

**DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE**

**(AUTONOMOUS)**

(Approved by AICTE & Affiliated to Anna University, Chennai)  
Accredited with 'A' Grade by NAAC, Accredited by TCS  
Accredited by NBA with BME, ECE & EEE

**PERAMBALUR - 621 212. Tamil Nadu.**

website : [www.dsengg.ac.in](http://www.dsengg.ac.in)



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**LAB MANUAL**



**U23CSP71 / CLOUD LABO RATORY**

**III YEAR / V SEMESTER**

**REGULATION 2023**

**Prepared By**

**AP / IT**

**Verified By**

**HOD/IT**

**U23CSP71**

**CLOUD LABORATORY**

**L T P C**  
**0 0 4 2**

**Pre-requisite:** Learn to run virtual machine of different configuration

**COURSE OBJECTIVES:**

1. To develop web applications in cloud
2. Be exposed to tool kits for grid and cloud environment.
3. To learn the design and development process involved in creating a cloud based application.
4. To learn to implement and use parallel programming using Hadoop.

**LIST OF EXPERIMENTS:**

1. Install Virtual box/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute SimplePrograms
3. Install Google App Engine. Create hello world app and other simple web applications using python/java.
3. Use GAE launcher to launch the web applications.
4. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present inCloudSim.
5. Find a procedure to transfer the files from one virtual machine to another virtual machine.
6. Find a procedure to launch virtual machine using trystack (Online Open stack Demo Version)
7. Install Hadoop single node cluster and run simple applications like word count

**TOTAL: 60 PERIODS**

**COURSE OUTCOMES:**

**Learners are able to**

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

**DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE  
(AUTONOMOUS)**



(Approved by AICTE & Affiliated to Anna University, Chennai)  
Accredited with 'A' Grade by NAAC, Accredited by TCS  
Accredited by NBA with BME, ECE & EEE  
**PERAMBALUR - 621 212. Tamil Nadu.**  
website : [www.dsengg.ac.in](http://www.dsengg.ac.in)



**LIST OF EXPERIMENTS**

<b>Ex. No.</b>	<b>Name of the Experiment</b>
1.	a) Installation of Virtual box with fedora OS on Windows 7m
	b) Installation of VMware Workstation with Ubuntu on Windows 7
2.	Installation of C compiler and C program execution
3.	Installation of Google App Engine. Creation of hello world app web applications using python.
4.	The web applications launching using GAE launcher
5.	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim
6.	Procedure to transfer the files from one virtual machine to another virtual machine
7.	Procedure to launch virtual machine using trystack (Online Open stack Demo Version)
8.	Installation of Hadoop single node cluster and run simple word count applications

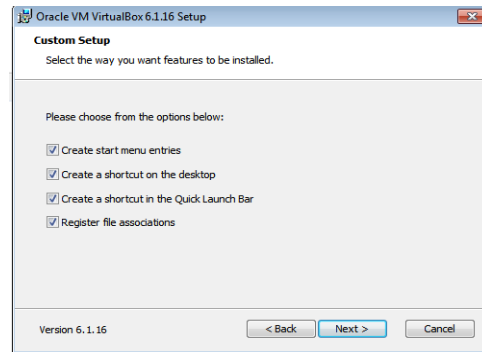
<b>Ex.No :</b>	<b>Installation of Virtual box with Fedora OS on Windows 7</b>
<b>Date:</b>	

**AIM:**

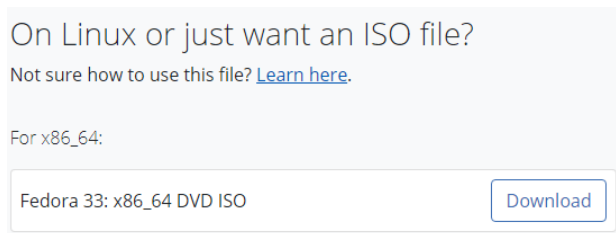
To Install Virtual box Workstation with Fedora operating system on top of windows7

**PROCEDURE:**

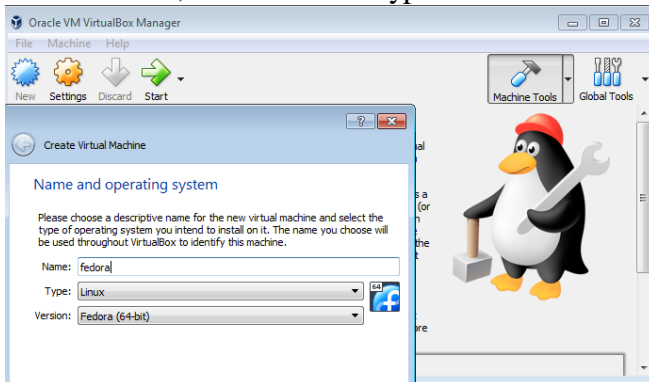
1. Download Virtual box from <https://www.virtualbox.org/wiki/Downloads> and click appropriate platform (here Windows hosts),
2. Open downloaded Virtualbox Application, Click next to continue



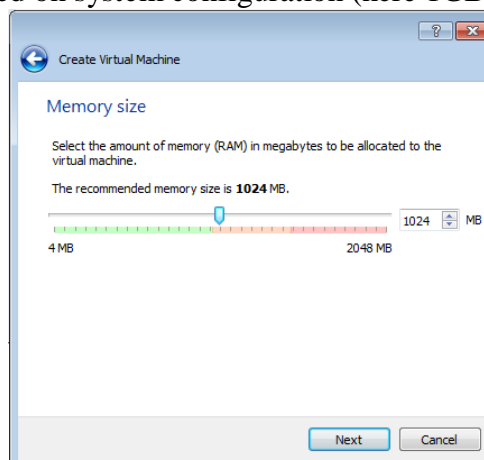
3. When finished installation, close the window
4. Visit <https://getfedora.org/en/workstation/download/> and click download to download iso file of fedora OS



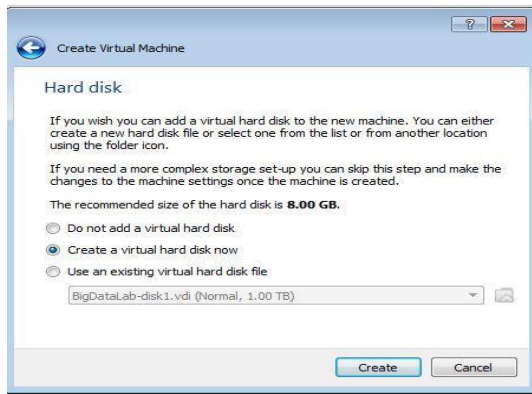
5. Open Virtual box, click new and type fedora in name field



6. Choose RAM memory size based on system configuration (here 1GB)



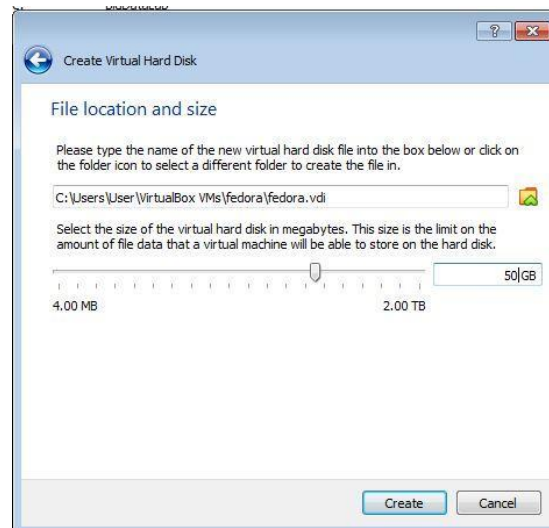
7. Select Create a virtual hard disk now and select VDI as hard disk file



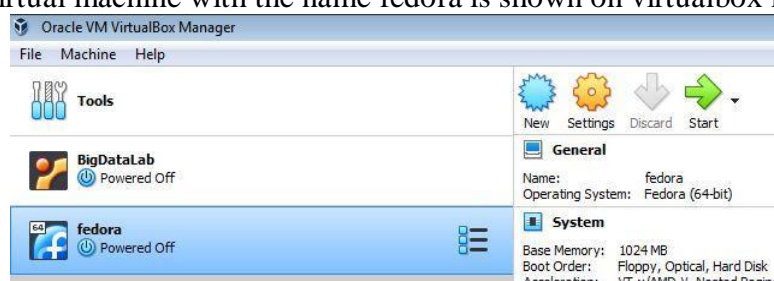
8. Choose dynamically allocated in Storage on Physical hard disk,



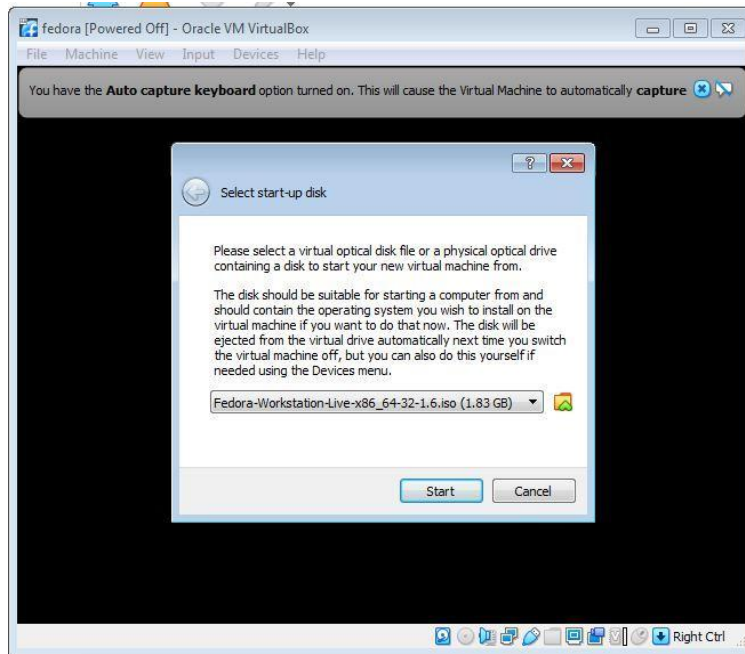
9. Click the folder icon and choose the fedora iso file you downloaded, Next give hard disk size (here 50GB), next click create



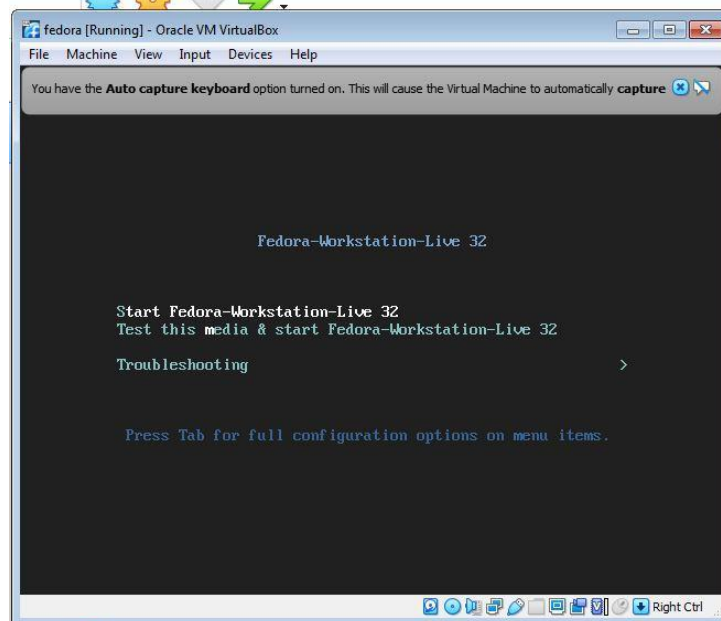
10. After creation, the virtual machine with the name fedora is shown on virtualbox manager window



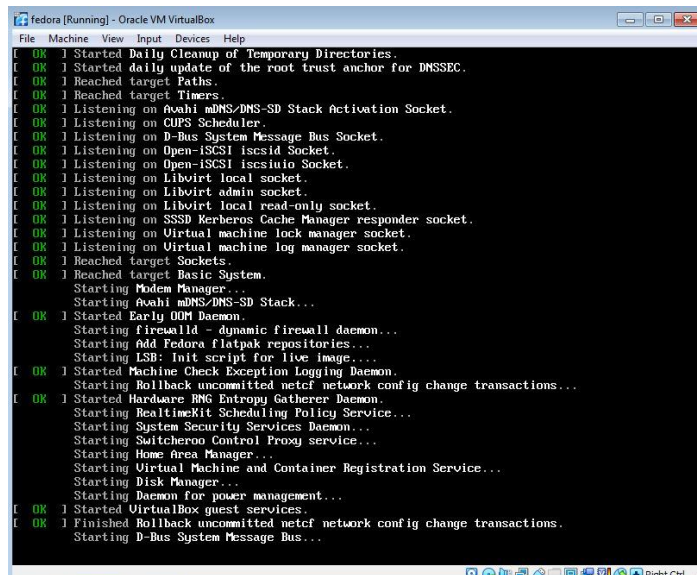
11. Choose fedora and click start and click folder icon and select downloaded fedora iso file then click start



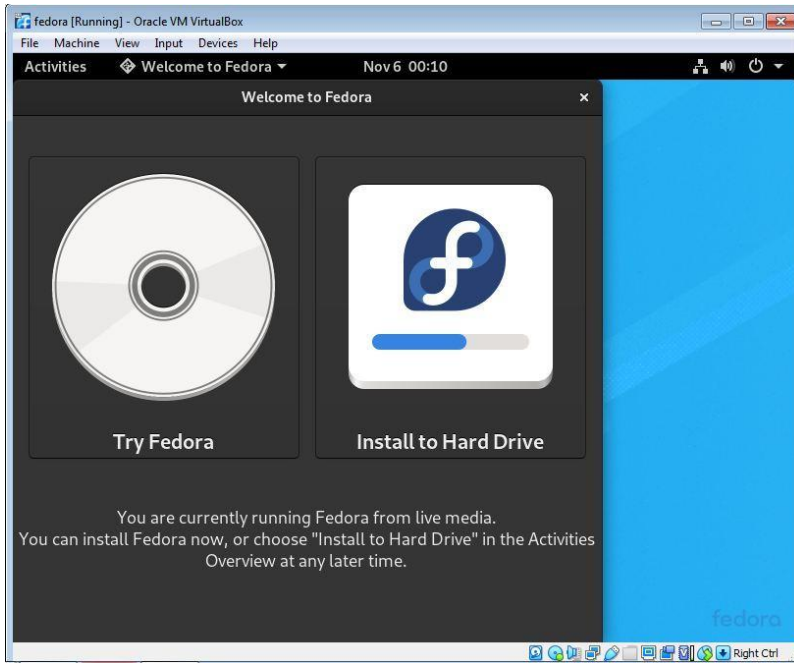
12. Hit “s” to start fedora workstation without media testing ( or you may go with second option )



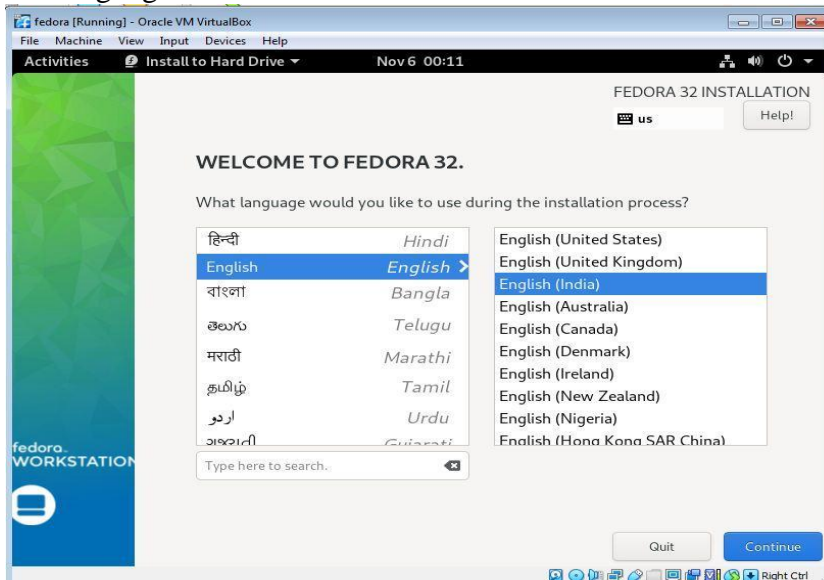
13. Fedora get started



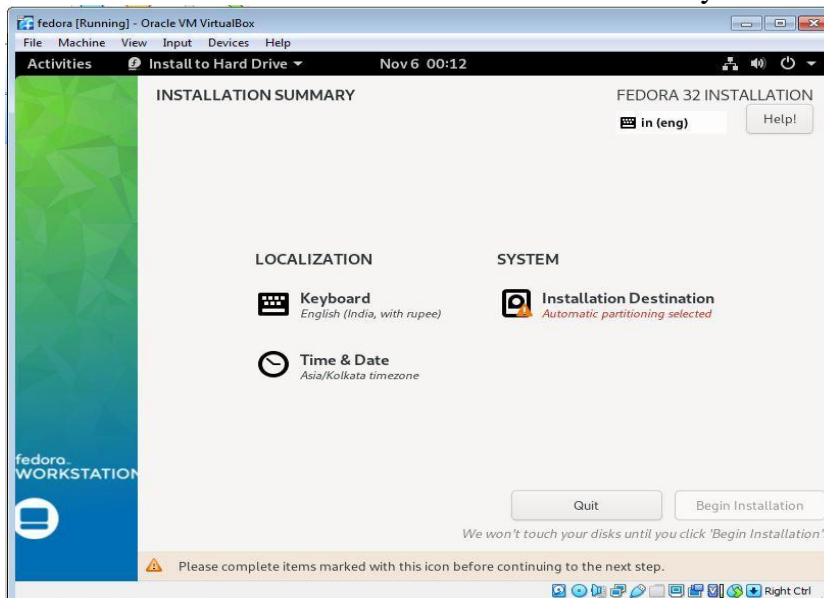
#### 14. To install fedora into the hard drive click Install to Hard Drive



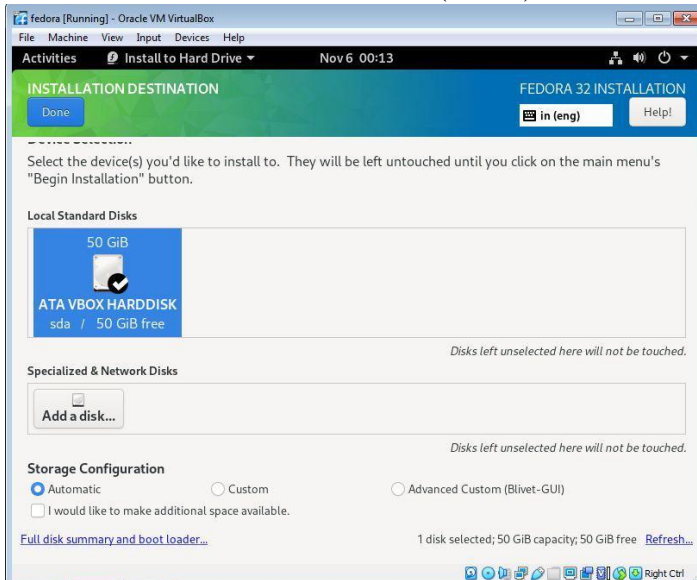
#### 15. Choose language and click continue



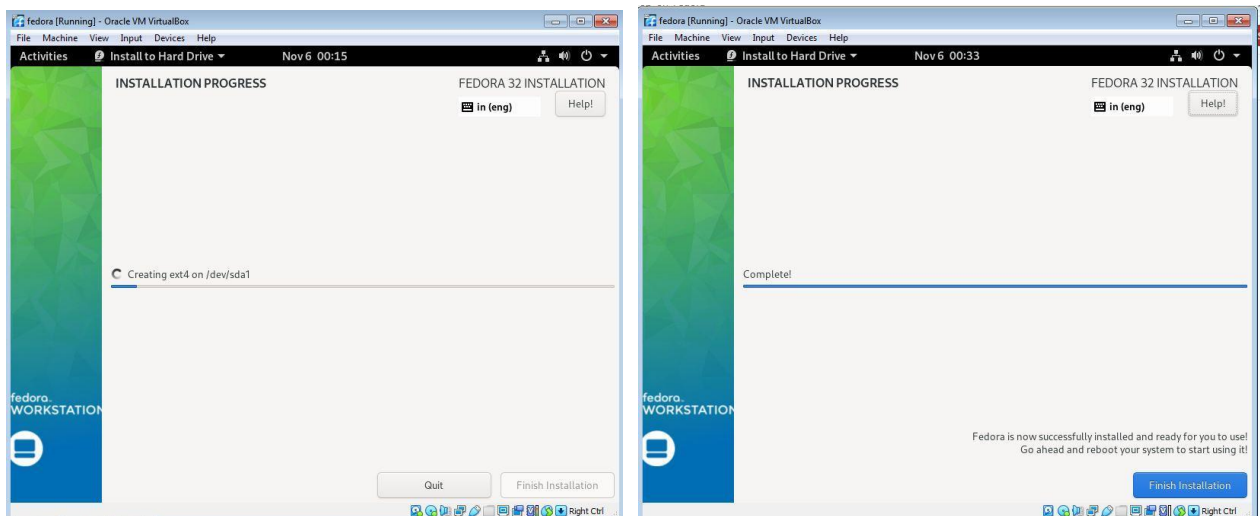
#### 16. Now click installation Destination to ensure the availability of Hard disk size



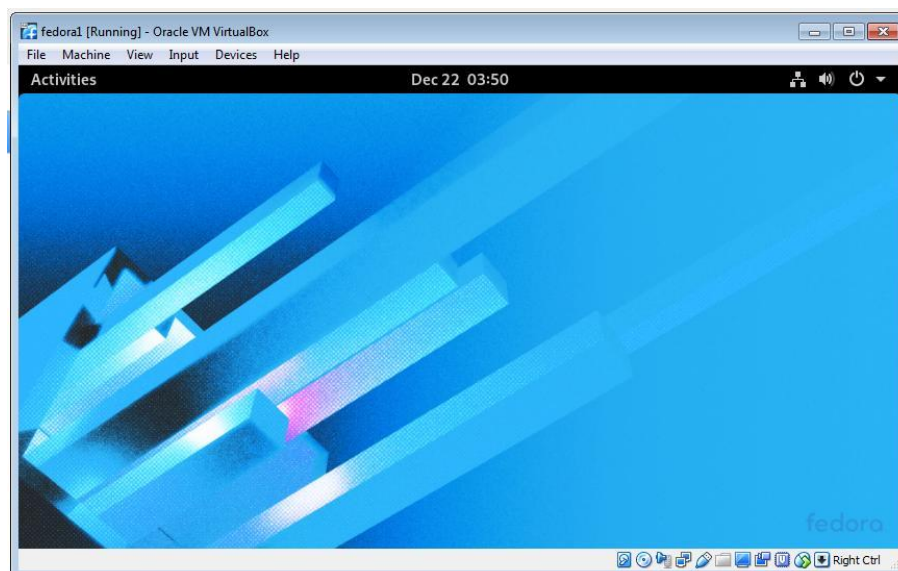
17. Ensure that the Local Standard Disks (50GB) is checked and click Done



18. Now installation process getting started, after completion click Finish Installation



19. Click finish installation, now fedora is running on Virtual machine



**RESULT:**

Thus the Installation of Virtualbox Workstation with Fedora on top of windows7 was done successfully

**EX.NO: 1b**  
**DATE:**

## Installation of VMware workstation with Ubuntu OS on Windows 7

### AIM:

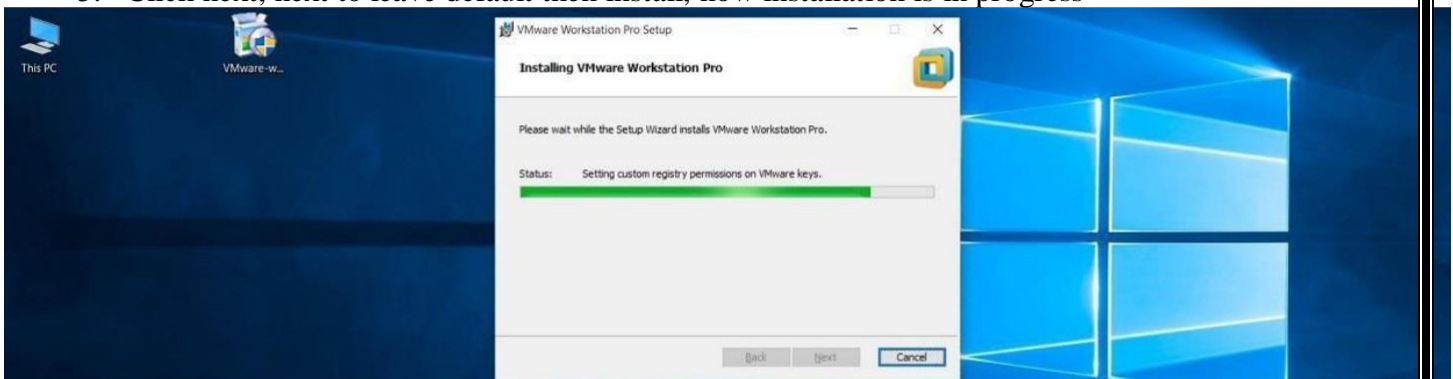
To Install VMware Workstation with Ubuntu on top of windows7

### PROCEDURE:

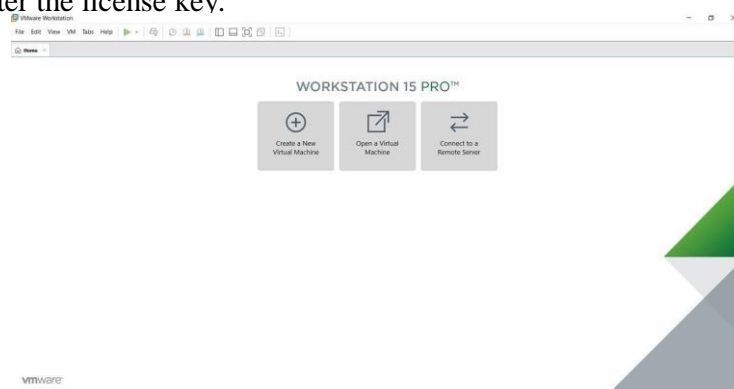
1. Download the VMware workstation from <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>. Download the software for windows. (Note: File name should be something like VMware-workstation-full-15.5.1-15018445.exe. This file name can change depending on the version of the software currently available for download)
2. Locate the downloaded installer file, right click and choose Run as administrator, Click yes to continue. In VMware workstation wizard, click next to continue



3. Check "I accept the terms in the Licence Agreement" box and press next to continue
4. 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.
5. Click next, next to leave default then install, now installation is in progress






6. Click finish and you are done with the installation process. You may be asked to restart your computer. Click on Yes to restart. Open the VMware, the dialog box appears, asking for license key, click on trial or enter the license key.

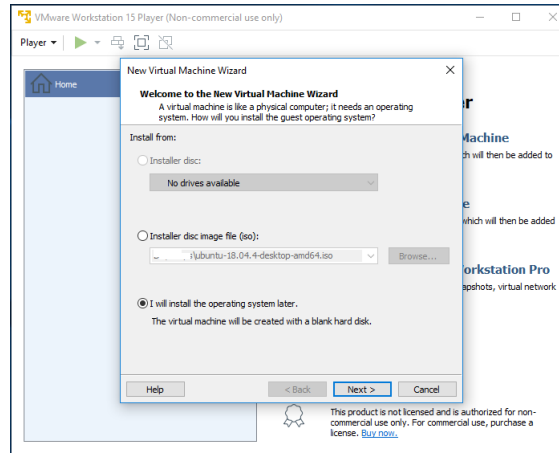


## Install guest OS ubuntu 18.04 using VMWare

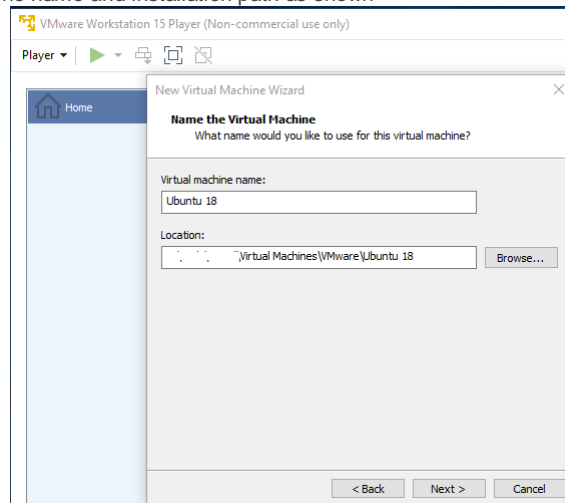
7. Download ubuntu from <https://old-releases.ubuntu.com/releases/18.04.4/>,

 ubuntu-18.04-server-5390x.template	2018-04-20 10:34 114M
 ubuntu-18.04.1-desktop-amd64.iso	2018-07-25 03:22 1.8G
 ubuntu-18.04.1-desktop-amd64.iso.torrent	2018-07-26 16:55 73K

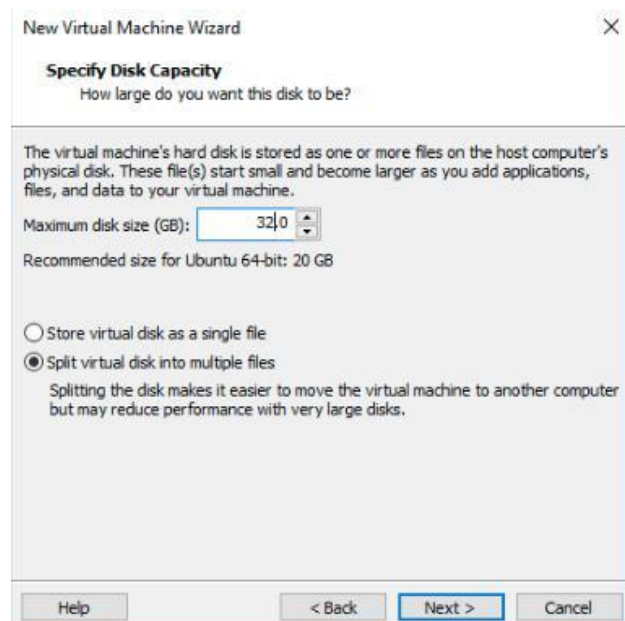
8. In VMWare, File -> New Virtual Machine OR Create a New Virtual Machine Link as highlighted in Fig 2. Choose the option - I will install the operating system later



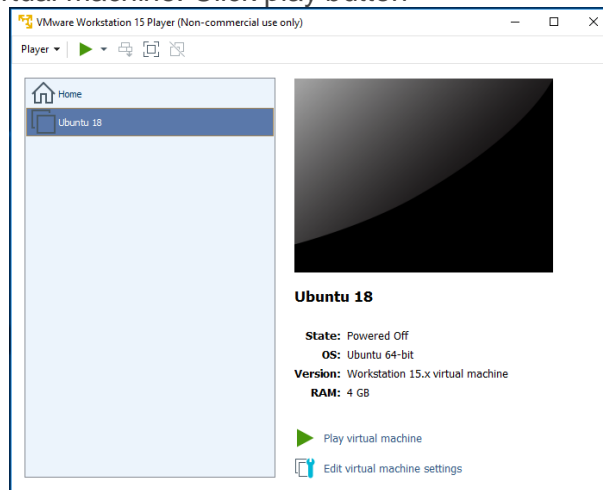
9. Click next, then Select Linux as the Guest operating system and Ubuntu 64-bit as the Version. Now click the Next Button to configure virtual machine name and installation path as shown



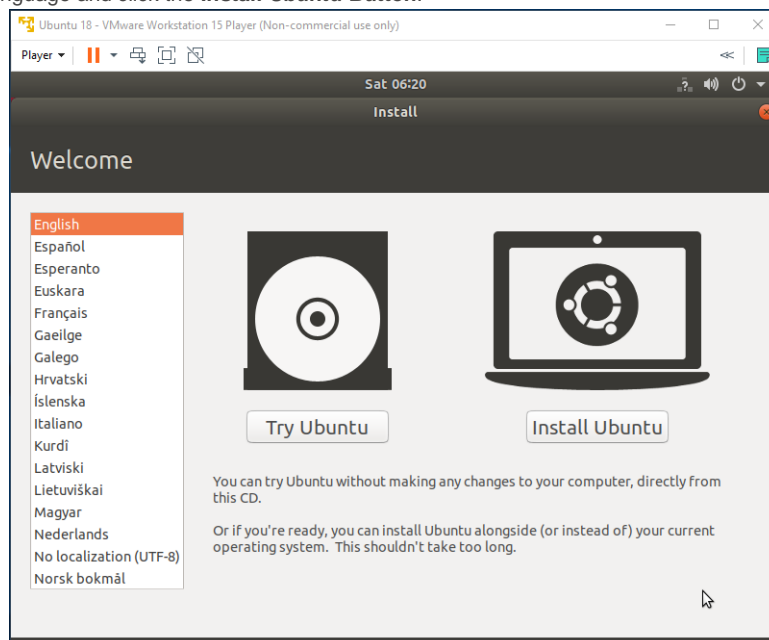
10. Click the Next Button to configure the disk and allocate space



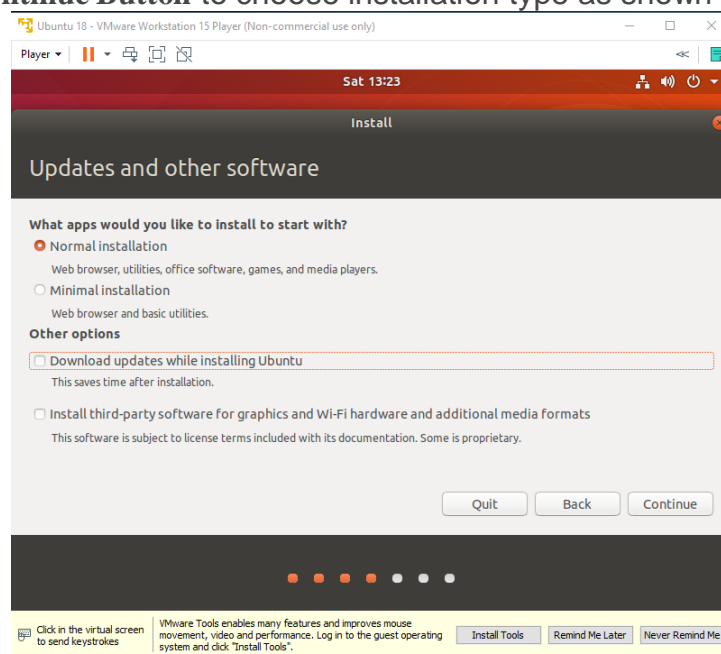
11. Now click the **Close Button** to finalize the hardware for the virtual machine and click **Finish Button** to create the virtual machine. Click play button

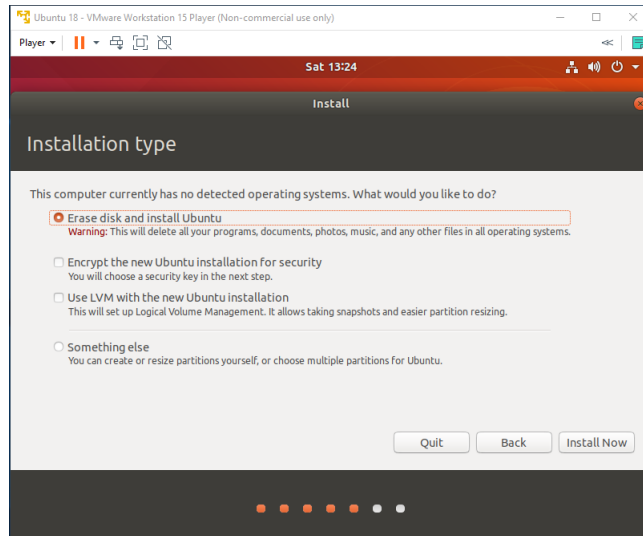


12. Choose appropriate language and click the **Install Ubuntu Button**.

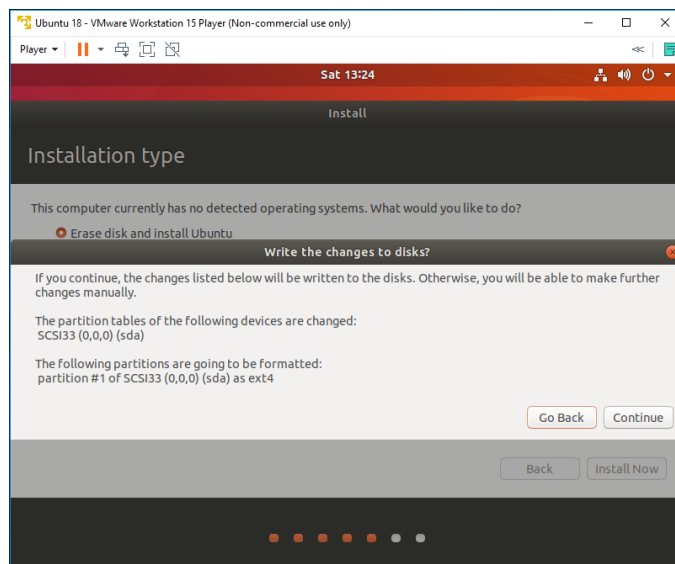


13. Now click the **Continue Button** to choose installation type as shown below

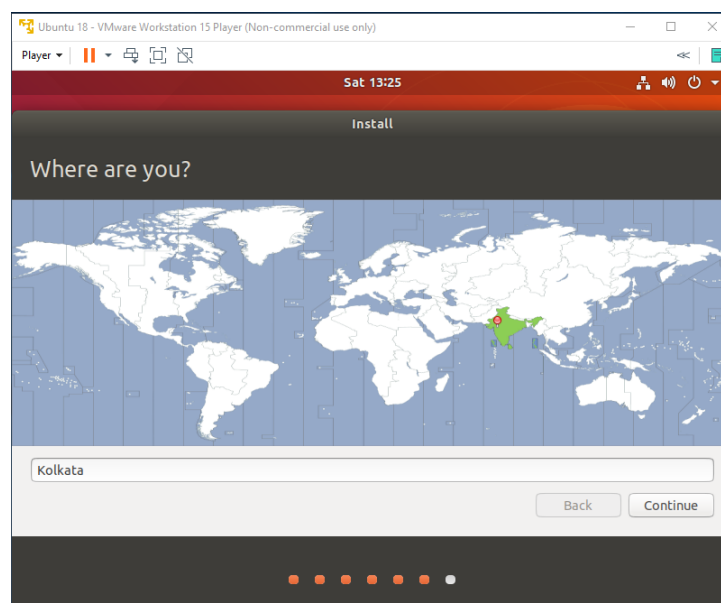


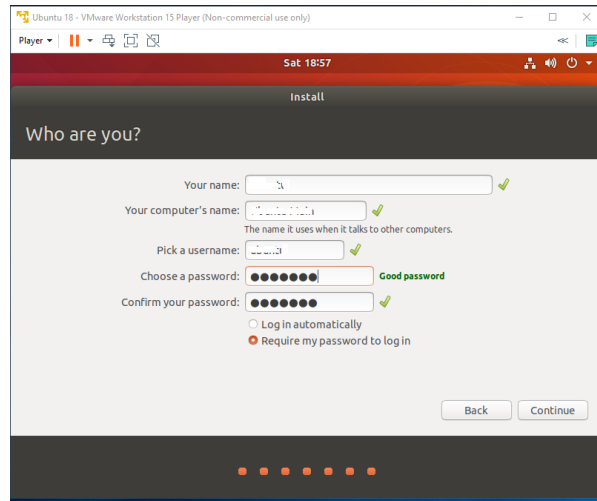


14. Click the **Install Now Button** to start installing Ubuntu 18.04. It will also ask to confirm the disk changes

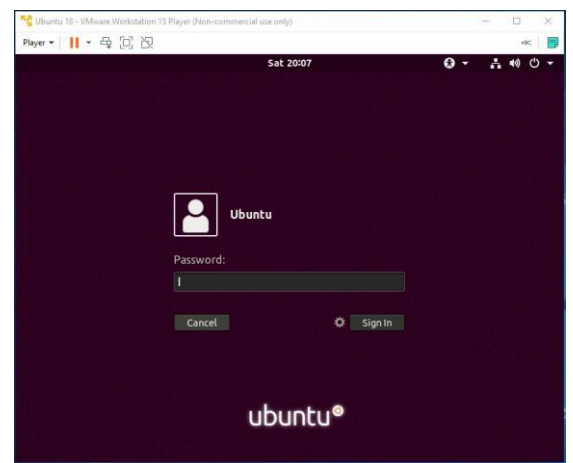
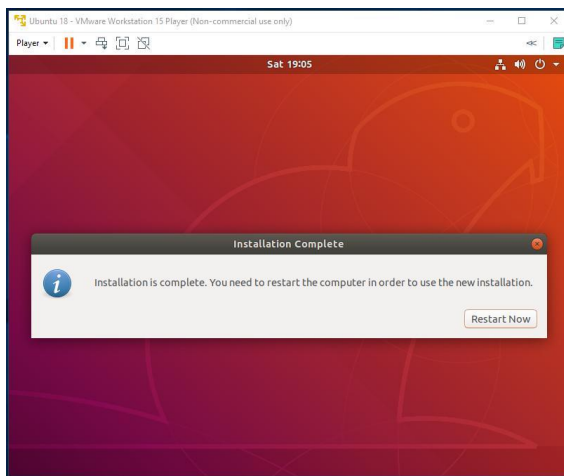


15. Now click the Continue Button to choose the Timezone and also configure the root user

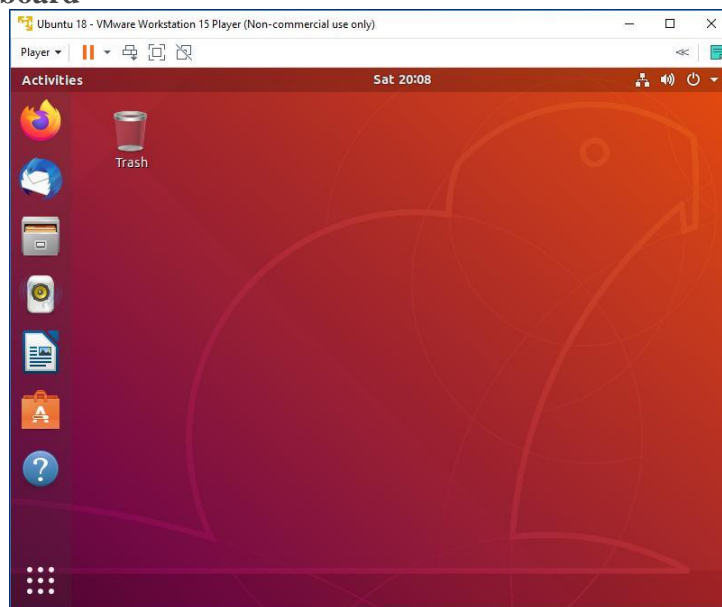




16. Now click the **Continue Button** to continue the installation. It will show the installation progress, Click the **Restart Now Button** to restart Ubuntu It will ask to remove the Installation media, press the **Enter Key** to continue



17. Now enter the password provided by you while installing Ubuntu. The player will show the **Ubuntu Dashboard**



**RESULT:**

Thus the Installation of VMWare Workstation with Ubuntu on top of windows7 was done successfully

**EX.NO: 2**  
**DATE:**

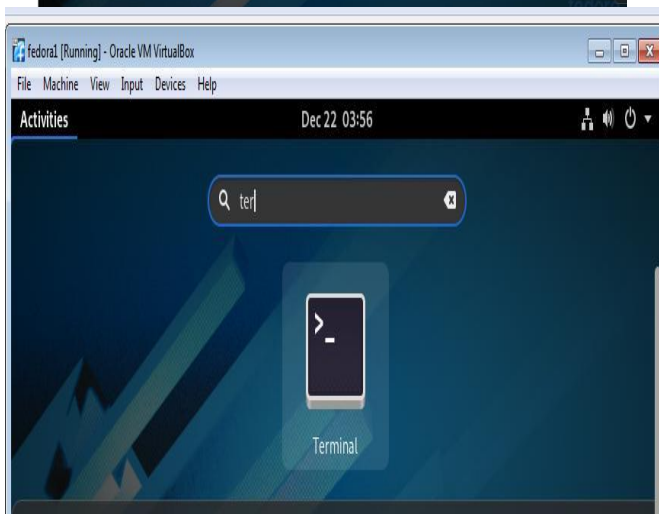
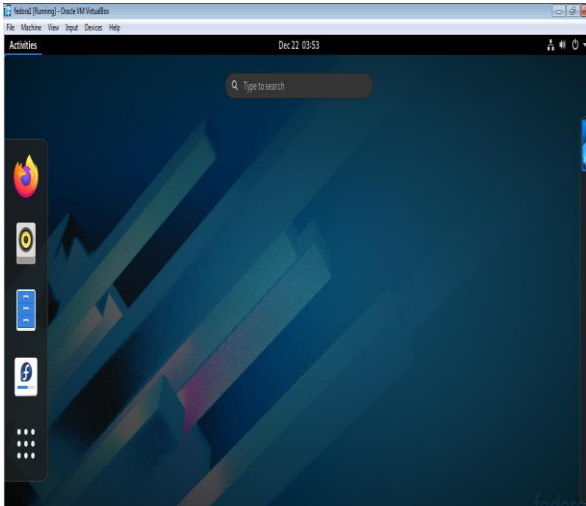
## Installation of C compiler and C program execution

### AIM:

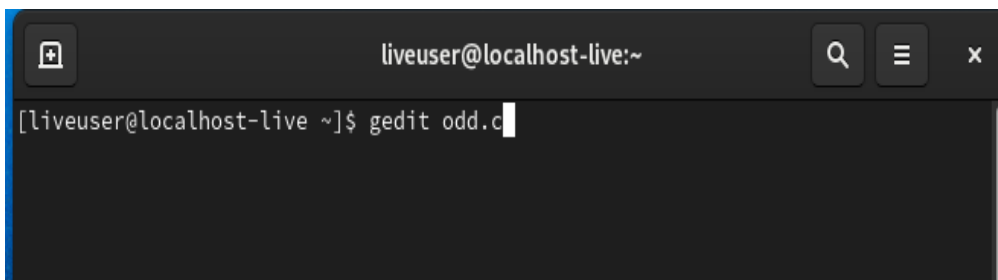
To Install a C compiler in the virtual machine created using virtual box and execute Simple Programs

### PROCEDURE:

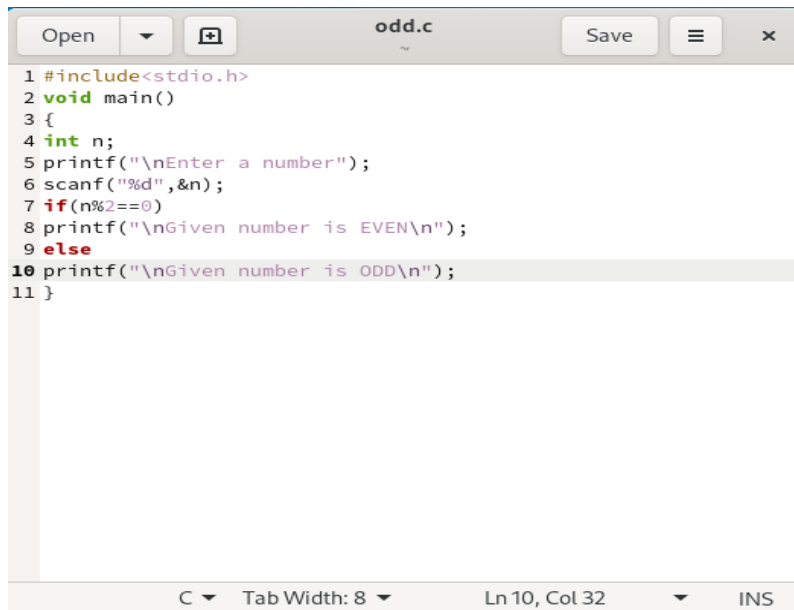
1. In fedora, to open terminal, Click Activities, then in type of search, type terminal



2. In terminal window, open text editor by typing gedit followed by filename (here, odd.c)



3. The gedit opens an editor, now Type the program in the editor window and save it



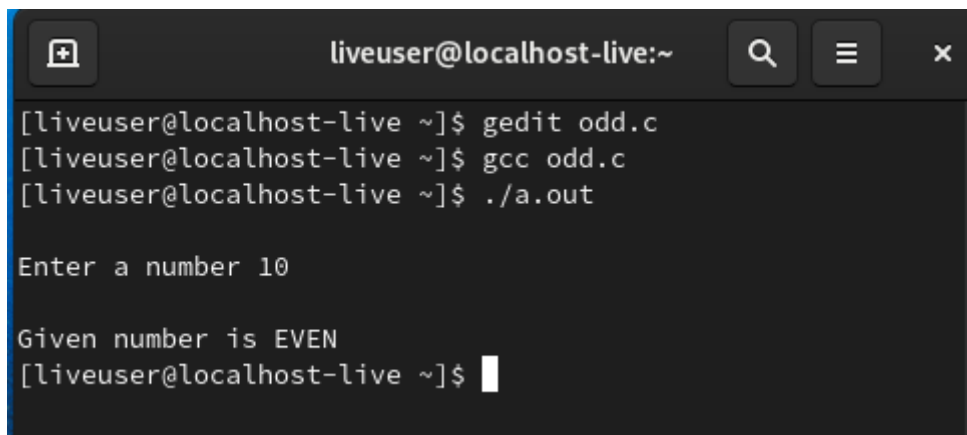
```
1 #include<stdio.h>
2 void main()
3 {
4     int n;
5     printf("\nEnter a number");
6     scanf("%d",&n);
7     if(n%2==0)
8     printf("\nGiven number is EVEN\n");
9     else
10    printf("\nGiven number is ODD\n");
11 }
```

4. Compile the program using **gcc filename.c** and to run the program **./a.out**, we get the output (if program have error then during compilation steps it throws error in the terminal)

**PROGRAM:**

```
#include<stdio.h>
void main()
{
int n;
printf("\n Enter a Number");
scanf("%d",&n);
if(n%2==0)
printf("\n Given number is EVEN\n");
else
printf("\n Given number is ODD\n");
}
```

**OUTPUT:**



```
liveuser@localhost-live:~
[liveuser@localhost-live ~]$ gedit odd.c
[liveuser@localhost-live ~]$ gcc odd.c
[liveuser@localhost-live ~]$ ./a.out

Enter a number 10

Given number is EVEN
[liveuser@localhost-live ~]$
```

**RESULT:**

Thus the Installation of a C compiler in the virtual machine created using virtual box and successfully executed Simple C Program.

**EX.NO: 3**  
**DATE:**

## Installation of Google App Engine. Creation of hello world app web applications using python

### AIM:

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

### PROCEDURE:

1. Download cloud SDK from <https://cloud.google.com/appengine/downloads>, Click python then click Install and initialize the Cloud SDK (ensure python was installed in system)

App Engine > Documentation > Standard Environment > Python 3 > Guides

Rate and review  

# Setting Up Your Development Environment

[Send feedback](#)

Python [2.7/3](#) | [Java 8/11](#) | [PHP 5/7](#) | [Ruby](#) | [Go 1.11 / 1.12+](#) | [Node.js](#)

To set up your environment for developing on Python 3:

1. Install the latest release of Python 3.

See [Python 3 Runtime Environment](#) for a list of the supported versions.

2. Install and initialize the Cloud SDK. If you already have the Cloud SDK installed and initialized, run the `gcloud components update` command to update to the latest release.

The Cloud SDK provides you the `gcloud` command-line tooling for deploying and managing your apps.

By downloading, you agree to be bound by the [Terms](#) that govern use of the Cloud SDK for App Engine.

[Install and initialize the Cloud SDK](#)

2. Click Cloud SDK installer, it will automatically download the software (save it in drive)

## Installing Google Cloud SDK

[Send feedback](#)

This page contains instructions for choosing and maintaining a Cloud SDK installation.

### Installation instructions

★ **Note:** If you are behind a proxy/firewall, see the [proxy settings](#) page for more information on installation.

Linux

Debian/Ubuntu

Red Hat/Fedora/CentOS

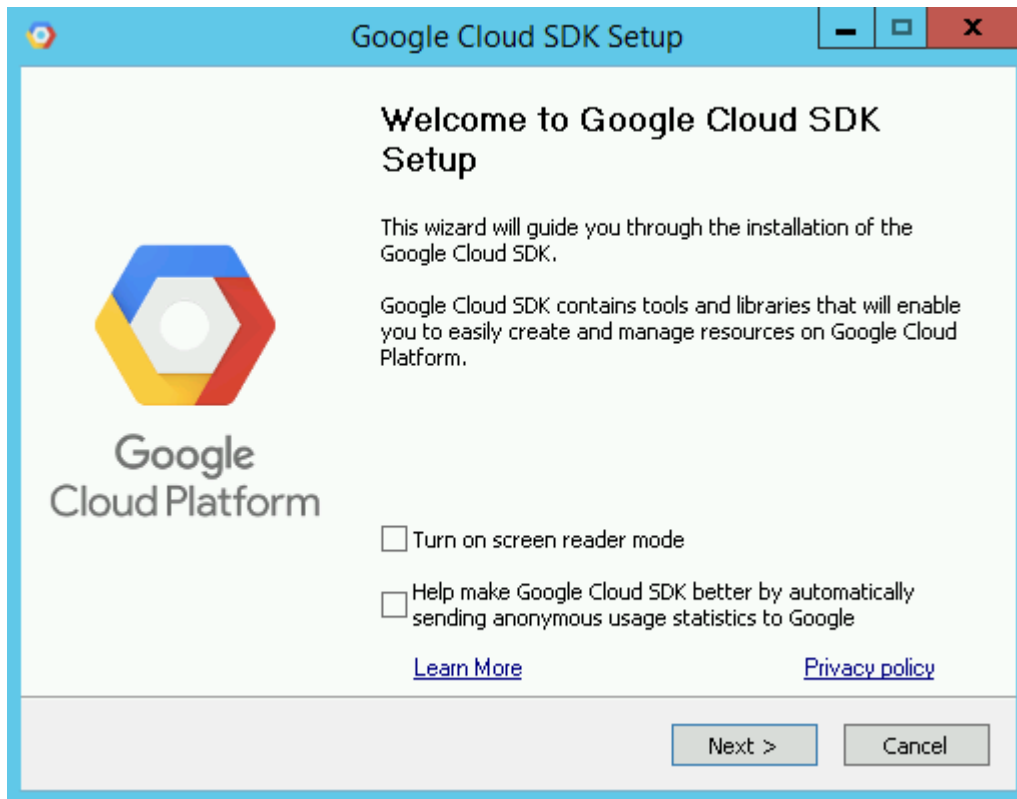
macOS

Windows

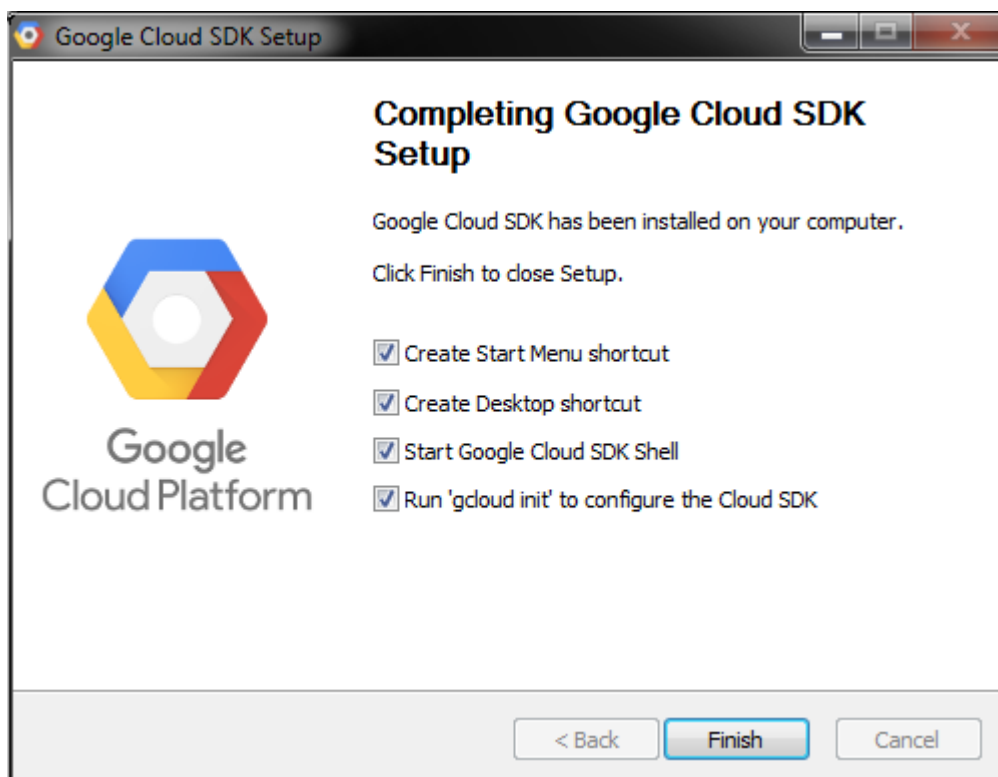
1. Download the [Cloud SDK installer](#).

Alternatively, open a PowerShell terminal and run the following PowerShell commands.

3. Launch the installer and follow the prompts



4. Cloud SDK requires Python; supported versions are Python 3 (preferred, 3.5 to 3.8) and Python 2 (2.7.9 or higher)
5. After installation has completed, the installer presents several options:



6. Open Google Cloud SDK shell and run the program by using the following command

```
C:\Users\User\AppData\Local\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py  
"D:\Dhanamathi\CC Lab\hello"
```

(Note: for the first time, it may ask some necessary components to update, just click Y to update)

### **PROGRAM:**

**//app.yaml**

```
application: test  
version: 1  
runtime: python  
api_version: 1
```

handlers:

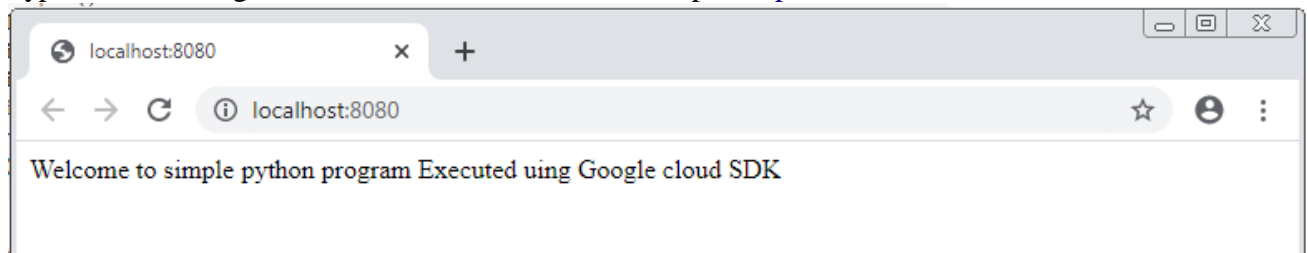
```
- url: /*  
  script: index.py
```

**//index.py**

```
print 'Content-Type: text/plain'  
print ''  
print 'Welcome to simple python program'  
print 'Executed using Google cloud SDK'
```

### **OUTPUT:**

Type the following URL in the browser to see the output <http://localhost:8080/>



### **RESULT:**

Thus the installation of Google App Engine and Creation of hello world web applications using python was done successfully

EX.NO: 4  
DATE:

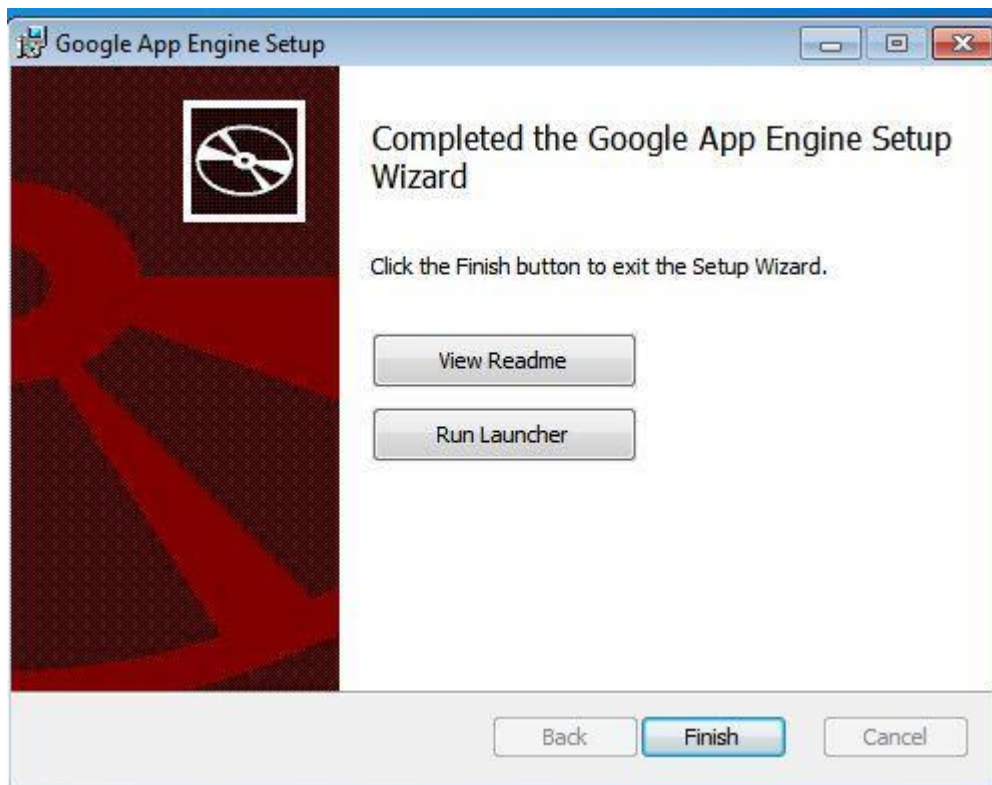
## The web applications launching using GAE launcher

### AIM:

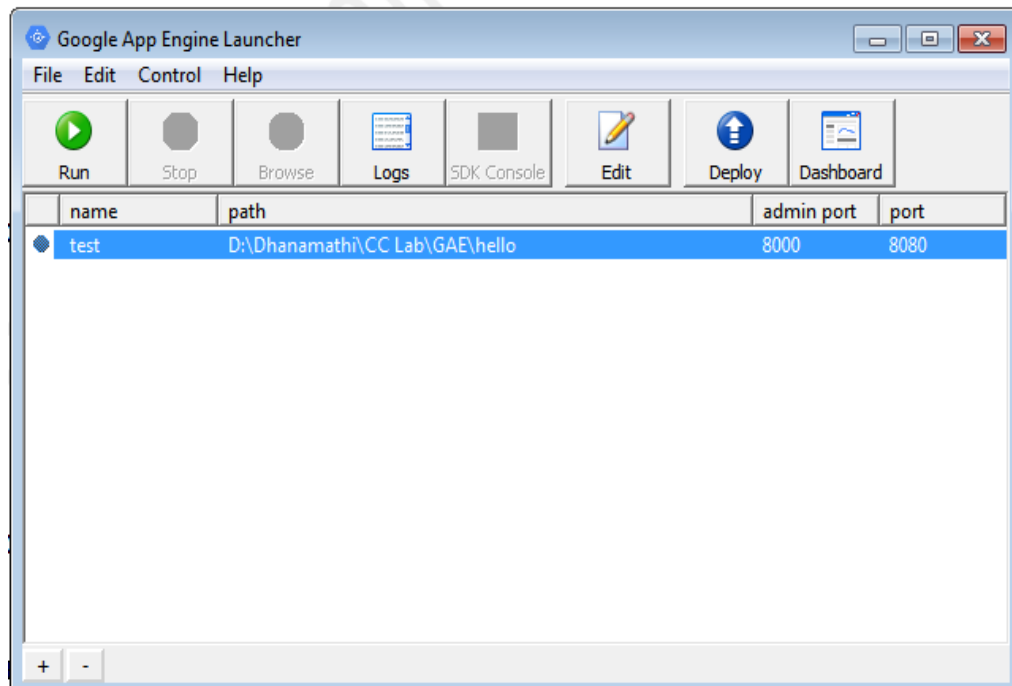
Use GAE launcher to launch the web applications

### PROCEDURE:

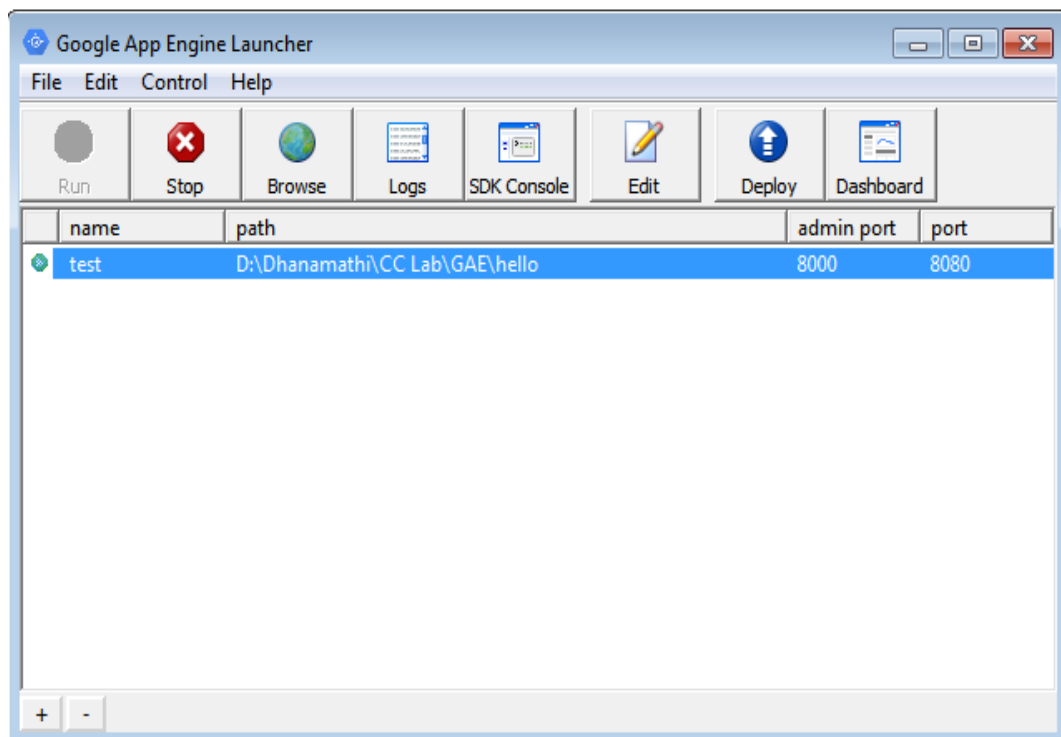
1. Download Google App Engine from <https://www.npackd.org/p/com.google.AppEnginePythonSDK/1.9.52>  
(note: currently not supported by Google cloud platform)
2. Download and install python 2.7.1
3. Install GAE, accept the subsequent steps then click finish



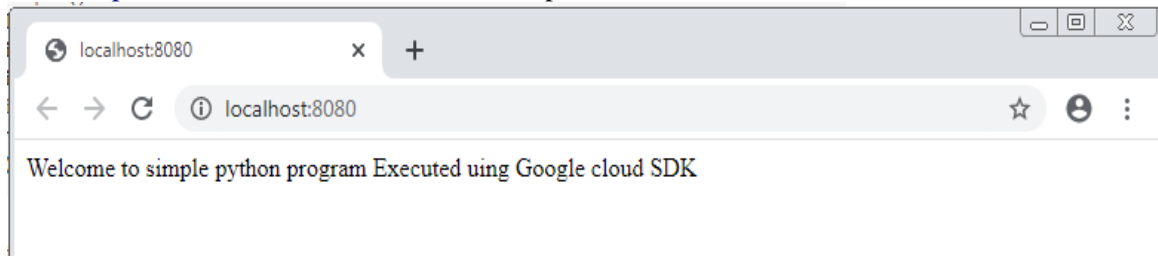
- Open GoogleAppEngine Launcher, in the Launcher window, Click File->Add Existing Application, then browse the folder where the python and yaml program present to set application path, then click add
- Now the launcher looks like below



- Select the application and click Run button, after successful execution of python application, click browse



7. After clicking the browse icon, the launcher open browser with the URL <http://localhost:8080/> and show the output



### **PROGRAM:**

#### **//app.yaml**

```
application: test
version: 1
runtime: python
api_version: 1
```

#### handlers:

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

#### **//index.py**

```
print 'Content-Type: text/plain'
print ''
print 'Welcome to simple python program'
print 'Executed using Google cloud SDK'
```

### **RESULT:**

Thus the python web applications launching using GAE launcher was done successfully



Entities started.

0.0: Broker: Cloud Resource List received with 1 resource(s)

0.0: Broker: Trying to Create VM #0 in Datacenter\_0

0.1: Broker: VM #0 has been created in Datacenter #2, Host #0

0.1: Broker: Sending cloudlet 0 to VM #0

400.1: Broker: Cloudlet 0 received

400.1: Broker: All Cloudlets executed. Finishing...

400.1: Broker: Destroying VM #0

Broker is shutting down...

Simulation: No more future events

CloudInformationService: Notify all CloudSim entities for shutting down.

Datacenter\_0 is shutting down...

Broker is shutting down...

Simulation completed.

Simulation completed.

===== OUTPUT =====

Cloudlet ID	STATUS	Data center ID	VM ID	Time Start	Time Finish
-------------	--------	----------------	-------	------------	-------------

0	SUCCESS	2	0	400.0.1	400.1
---	---------	---	---	---------	-------

\*\*\*\*\*Datacenter: Datacenter\_0\*\*\*\*\*

User id Debt

3 35.6

CloudSimExample1 finished!

## **RESULT:**

Thus the Simulation of a cloud scenario using CloudSim and run a scheduling algorithm in CloudSim was done successfully

**EX.NO: 6**  
**DATE:**

## **Procedure to transfer the files from one virtual machine to another virtual machine**

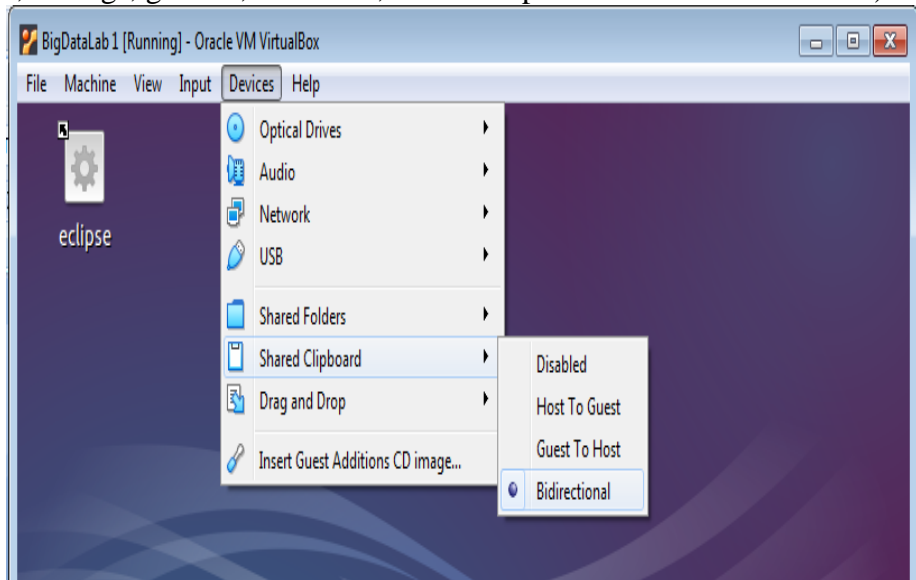
### **AIM:**

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

### **PROCEDURE:**

#### 1. Copy few (or more) lines with **copy & paste** mechanism:

- ↓ Install **Guest Addition** on the virtual machines
- ↓ In VM, go to Devices -> Shared Clipboard and click on Bidirectional ( or Click on the machine, settings, general, advanced, Shared clipboard: set to bidirectional )



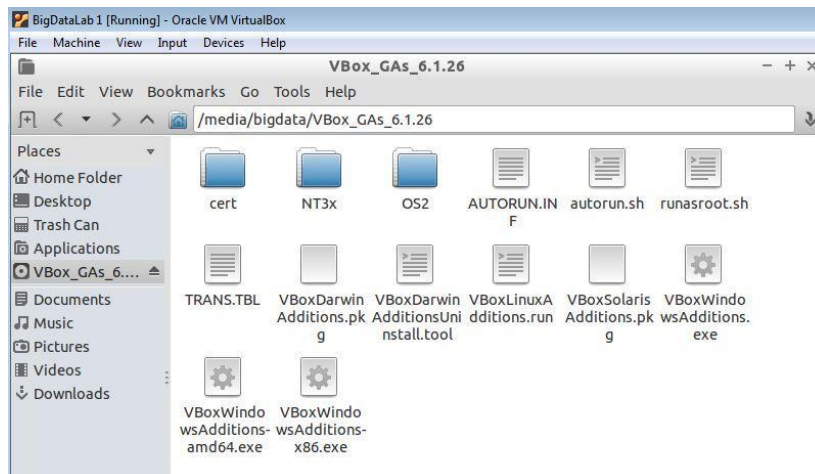
- ↓ Now copy from guest OS in the clipboard that is shared with the host OS then you paste from the host OS to the second guest OS.

#### 2. Enable **drag and drop**:

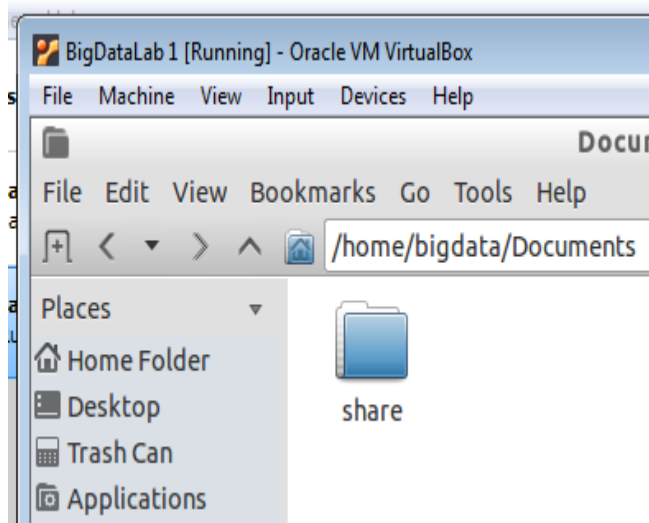
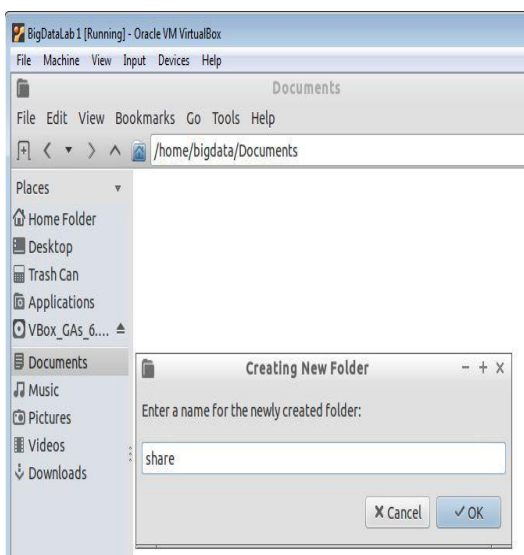
- ↓ Install **Guest Addition** on the virtual machines
- ↓ In VM, go to Devices -> Drag and Drop and click on Bidirectional ( or Click on the machine, settings, general, advanced, drag and drop: set to bidirectional )
- ↓ Now drag and drop files / folders between Guest and Host OS

#### 3. **Common Shared Folders** on virtual machines and use one of the directory shared as buffer to copy.

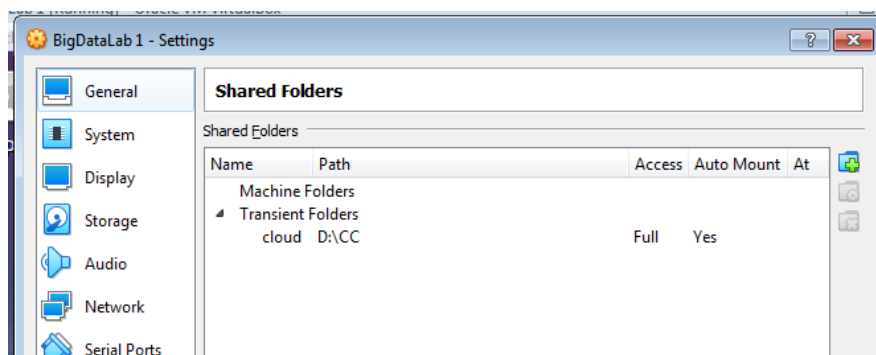
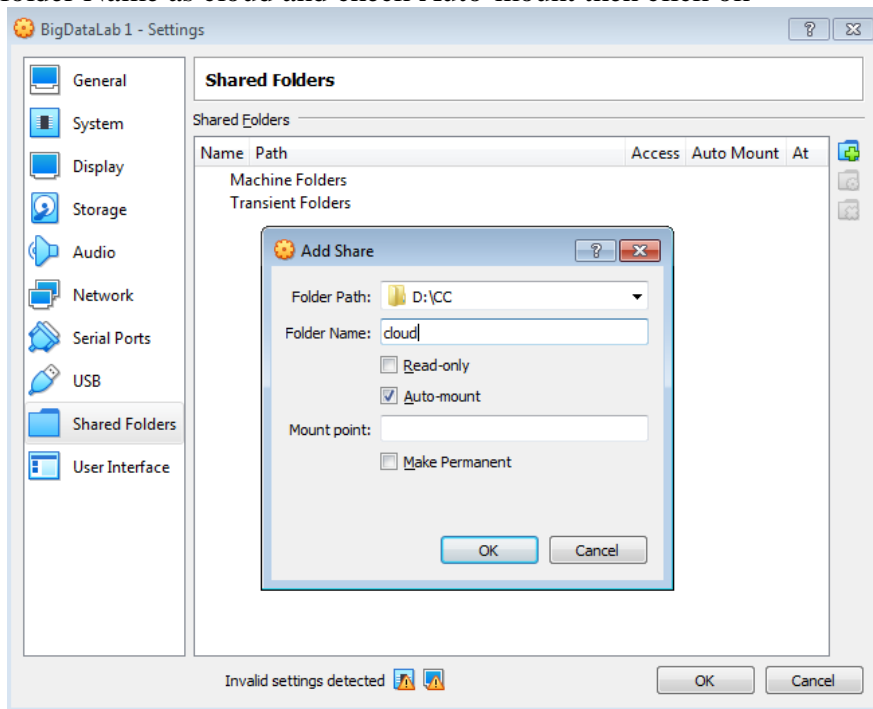
- ↓ Install **Guest Additions** have the possibility to set Shared Folders too.



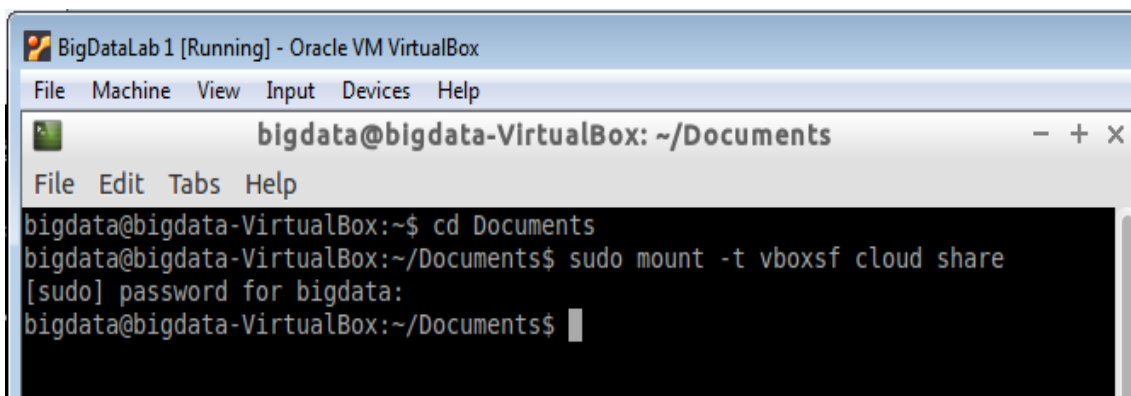
- ❑ Create new folder in Guest OS (ubuntu) with the name share inside home/bigdata/Documents



- ❑ To share folder from Host OS ( Windows 7), Click on the Machine -> Settings -> Shared Folders , in the right side click on + icon , an Add share window pops up
- ❑ In Add share window, Folder Path, click others and choose the folder to share (here D:\CC), give the folder Name as cloud and check Auto-mount then click ok

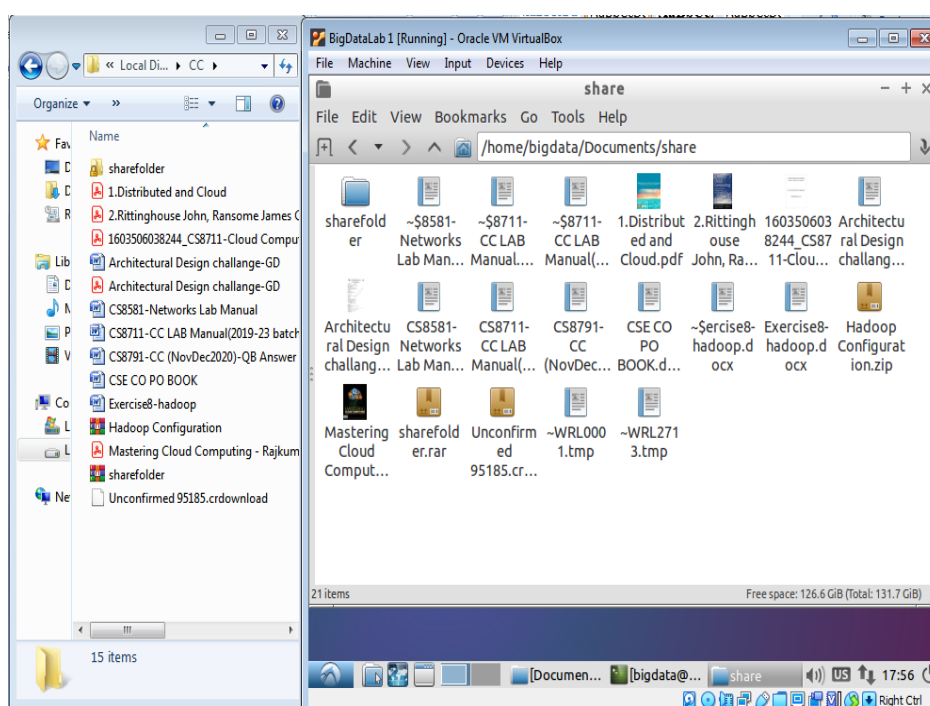


- Open terminal (Start->Accessories->LXTerminal) and type the following commands to enable shared folder between Guest and Host OS (ask password for the first time)



```
BigDataLab 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
bigdata@bigdata-VirtualBox: ~/Documents
File Edit Tabs Help
bigdata@bigdata-VirtualBox:~$ cd Documents
bigdata@bigdata-VirtualBox:~/Documents$ sudo mount -t vboxsf cloud share
[sudo] password for bigdata:
bigdata@bigdata-VirtualBox:~/Documents$
```

- The contents in both the folders (Left Host OS(Windows7) , Right Guest OS (ubuntu)) are now shared, updates in one folder reflects on other



4. 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)

Need an active server (sshd) on the receiving machine and a client on the sending machine, with authorization settled (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. Install with `sudo apt-get install openssh-server`.

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

- 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.

## RESULT:

Thus the various methods to transfer the files from one virtual machine to another virtual machine was done successfully

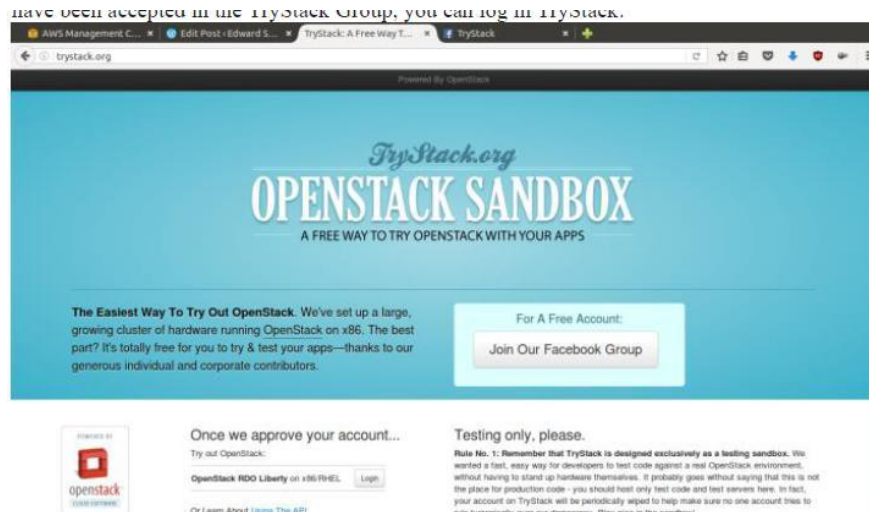
EX.NO: 7  
DATE:

## Procedure to launch virtual machine using trystack (Online Openstack Demo Version)

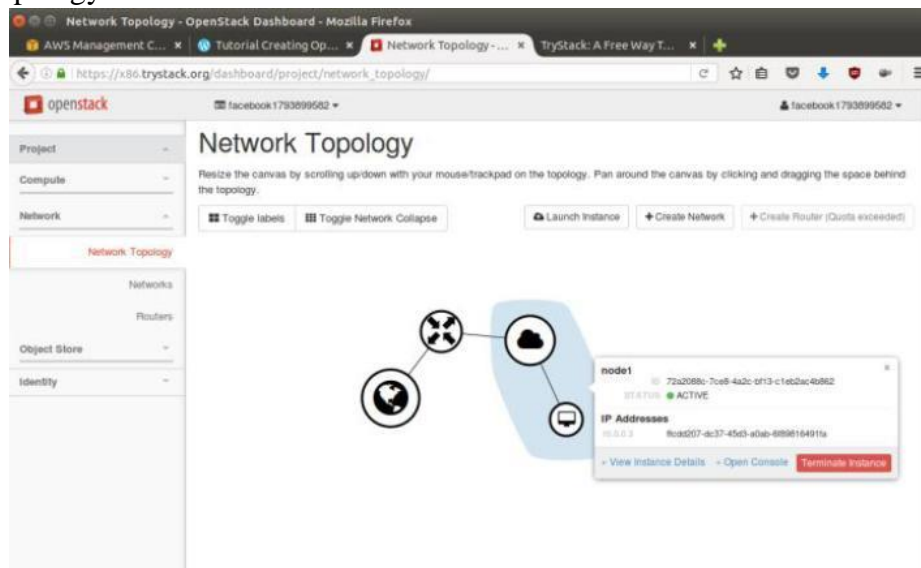
### AIM:

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

### PROCEDURE:



### Network topology



### Step 1: Create Network

1. Go to **Network > Networks** and then click **Create Network**.
2. In **Network** tab, fill **Network Name** for example internal and then click **Next**.
3. In **Subnet** tab,
  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:** 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,

↓ Fill **Key Pair Name** with your machine name (for example Edward-Key).

↓ Fill **Public Key** with your **SSH public key** (usually is in `~/.ssh/id_rsa.pub`). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate key pair.

↓ Click **Import key pair**.

3. In **Security Groups**, mark/check **default**.

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

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

### Step 3: Create Router

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

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

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

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

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

1. Select **External networks** with **external**.

2. Then **OK**.

5. Click **Add Interface** button.

1. Select **Subnet** with the network that you have been created in Step 1.

2. Click **Add interface**.

6. Go to **Network > Network Topology**. You will see the network topology. In the example, there are two networks, 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. OpenStack, the public IPs is collected in a pool and managed by admin (in our case is TryStack).

To request a public (floating) IP address to be assigned to your instance.

1. Go to **Compute > Instance**.

2. In one of your instances, click **More > Associate Floating IP**.

3. In **IP Address**, click Plus [+].

4. Select **Pool** to **external** and then click **Allocate IP**.

5. Click **Associate**.

6. Now you will get a public IP, e.g. 8.21.28.120, for your instance.

**Step 5: Configure Access & Security:** OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called *Security Group*.

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

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

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

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

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

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

**Step 6: SSH to Your Instance:** SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be ubuntu.

### RESULT:

Thus the procedure to launch virtual machine using trystack (Online Openstack Demo Version) was done successfully

EX.NO: 8  
DATE:

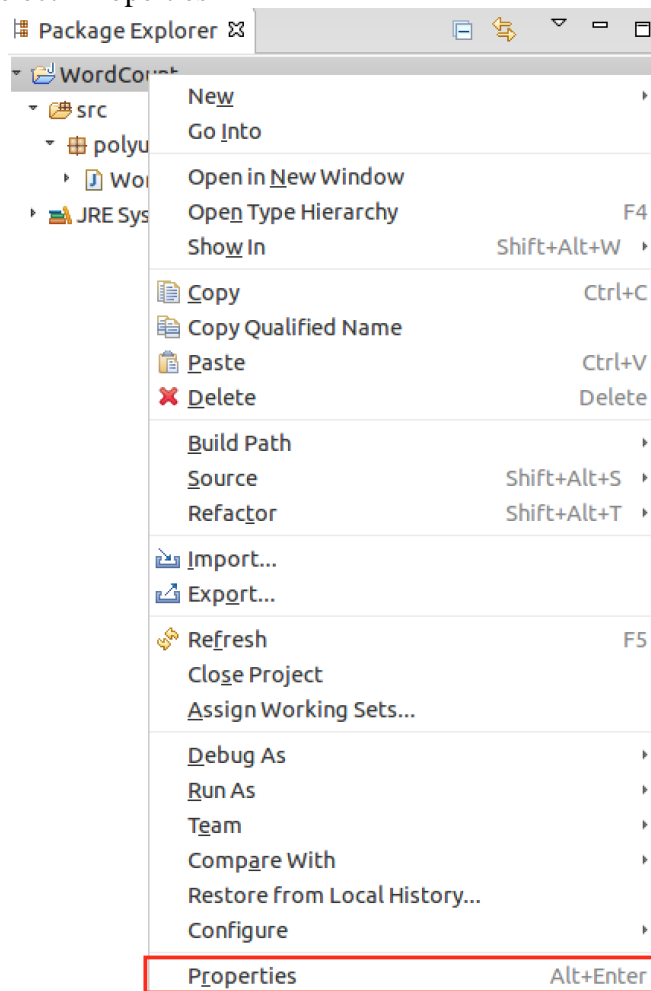
## Installation of Hadoop single node cluster and run simple word count applications

### AIM:

Install Hadoop single node cluster and run simple applications like word count

### PROCEDURE:

1. Open eclipse and create a java project by clicking **File -> New -> Java Project** in the Eclipse window, and assigning project name (i.e. WordCount) in the dialogue.
2. Create a package called **polyu.bigdata** under the **src** folder, by clicking File -> New -> Package and assigning name as polyu.bigdata in the dialogue
3. Create a class called WordCount, by clicking File -> New -> Class and assigning name as WordCount in the dialogue. And then type WordCount.java and save it.
4. Import the Hadoop Java library to resolve the dependency problem by Right clicking on WordCount project , select "Properties"

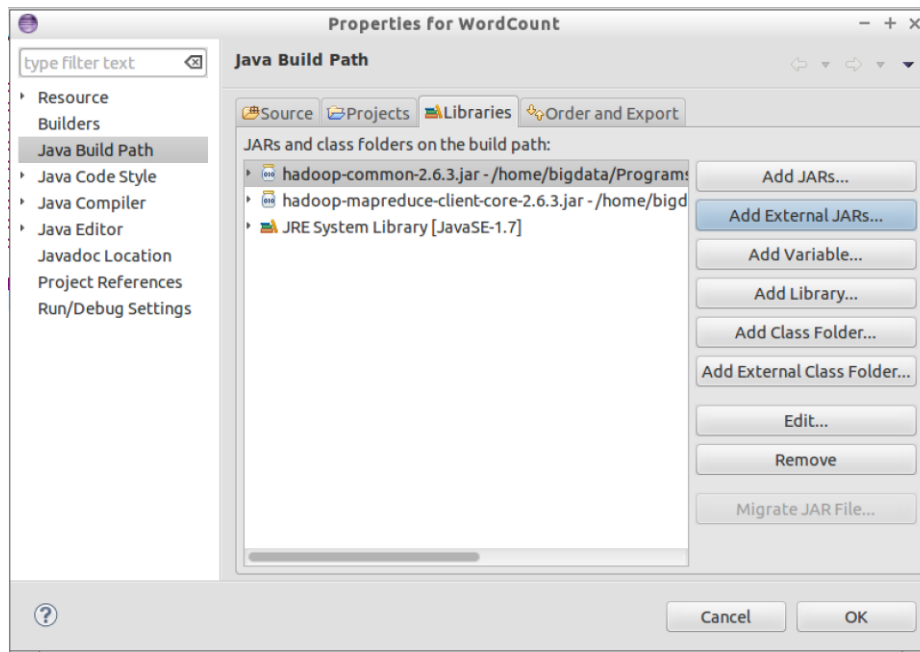


5. Select **Java Build Path->Configure Build Path->Libraries->Add External JARs** import jars "hadoop-common-2.6.3.jar" and "hadoop-mapreduce-client-core-2.6.3.jar" which located at :

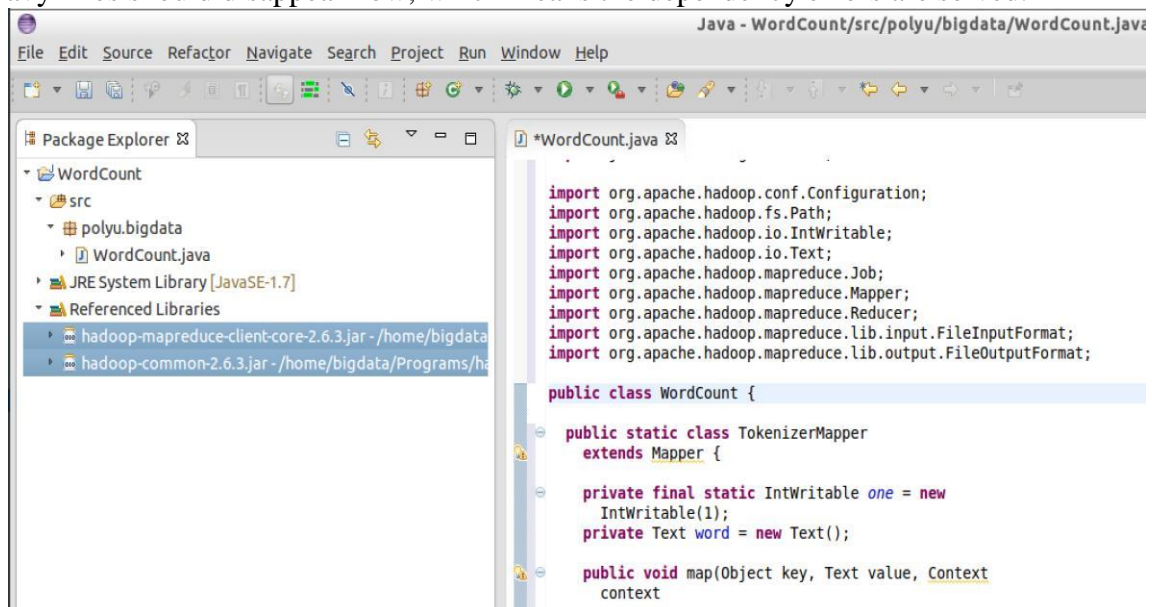
/home/bigdata/Programs/hadoop/share/hadoop/common/hadoop-common-2.6.3.jar

/home/bigdata/Programs/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.6.3.jar

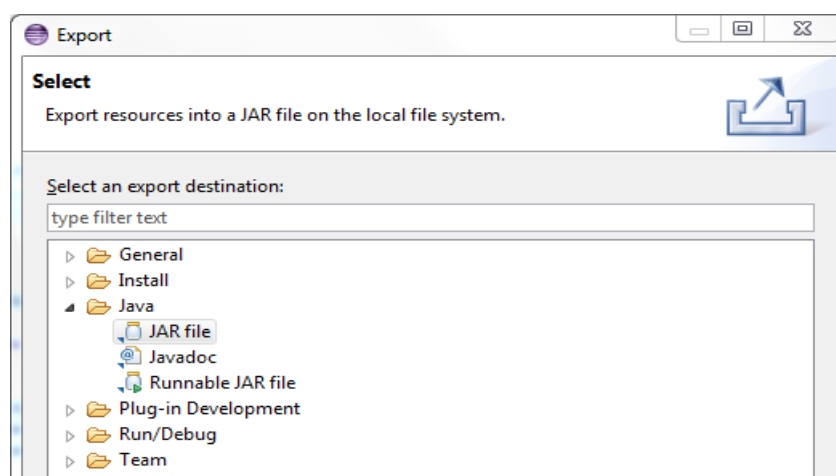
6. After both jars are imported, "Java Build Path" should look like:



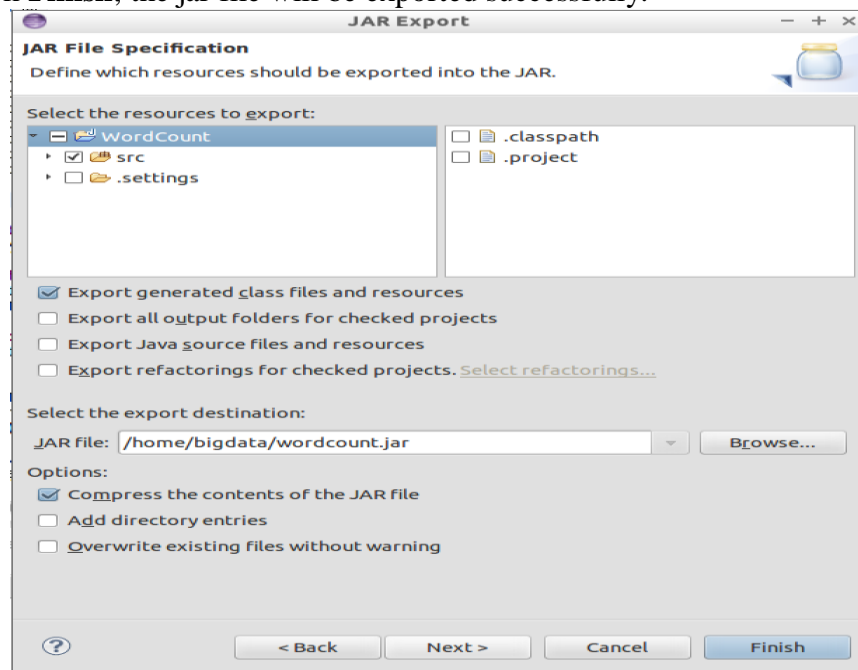
Red wavy lines should disappear now, which means the dependency errors are solved.



7. Next export the source files as a jar file and upload it to Hadoop system to run it. Right click WordCount project, then click **Export**, it is like this:



8. Select JAR file, click **Next**, Select the export destination to **/home/bigdata/wordcount.jar**. Then click **Finish**, the jar file will be exported successfully.



9. To Start Hadoop, open LXTerminal in applications/Accessories, type the following commands  
~/Programs/hadoop/sbin/start-all.sh
10. Check Hadoop status, open the browser, type the following urls, and view the running status of Hadoop.  
http://localhost:8088 and http://localhost:50070

### Upload data to Hadoop File System(HDFS)

11. The Hadoop must be started first, then type the following commands in terminal.

#### Create folder for data on HDFS

```
~/Programs/hadoop/bin/hadoop fs -mkdir -p /user/bigdata/wordcount/input
```

#### Upload data to HDFS

```
~/Programs/hadoop/bin/hadoop fs -put ~/hadoop.txt /user/bigdata/wordcount/input
```

### PROGRAM:

```
// WordCount.java
```

```
package polyu.bigdata; import
java.io.IOException; import
java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {
    //Mapper which implement the mapper() function
    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>
    {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
```

```

public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
    }
}
}
//Reducer which implement the reduce() function
public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        } result.set(sum);
        context.write(key, result);
    }
}
//Driver class to specific the Mapper and Reducer
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

```

### //hadoop.txt

Youth is not a time of life; it is a state of mind; it is not a matter of rosy cheeks, red lips and supple knees; it is a matter of the will, a quality of the imagination, a vigor of the emotions; it is the freshness of the deep springs of life.

Youth means a temperamental predominance of courage over timidity of the appetite, for adventure over the love of ease. This often exists in a man of sixty more than a body of twenty. Nobody grows old merely by a number of years. We grow old by deserting our ideals.

Years may wrinkle the skin, but to give up enthusiasm wrinkles the soul. Worry, fear, self-distrust bows the heart and turns the spirit back to dust.

Whether sixty or sixteen, there is in every human being's heart the lure of wonder, the unfailling child-like appetite of what's next, and the joy of the game of living. In the center of your heart and my heart there is a wireless station; so long as it receives messages of beauty, hope, cheer, courage and power from men and from the Infinite, so long are you young.

When the aerials are down, and your spirit is covered with snows of cynicism and the ice of pessimism, then you are grown old, even at twenty, but as long as your aerials are up, to catch the waves of optimism, there is hope you may die young at eighty.

## OUTPUT:

### Run program

```
~/Programs/hadoop/bin/hadoop jar ~/wordcount.jar polyu.bigdata.WordCount
```

```
/user/bigdata/wordcount/input /user/bigdata/wordcount/output
```

If it runs successfully, the following information will pops up:

```
bigdata@bigdata-VirtualBox:~$ ~/Programs/hadoop/bin/hadoop jar ~/wordcount.jar /user/bigdata/wordcount/input /user/bigdata/wordcount/output
16/01/21 14:19:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/01/21 14:19:54 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/01/21 14:19:54 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/01/21 14:19:55 INFO input.FileInputFormat: Total input paths to process : 1
16/01/21 14:19:55 INFO mapreduce.JobSubmitter: number of splits:1
16/01/21 14:19:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1453354809925_0001
16/01/21 14:19:55 INFO impl.YarnClientImpl: Submitted application application_1453354809925_0001
16/01/21 14:19:55 INFO mapreduce.Job: The url to track the job: http://bigdata-VirtualBox:8088/proxy/application_1453354809925_0001/
16/01/21 14:19:55 INFO mapreduce.Job: Running job: job_1453354809925_0001
16/01/21 14:20:00 INFO mapreduce.Job: Job job_1453354809925_0001 running in uber mode : false
16/01/21 14:20:00 INFO mapreduce.Job: map 0% reduce 0%
```

To View execution status, **open browser and access URL below.**

<http://localhost:8088>

App ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Track UI
application_1453354809925_0002	bigdata	word count	MAPREDUCE	default	Thu, 21 Jan 2016 06:34:03 GMT	Thu, 21 Jan 2016 06:34:14 GMT	FINISHED	SUCCEEDED		Histor
application_1453354809925_0001	bigdata	word count	MAPREDUCE	default	Thu, 21 Jan 2016 06:19:55 GMT	Thu, 21 Jan 2016 06:20:12 GMT	FINISHED	FAILED		Histor

To Check output, first **download the output folder from HDFS by using the command below.**

```
~/Programs/hadoop/bin/hadoop fs -get /user/bigdata/wordcount/output ~/
```

The output folder now is in the home folder of the guest system, use text editor to open the result file, the result file name is **part-r-00000**.

The output is like this:

```
In 1
Infinite, 1
Nobody 1
This 1
We 1
When 1
Whether 1
Worry, 1
Years 1
Youth 2
a 11
adventure 1
aerials 2
and 8
```

## RESULT:

Thus the installation of Hadoop single node cluster and execution of java wordcount applications was done successfully