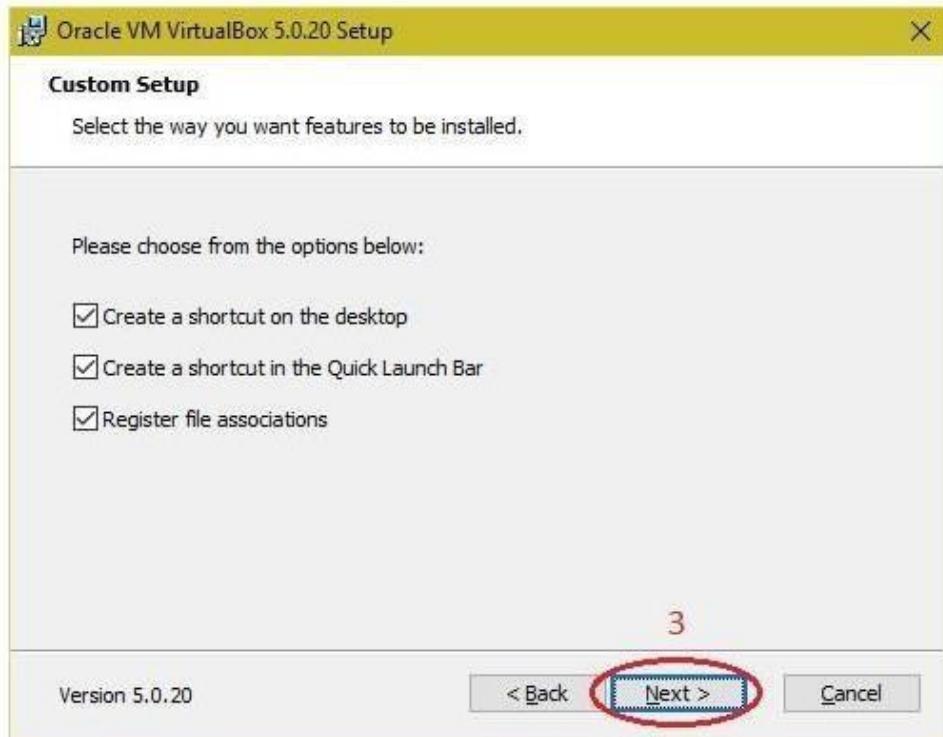
1. Download the Virtual box exe and click the exe file...and select next button..



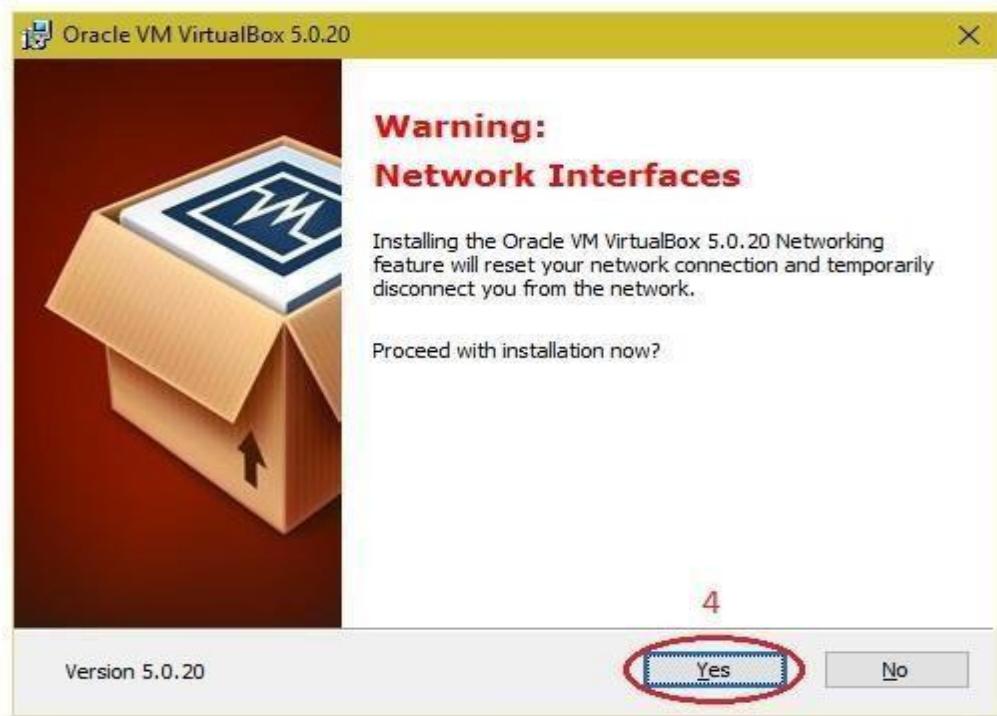
2. Click the next button..



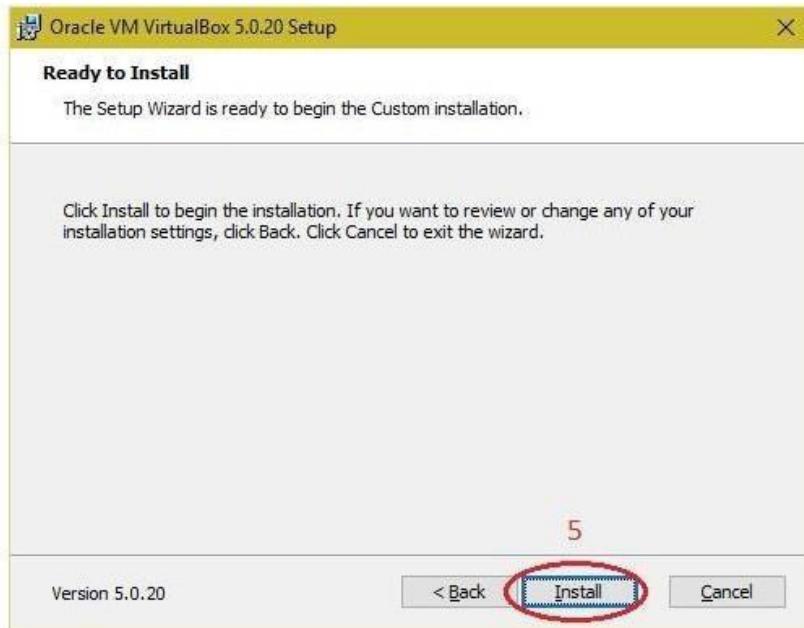
3. Click the next button



4. Click the YES button..



5. Click the install button...

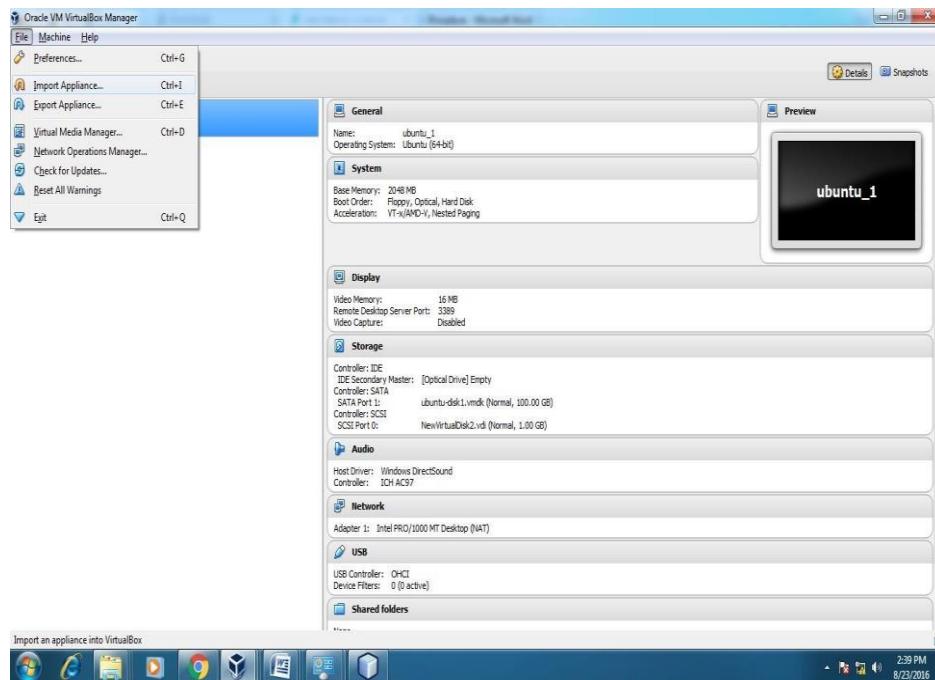


6. Then installation was completed..the show virtual box icon on desktop screen....

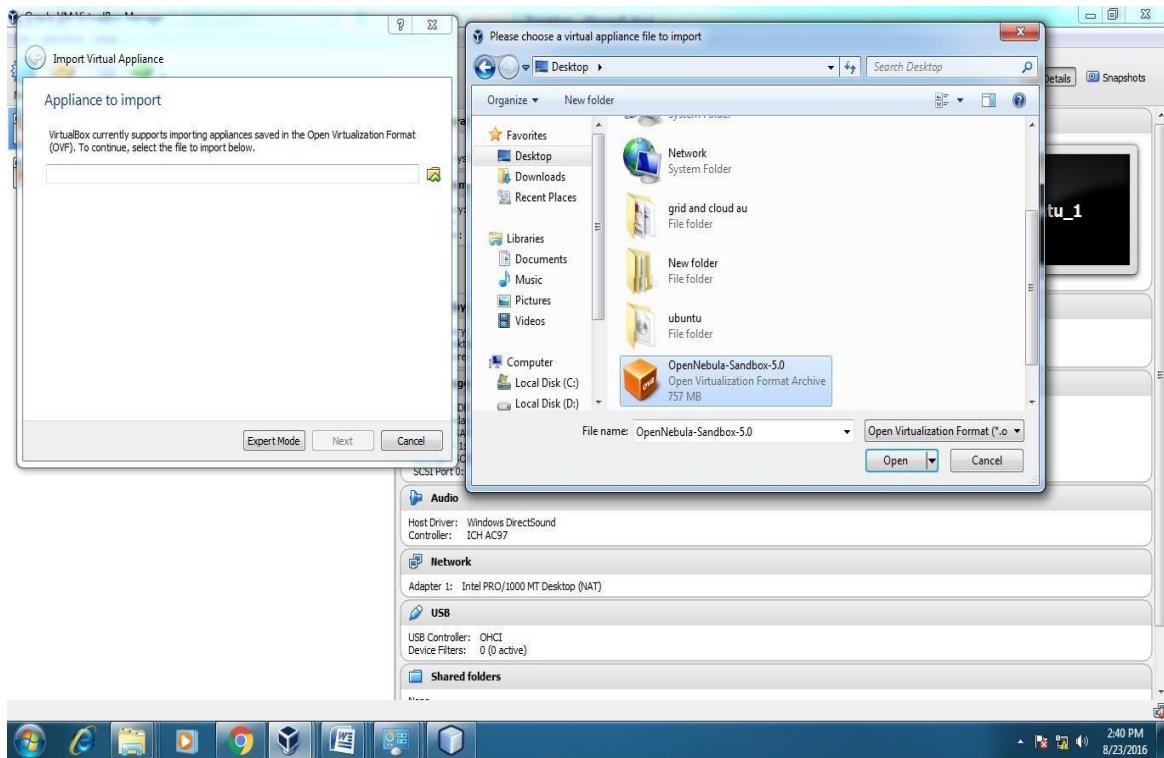


Steps to import Open nebula sandbox:

1. Open Virtual box
2. File → Import Appliance
3. Browse OpenNebula-Sandbox-5.0.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the Open Nebula

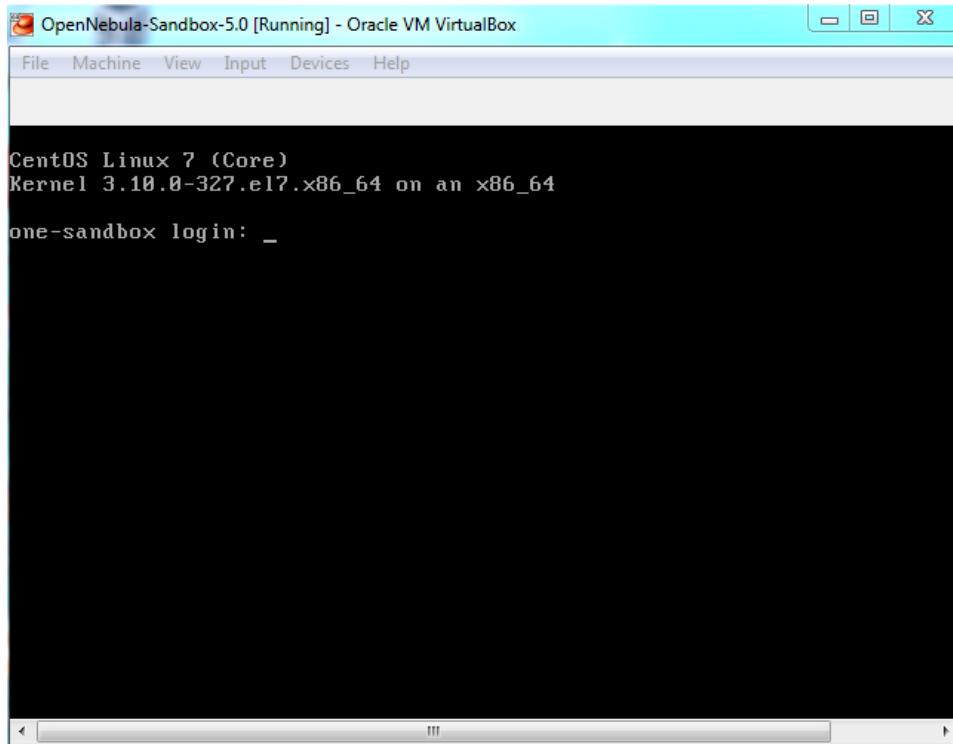


6. Login using username: root, password:opennebula.

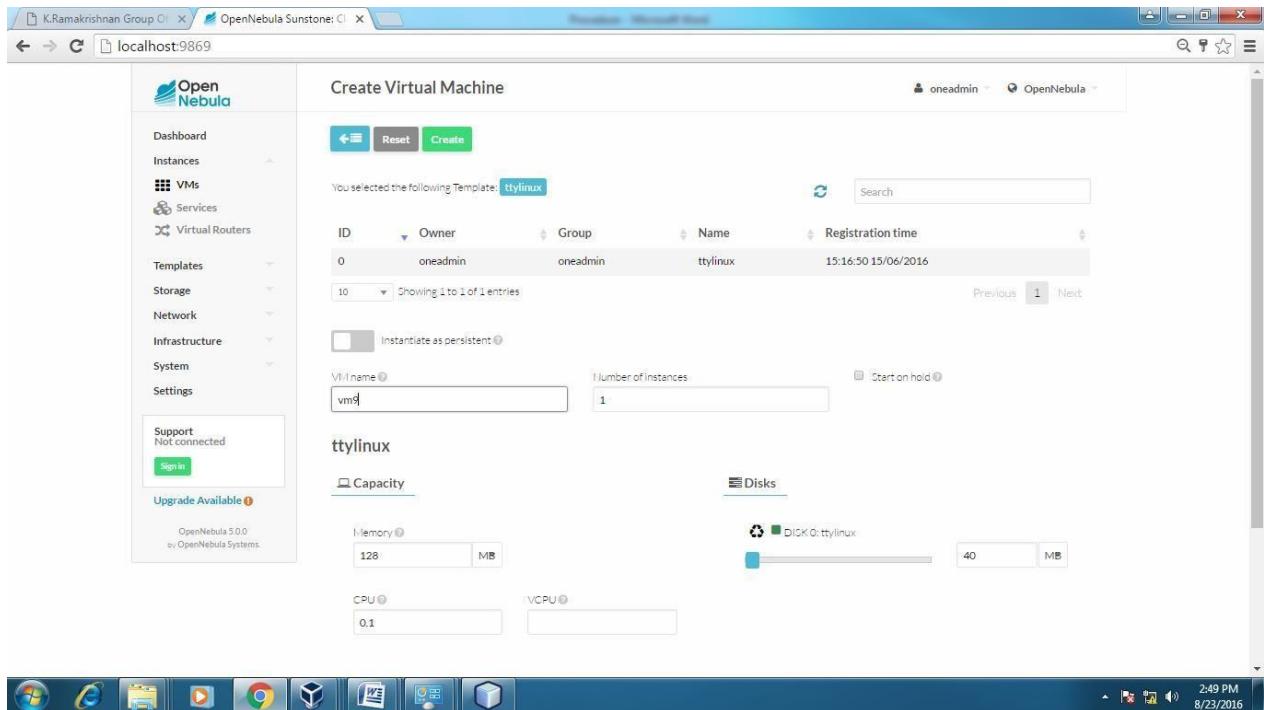


Steps to create Virtual Machine through opennebula

1. Open Browser, type localhost:9869
2. Login using username: oneadmin, password: opennebula
3. Click on instances, select VMs then follow the steps to create Virtaul machine
 - a. Expand the + symbol
 - b. Select user oneadmin
 - c. Then enter the VM name,no.of instance, cpu.
 - d. Then click on create button.
 - e. Repeat the steps the C,D for creating more than one VMs.







Result:

Thus the procedure to run the virtual machine of different configuration using open Nebula was done successfully.