Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY
University with Graded Autonomy Status
(An ISO 21001 : 2018 Certified Institution)
Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.

# RECORD NOTEBOOK

## CLOUD COMPUTING LAB — (EBCS22L10)

## 2025-2026(ODD SEMESTER)

## DEPARTMENT

## Of

## ARTIFICIAL INTELLIGENCE

| | | |
|---|---|---|
| **NAME** | **:** | **NUNE SOMA SEKHARA GUPTA** |
| **REGISTER NO** | **:** | **221211101107** |
| **COURSE** | **:** | **B.TECH - AI** |
| **YEAR/SEM/SEC** | **:** | **IV / VII / B** |

Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY
University with Graded Autonomy Status
(An ISO 21001 : 2018 Certified Institution)
Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.

## BONAFIDE CERTIFICATE

**REGISTER NO : 221211101107**

**NAME OF THE LAB : CLOUD COMPUTING LAB**

**DEPARTMENT : ARTIFICIAL INTELLIGENCE**

Certified that this is the bonafide record of work done by **NUNE SOMA SEKHARA GUPTA 221211101107** of IV Year B.Tech(AI), Sec-'B' in the **CLOUD COMPUTING LAB** during the year 2025-2026.

**Signature of Lab-in-Charge** **Signature of Head of Dept**

Submitted for the Practical Examination held on ----------

**Internal Examiner** **External Examiner**

# INDEX


| S.NO | DATE | NAME OF THE PROGRAM | PAGE.NO | STAFF SIGN |
|---|---|---|---|---|
| **1** | **09/07/2025** | Install Virtualbox / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8 | **1 – 7** | |
| **2** | **16/07/2025** | Install a C compiler in the virtual machine created using virtual box and execute Simple Programs | **8 – 12** | |
| **3** | **23/07/2025** | Install Google App Engine. Create hello world app and other simple web applications using python/java | **13 – 20** | |
| **4** | **30/07/2025** | Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim | **21 – 24** | |
| **5** | **06/08/2025** | Use GAE launcher to launch the web applications | **25 – 29** | |
| **6** | **20/08/2025** | Find a procedure to transfer the files from one virtual machine to another virtual machine | **30 – 36** | |
| **7** | **03/09/2025** | Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version) | **37 – 41** | |
| **8** | **10/09/2025** | Install Hadoop single node cluster and run simple applications like wordcount | **41 – 52** | |

**EXP NO : 1** **DATE: 09/07/2025**

# Install Virtualbox / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

**Aim:**

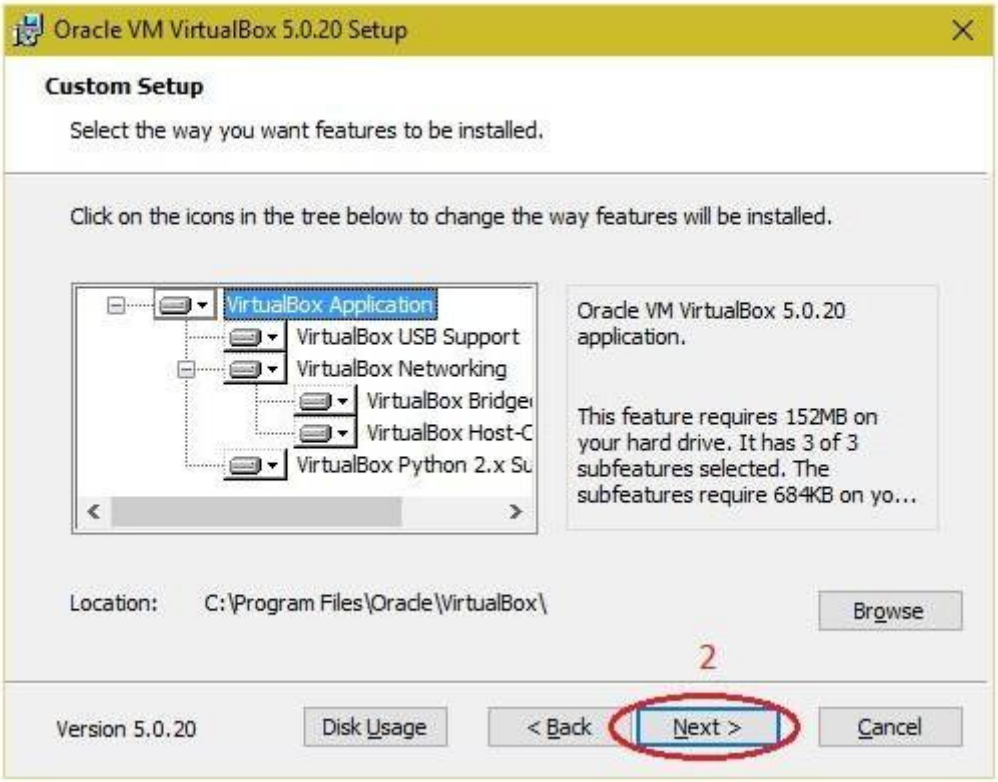
To Install Virtualbox / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

**Procedure:**

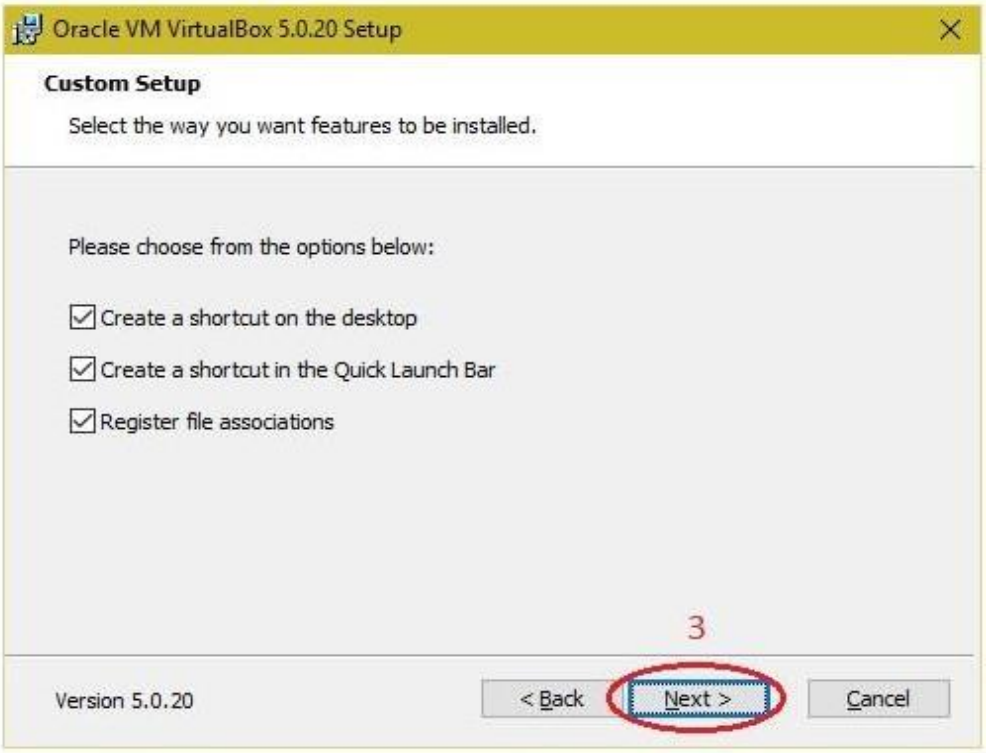
**Steps to install Virtual Box:**

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

2. Click the next button..

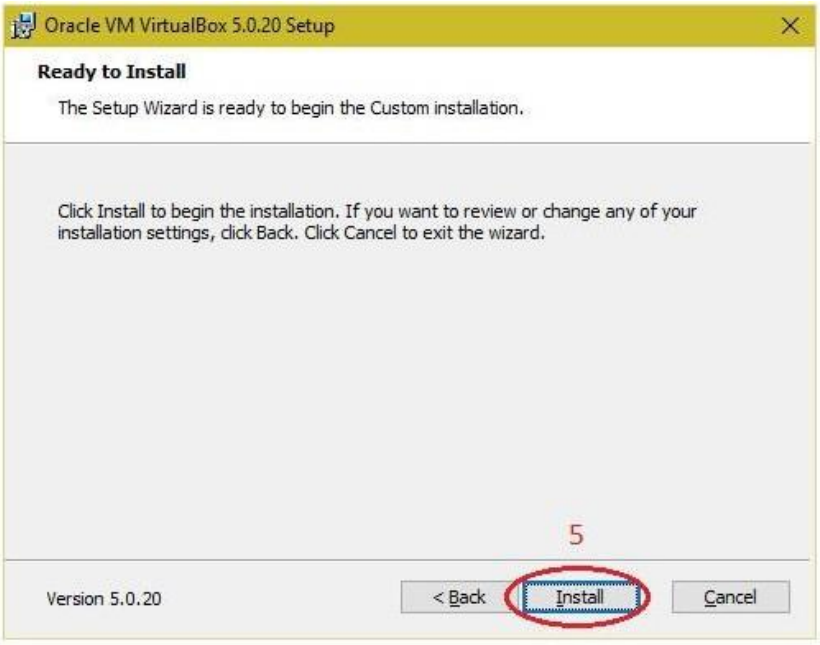



3. Click the next button



4. Click the YES button..

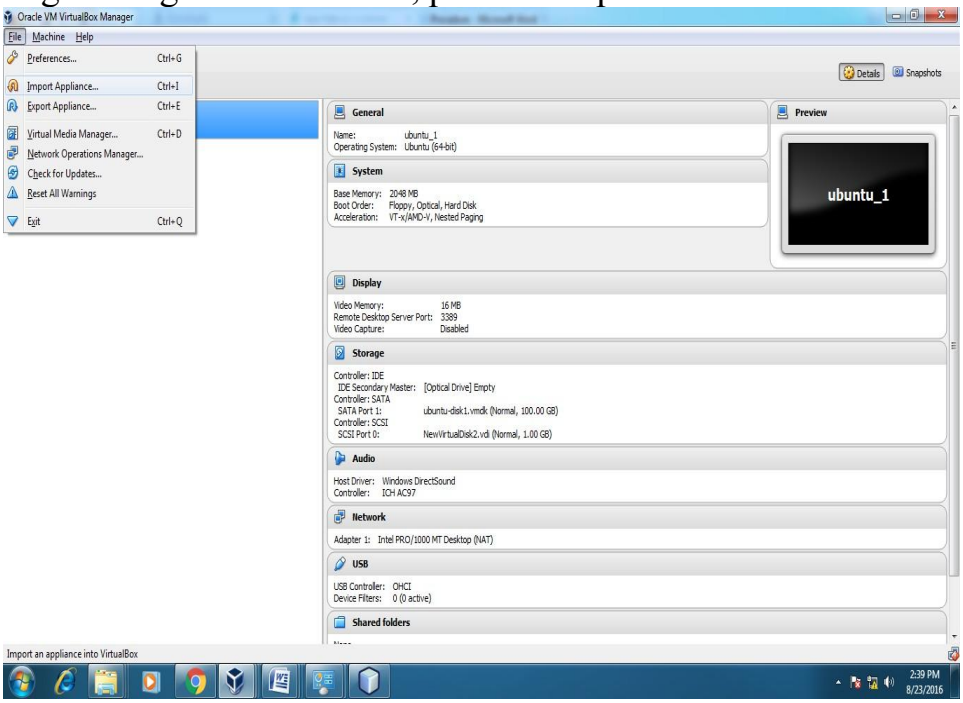
5. Click the install button…

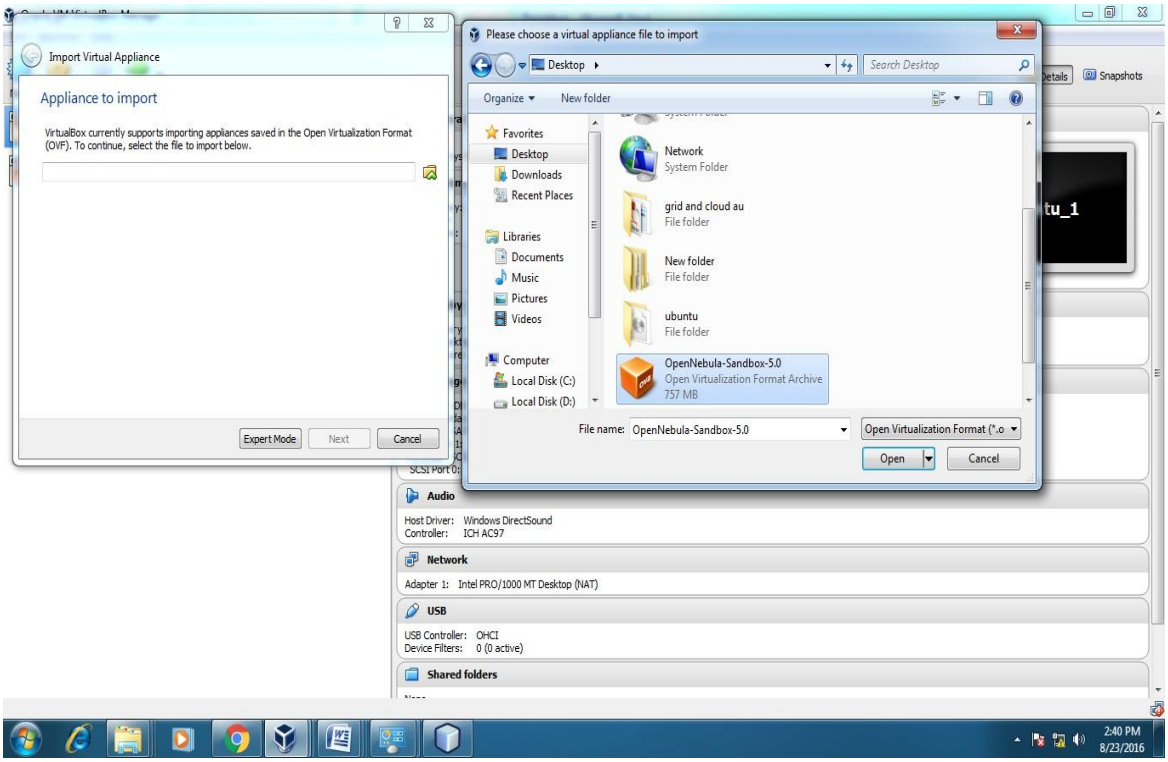


6. Then installation was completed..the show virtual box icon on desktop screen….

**Steps to import Open nebula sandbox:**

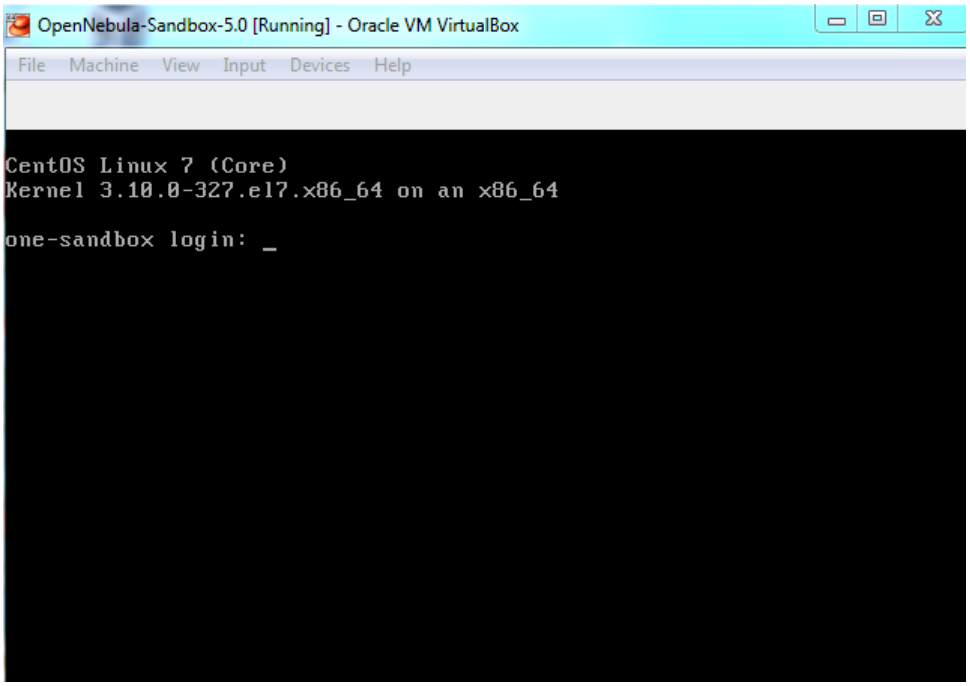
1. Open Virtual box
2. File →import Appliance
3. Browse OpenNebula-Sandbox-5.0.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the Open Nebula
6. Login using username: root, password:opennebula

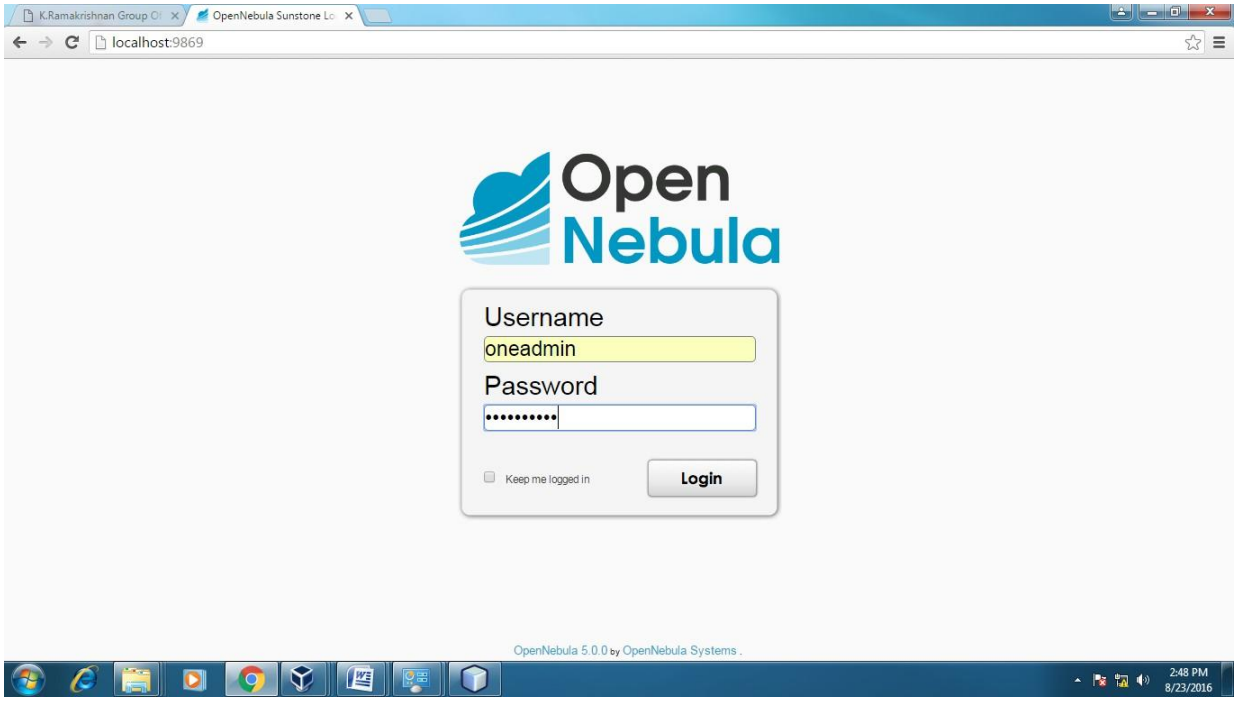


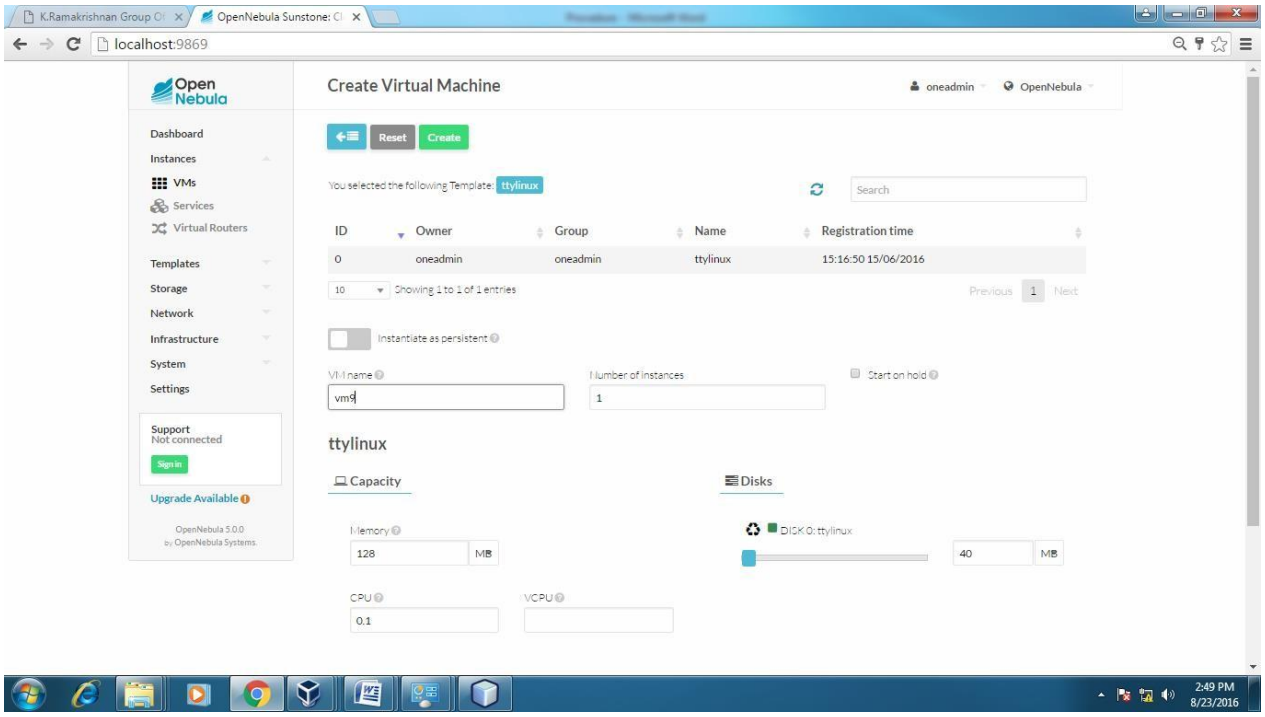



## Steps to create Virtual Machine through opennebula

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.






## Applications:

There are various applications of cloud computing in today's network world. Many search engines and social websites are using the concept of cloud computing like www.amazon.com, hotmail.com, facebook.com, linkedln.com etc. the advantages of cloud computing in context to scalability is like reduced risk , low cost testing ,ability to segment the customer base and auto-scaling based on application load.

**Result:**

Thus the procedure to run the virtual machine of different configuration.

**EXP NO : 2** **DATE: 16/07/2025**

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

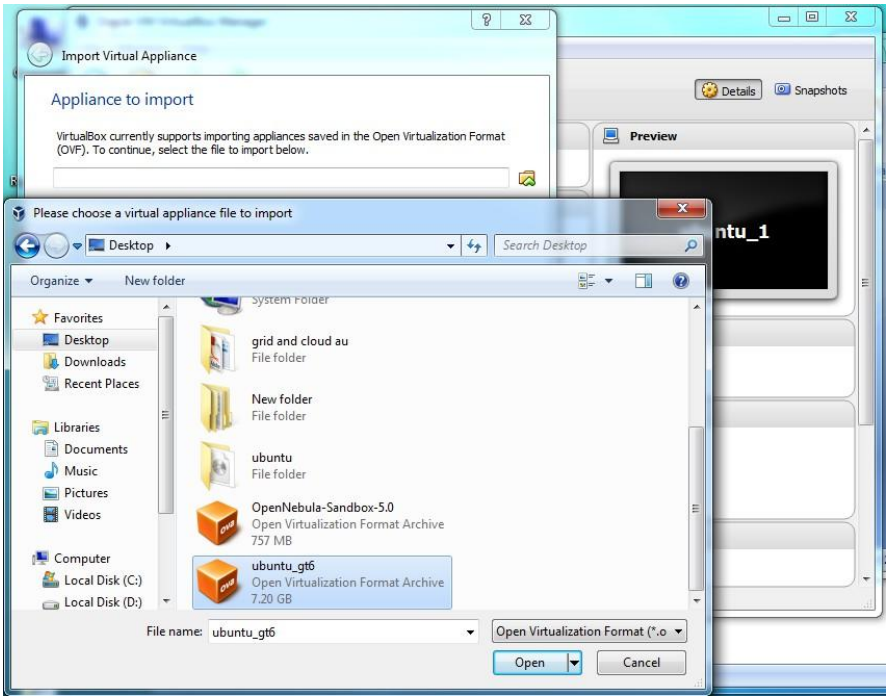
## Aim:

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

## Procedure:

## Steps to import .ova file:

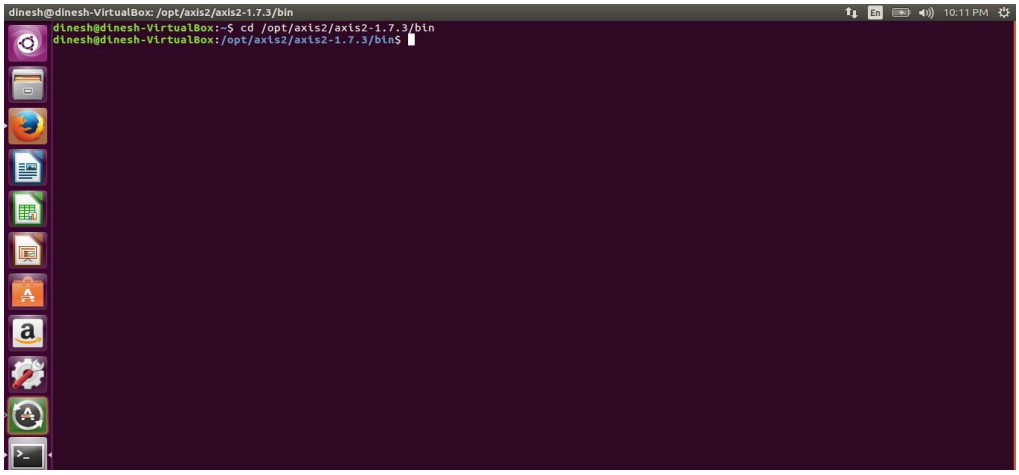
1. Open Virtual box
2. File →import Appliance
3. Browse ubuntu_gt6.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the ubuntu_gt6
6. Login using username: dinesh, password:99425.



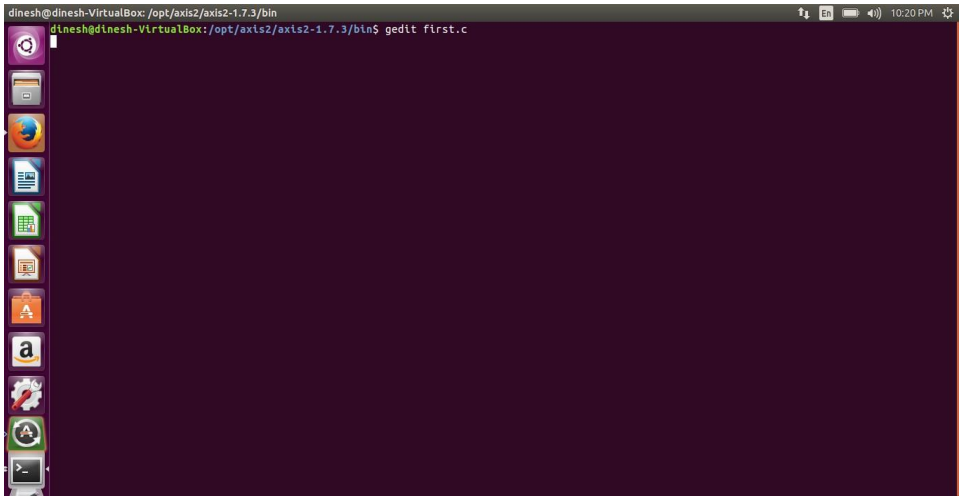
**Steps to run c program:**

1. Open the terminal
2. Type cd /opt/axis2/axis2-1.7.3/bin then press enter
3. gedit hello.c
4. gcc hello.c
5. ./a.out

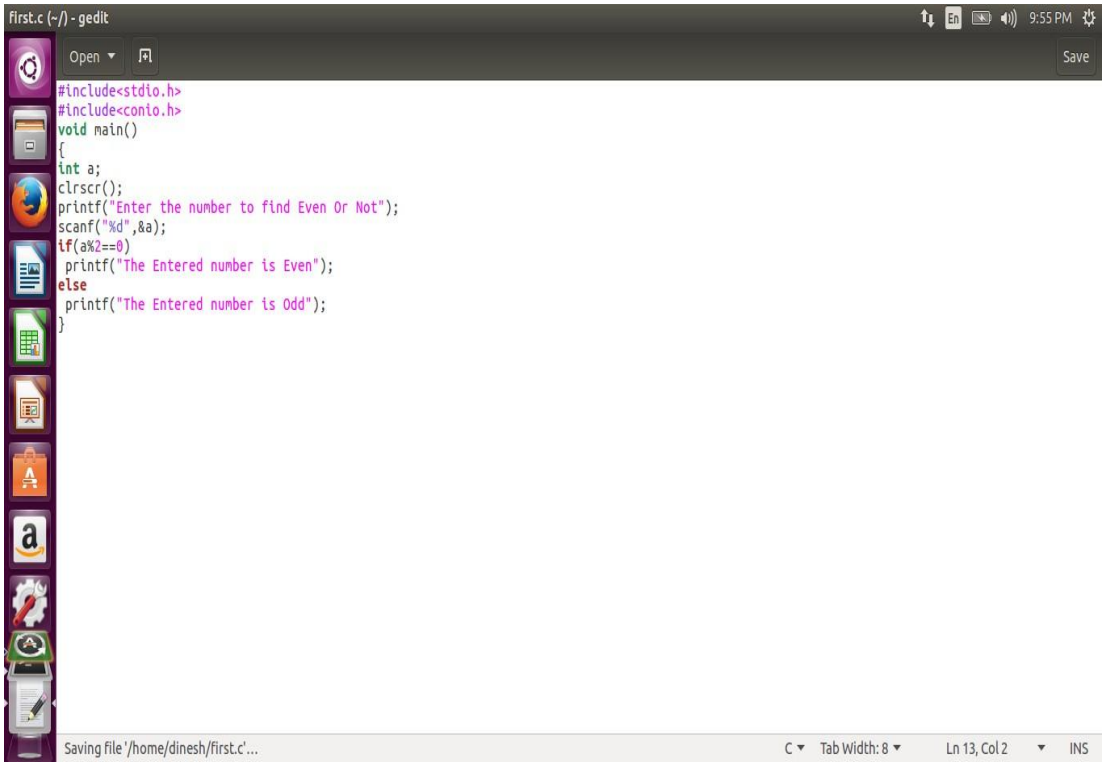
1. **Type cd /opt/axis2/axis2-1.7.3/bin then press enter**



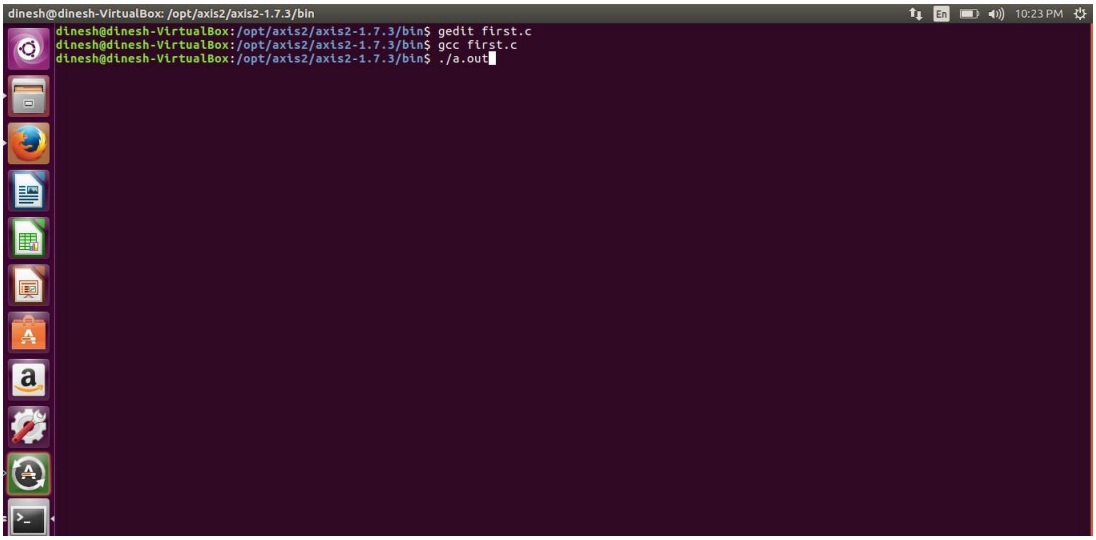

2. **Type gedit first.c**



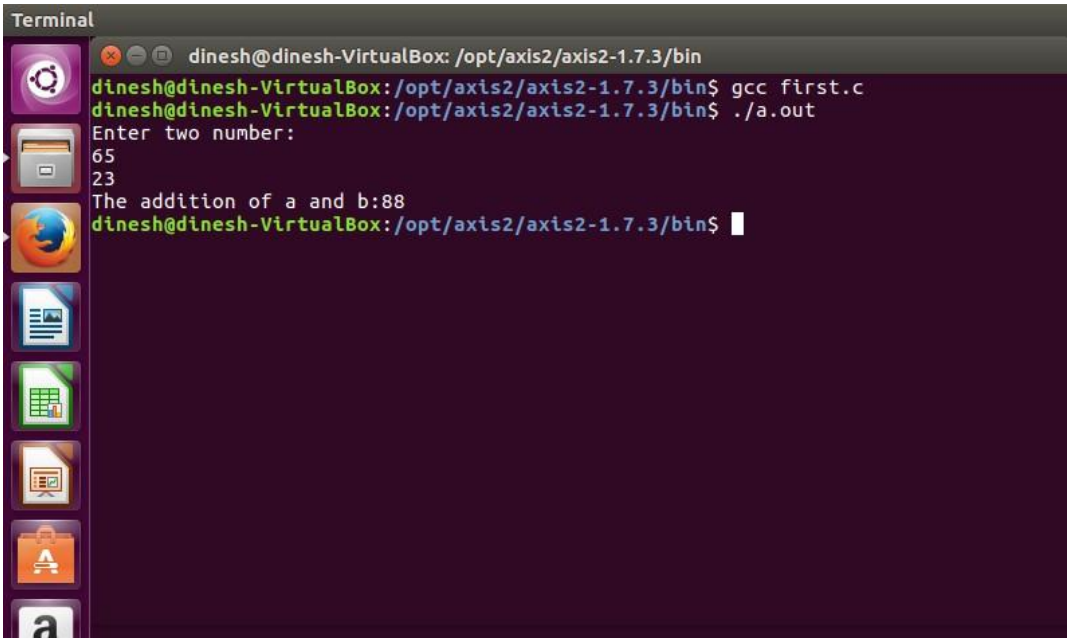
## 3. Type the c program




## 4. Running the C program



5. **Display the output:**




**Applications:**

Simply running all programs in grid environment.

**Result:**

Thus the simple C programs executed successfully.

**EXP NO : 3** **DATE: 23/07/2025**

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

**Aim:**

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

**Procedure:**

1. *Install Google Plugin for Eclipse*

   Read this guide – how to install Google Plugin for Eclipse. If you install the Google App Engine Java SDK together with "**Google Plugin for Eclipse**", then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.

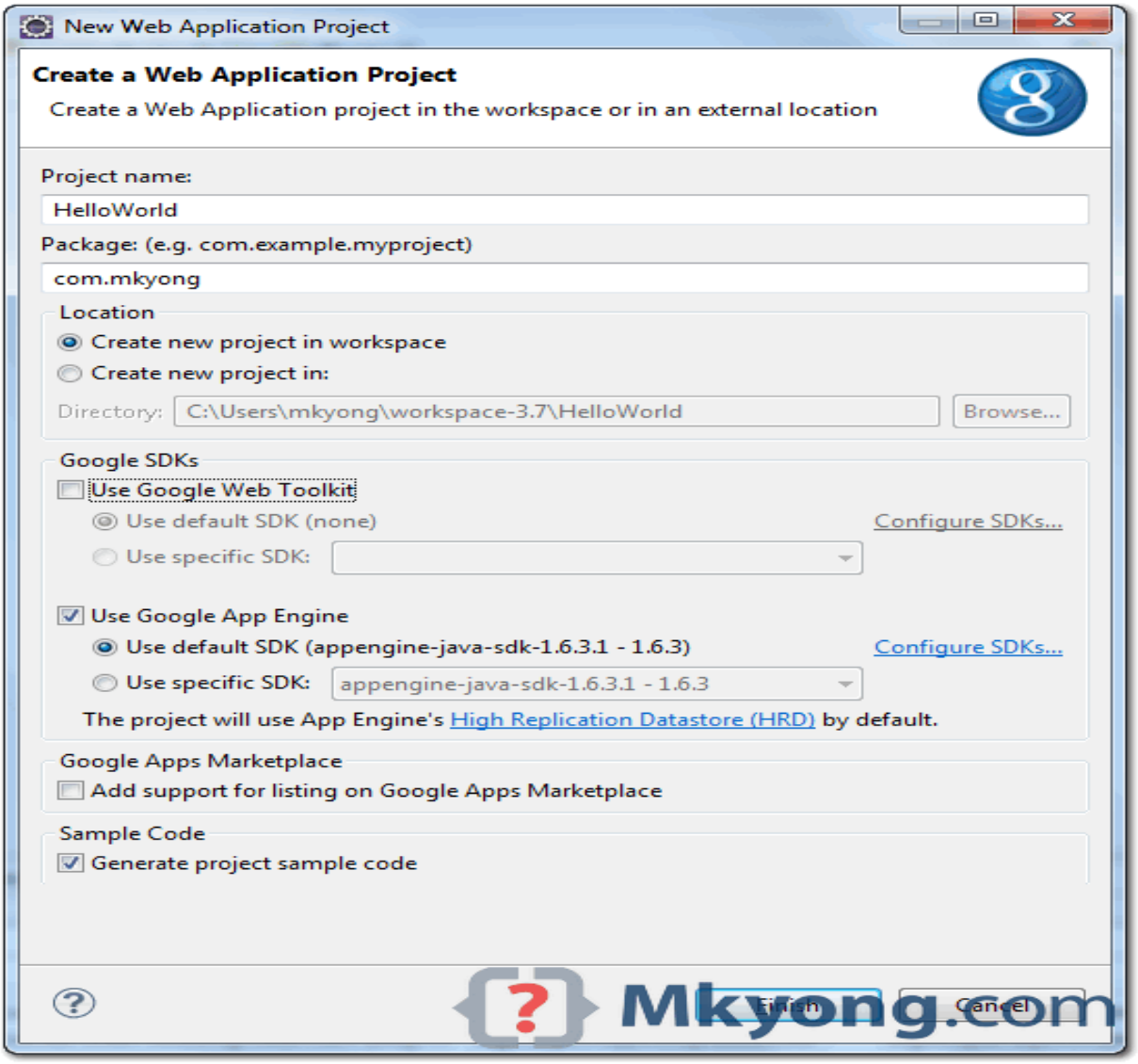
2. *Create New Web Application Project*

   In Eclipse toolbar, click on the Google icon, and select "**New Web Application Project…**"

   *Figure – New Web Application Project*

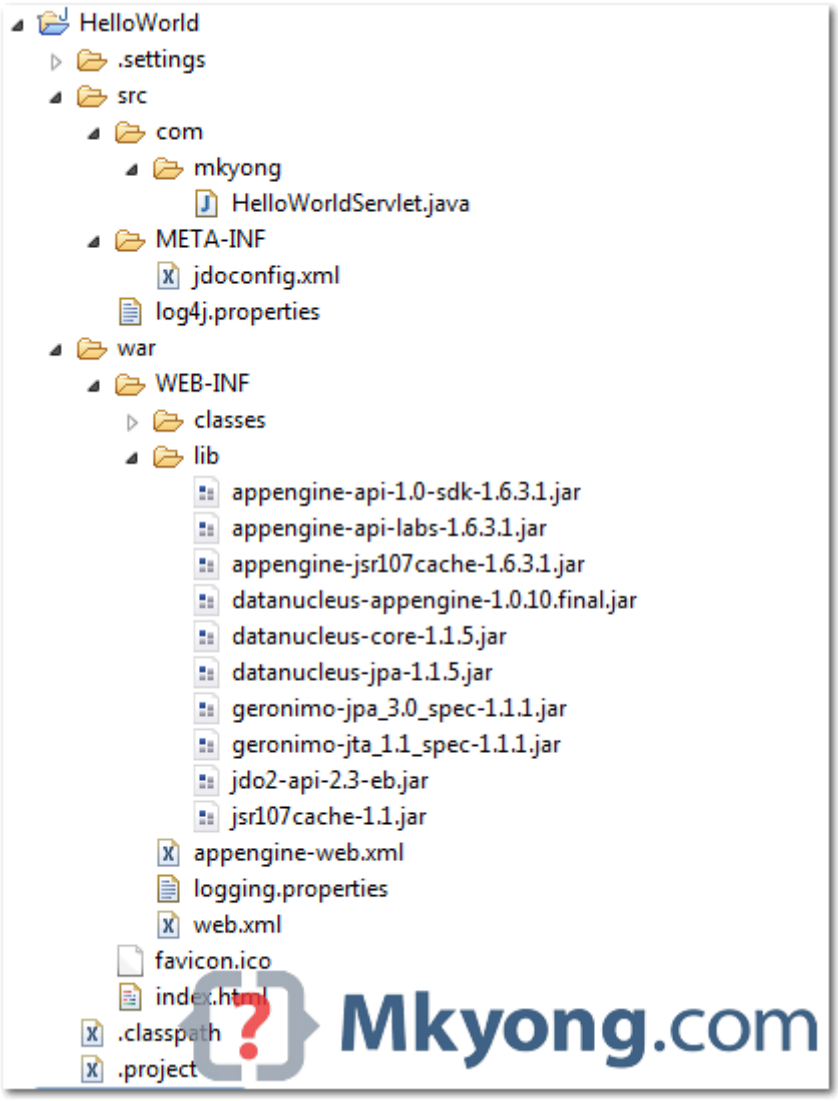
   *Figure – Deselect the "**Google Web ToolKit**", and link your GAE Java SDK via the "**configure SDK**" link*

   Click finished, Google Plugin for Eclipse will generate a sample project automatically.




*3. Hello World*

Review the generated project directory.




Nothing special, a standard Java web project structure

HelloWorld/ src/
...Java source code...
META-INF/
...other configuration... war/
...JSPs, images, data files...
WEB-INF/
...app configuration... lib/
...JARs for libraries... classes/
...compiled classes...

Copy

The extra is this file "appengine-web.xml", Google App Engine need this to run and deploy the application.

*File : appengine-web.xml*

```
<?xml version="1.0" encoding="utf-8"?>
```

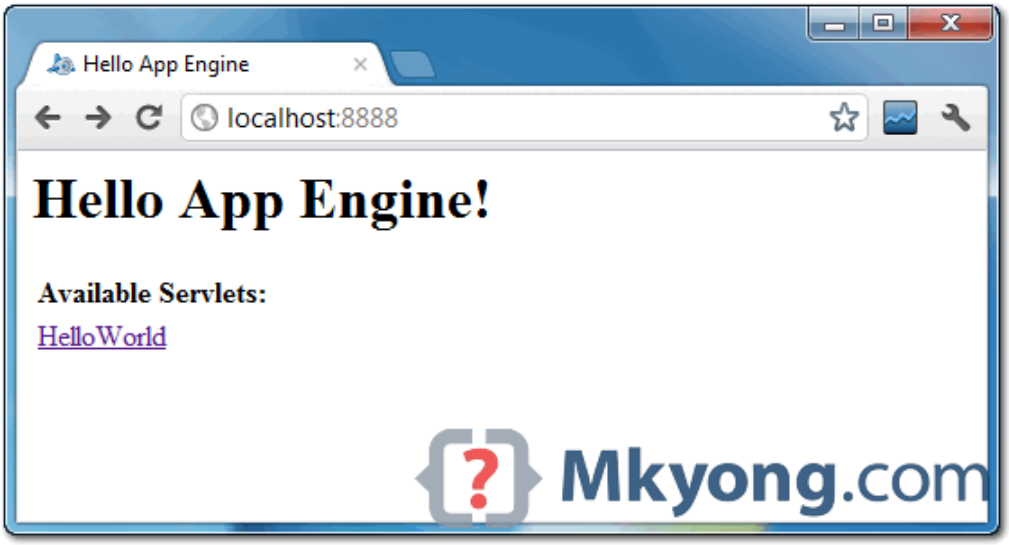
```
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
<application></application>
<version>1</version>

<!-- Configure java.util.logging -->
<system-properties>
<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
</system-properties>
</appengine-web-app>
```

Copy

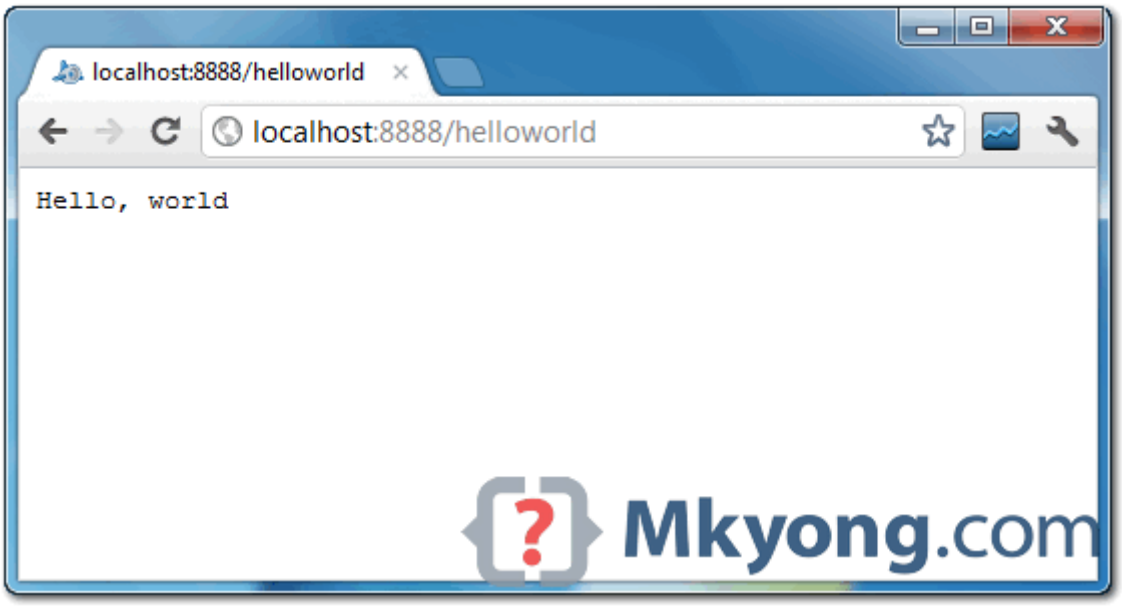
*4. Run it local*

Right click on the project and run as "**Web Application**".
*Eclipse console :*
//...
INFO: The server is running at http://localhost:8888/
30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl start INFO: The admin console is running at http://localhost:8888/_ah/admin

Copy
Access URL http://localhost:8888/, see output



and also the hello world servlet – http://localhost:8888/helloworld




*5. Deploy to Google App Engine*

Register an account on https://appengine.google.com/, and create an application ID for your web application.

In this demonstration, I created an application ID, named "mkyong123", and put it in appengine web.xml.

*File : appengine-web.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
<application>mkyong123</application>
<version>1</version>

<!-- Configure java.util.logging -->
<system-properties>
<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
</system-properties>

</appengine-web-app>
```

Copy
To deploy, see following steps:

*Figure 1.1 – Click on GAE deploy button on the toolbar.*

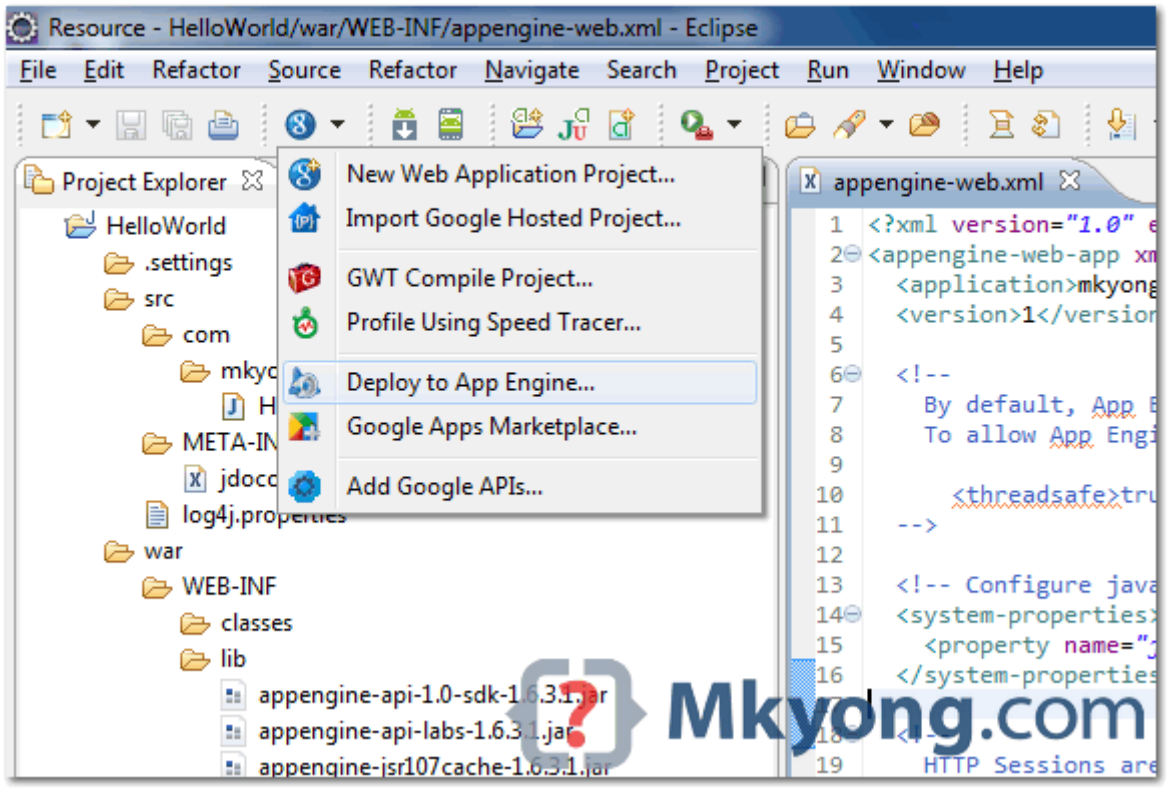



*Figure 1.2 – Sign in with your Google account and click on the Deploy button.*

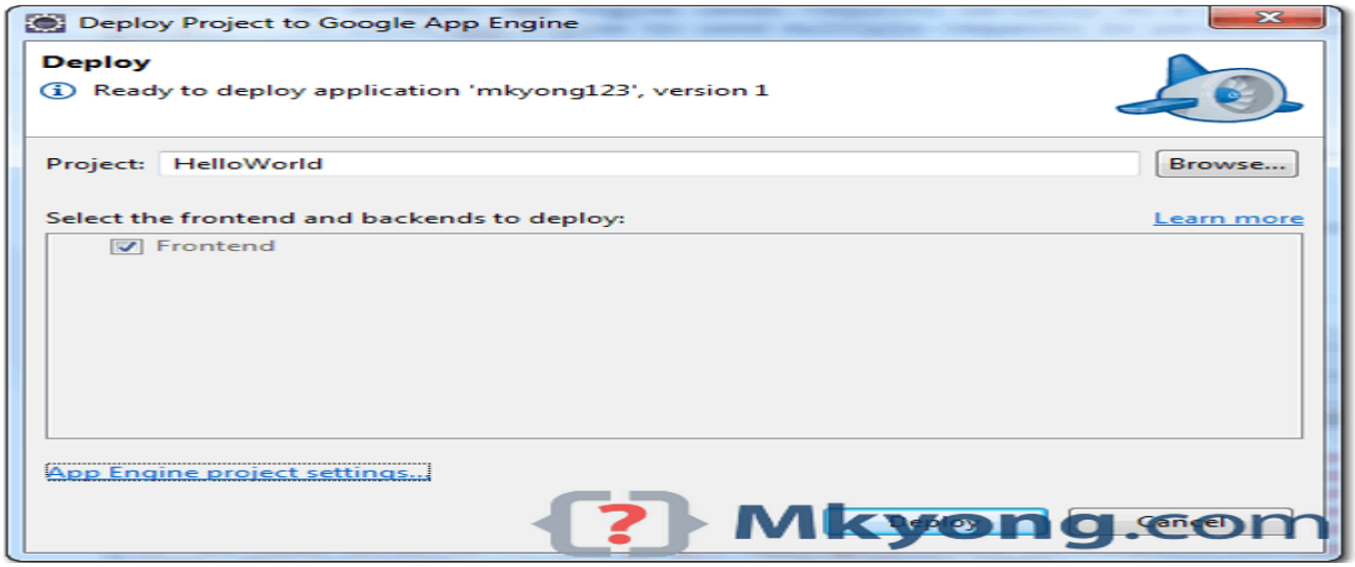



*Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – [http://mkyong123.appspot.com/](http://mkyong123.appspot.com/)*

Hello App Engine
mkyong123.appspot.com
Hello App Engine!
Available Servlets:
HelloWorld
Mkyong.com

**Result:**

Thus the simple application was created successfully.

**EXP NO : 4** **DATE: 30/07/2025**

# Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

**Aim:**

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

**Steps:**

**How to use CloudSim in Eclipse**

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

To use CloudSim in Eclipse:

1. Download CloudSim installable files from *https://code.google.com/p/cloudsim/downloads/list and unzip*
2. Open Eclipse
3. Create a new Java Project: File -> New
4. Import an unpacked CloudSim project into the new Java Project

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

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

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

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

5. The third step is to create a broker:

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

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

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

Submit the VM list to the broker: broker.submitVmList(vmlist)

9. Create a cloudlet with length, file size, output size, and utilisation model: Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationMode

10. Submit the cloudlet list to the broker: broker.submitCloudletList(cloudletList)

Sample Output from the Existing Example:

```
Starting
CloudSimExample1... Initialising...
Starting CloudSim version 3.0 Datacenter_0 is starting...
>>>
```

```
========== OUTPUT ==========
Cloudlet ID  STATUS  Data center ID    VM ID   Time    Start Time
    Finish Time 0        SUCCESS   2                 0            400
              0.1          400.1
*****Datacenter: Datacenter_0***** User id
            Debt
    3       35.6

CloudSimExample1 finished!
```

**Result:**

The simulation was successfully executed

**EXP NO : 5** **DATE: 06/09/2025**

## Use GAE launcher to launch the web applications.

**Aim:**

To Use GAE launcher to launch the web applications.

**Steps:**

**MakingyourFirst Application**

Nowyouneedtocreateasimpleapplication. Wecouldusethe "+"optiontohave the launcher make us an application – but instead we willdo it byhand to get a better sense of what is going on.

Makeafolder for your Google App Engineapplications. I amgoing tomakethe Folder on my Desktop called "**apps**" – the pathto this folder is:

C:\Documents and Settings\csev\Desktop\apps

And then make a sub---folder in within apps called "ae--01--trivial" – the path to this folder wouldbe:

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

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

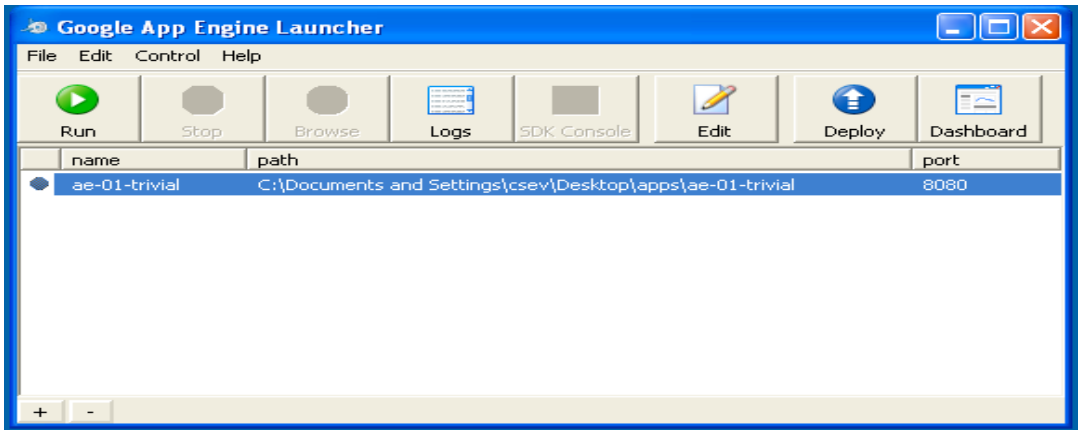
```
application: ae-01-trivial version: 1
runtime: python api_version: 1 handlers:- url: /.*
script: index.py
```

Note: Please do notcopyandpaste these lines into yourtexteditor– youmightend up with strange characters – simply type them into your editor.

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

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

Then start the GoogleAppEngineLauncher program that can be found under Applications. Use the File --> Add Existing Application command and navigate into the apps directory and select the ae--01--trivial folder. Once you have added the application, select it so that youcancontrolthe application usingthelauncher.




Once youhave selected yourapplicationand press Run. Aftera few moments your application willstart and the launcher willshow a little green icon next to your application. Then press Browse to open a browser pointing at your application which is running at http://localhost:8080/

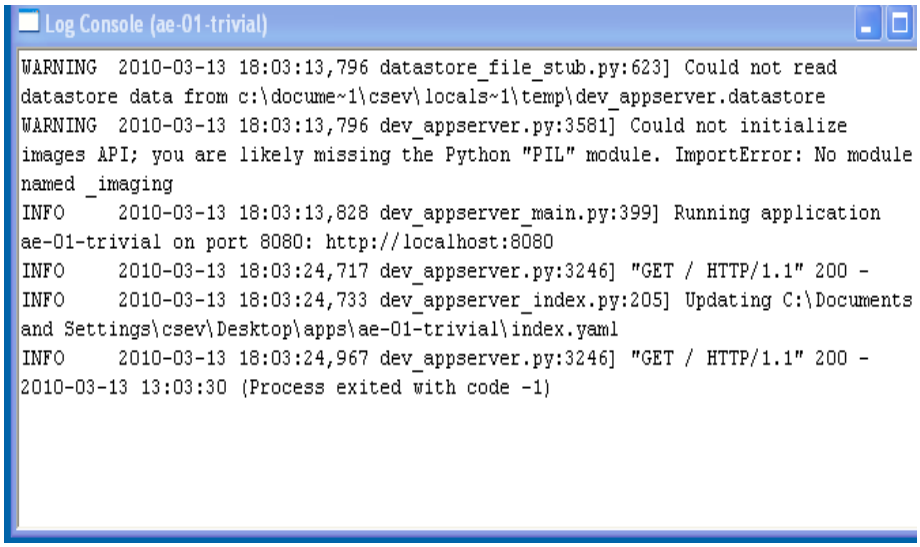
Paste http://localhost:8080 into your browser and you should see your application as follows:

Just for fun, edit the index.py to change the name "Chuck" to your own name and press Refresh in the browser to verify your updates.

**Watching the Log**

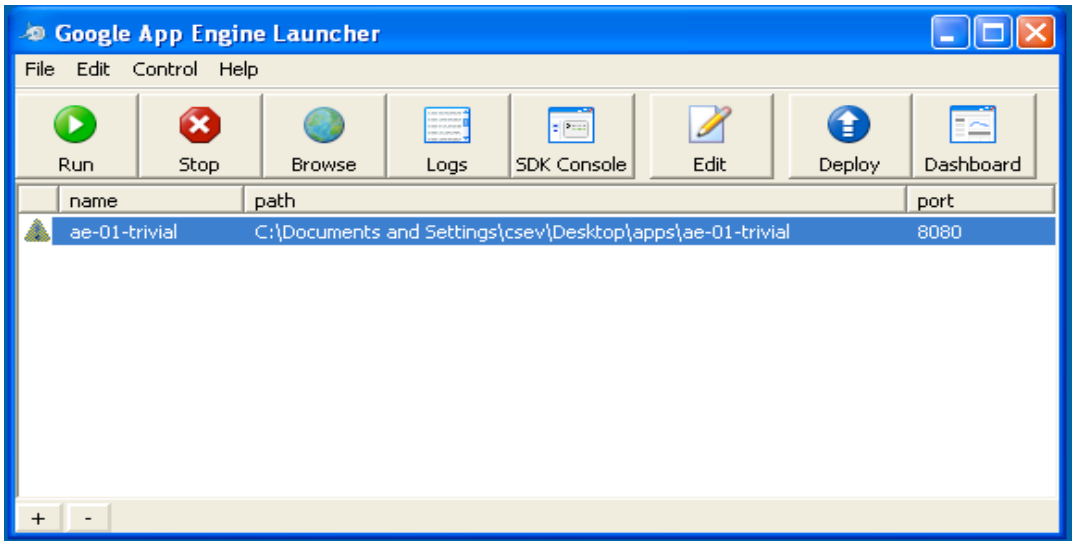
You can watch the internal log of the actions that the webserver is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the Logs button to bring up a log window:

EachtimeyoupressRefreshinyourbrowser–you can see it retrieving the output with a GET request.

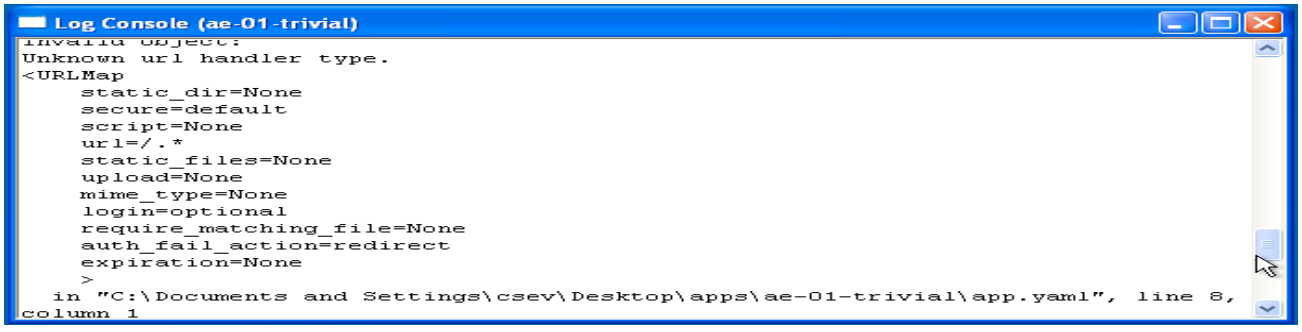



## Dealing With Errors

With two files to edit, there are two general categories of errors that you may encounter. If youmake a mistake onthe app.yamlfile, the App Engine willnotstart and your launcher will show a yellow icon near your application:

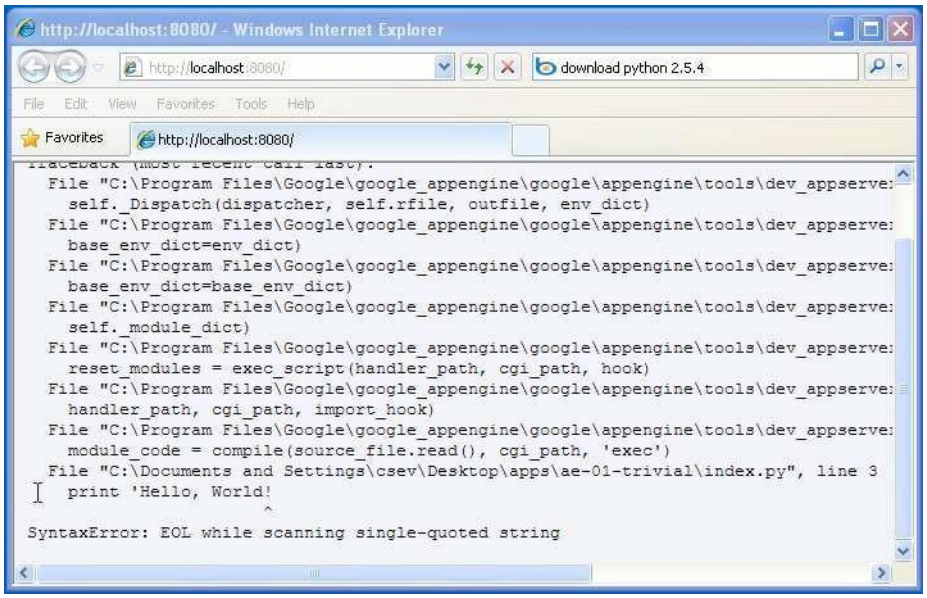



To get more detail on what is going wrong, take a look at the log for the application:



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

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




The error you need to see is likely to be the last few lines of the output – in this case I made a Python syntax error on line one of our one---line application.

Reference: http://en.wikipedia.org/wiki/Stack_trace

When you make a mistake in the app.yaml file – you must the fix the mistake and attempt to start the application again.

If you make a mistake in a file like index.py, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

**Shutting Down the Server**

To shut down the server, use the Launcher, select your application and press the Stop button.

**Result:**

Thus the GAE web applications was created.

**EXP NO : 6** **DATE: 20/08/2025**

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

**Aim:**

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

**Steps:**

1. You can copy few (or more) lines with *copy & paste* mechanism.
   For this you need to share clipboard between host OS and guest OS, installing Guest Addition on both the virtual machines (probably setting *bidirectional* and restarting them).
   You *copy* from *guest OS* in the clipboard that is shared with the *host OS*.
   Then you *paste* from the *host OS* to the second *guest OS*.
2. You can enable drag and drop too with the same method (Click on the machine, settings, general, advanced, drag and drop: set to *bidirectional* )
3. You can have common *Shared Folders* on both virtual machines and use one of the directory shared as buffer to copy.
   Installing Guest Additions you have the possibility to set Shared Folders too. As you put a file in a shared folder from *host OS* or from *guest OS*, is immediately visible to the other. (Keep in mind that can arise some problems for date/time of the files when there are different clock settings on the different virtual machines).
   *If you use the same folder shared on more machines you can exchange files directly copying them in this folder.*
4. You can use usual method to copy files between 2 different computer with client-server application. (e.g. scp with sshd active for linux, winscp... you can get some info about SSH servers e.g. here)
   You need an active server (sshd) on the receiving machine and a client on the sending machine. Of course you need to have the authorization setted (via password or, better, via an automatic authentication method).
   Note: many Linux/Ubuntu distribution install sshd by default: you can see if it is running with pgrep sshd from a shell. You can install with sudo apt-get install openssh-server.
5. You can mount part of the file system of a virtual machine via NFS or SSHFS on the other, or you can share file and directory with Samba. You may find interesting the article Sharing files between guest and host without VirtualBox shared folders with detailed step by step instructions.

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

- Each virtual machine has its own operative system running on and acts as a physical machine.
- Each virtual machine is an instance of a program *owned* by an *user* in the hosting

operative system and should undergo the restrictions of the *user* in the *hosting OS*.
E.g Let we say that Hastur and Meow are users of the hosting machine, but they did not allow each other to see their directories (no read/write/execute authorization). When each of them run a virtual machine, for the hosting OS those virtual machine are two normal programs owned by Hastur and Meow and cannot see the private directory of the other user. This is a restriction due to the *hosting OS*. It's easy to overcame it: it's enough to give authorization to read/write/execute to a directory or to chose a different directory in which both users can read/write/execute.

- Windows likes mouse and Linux fingers. :-)

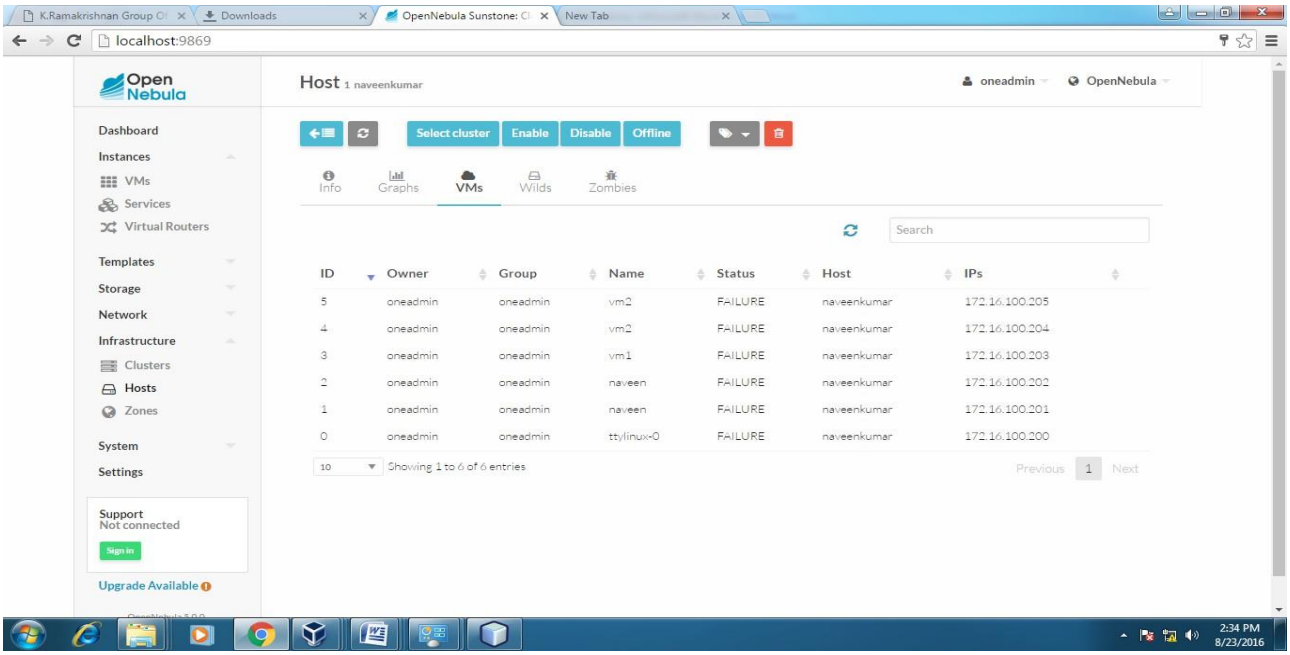
I mean I suggest you to enable *Drag & drop* to be cosy with the Windows machines and the *Shared folders* or to be cosy with Linux.
When you will need to be fast with Linux you will feel the need of ssh-keygen and to Generate once SSH Keys to copy files on/from a remote machine without writing password anymore. In this way it functions bash auto-completion remotely too!

**Procedure:**

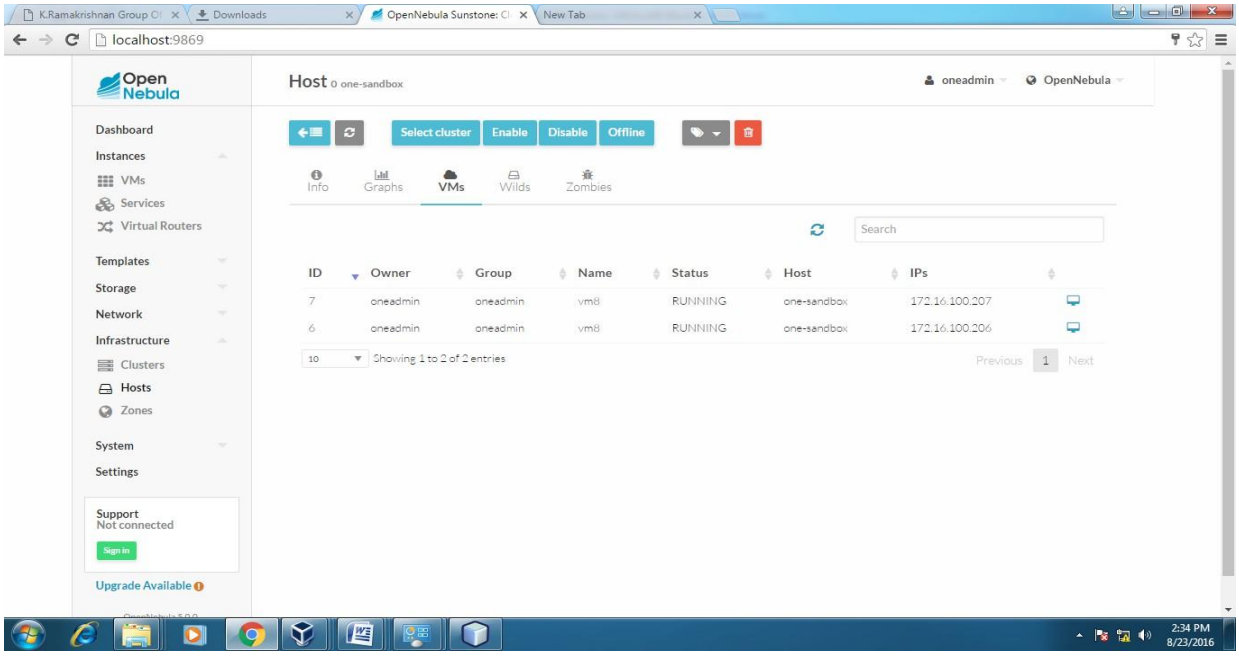
**Steps:**

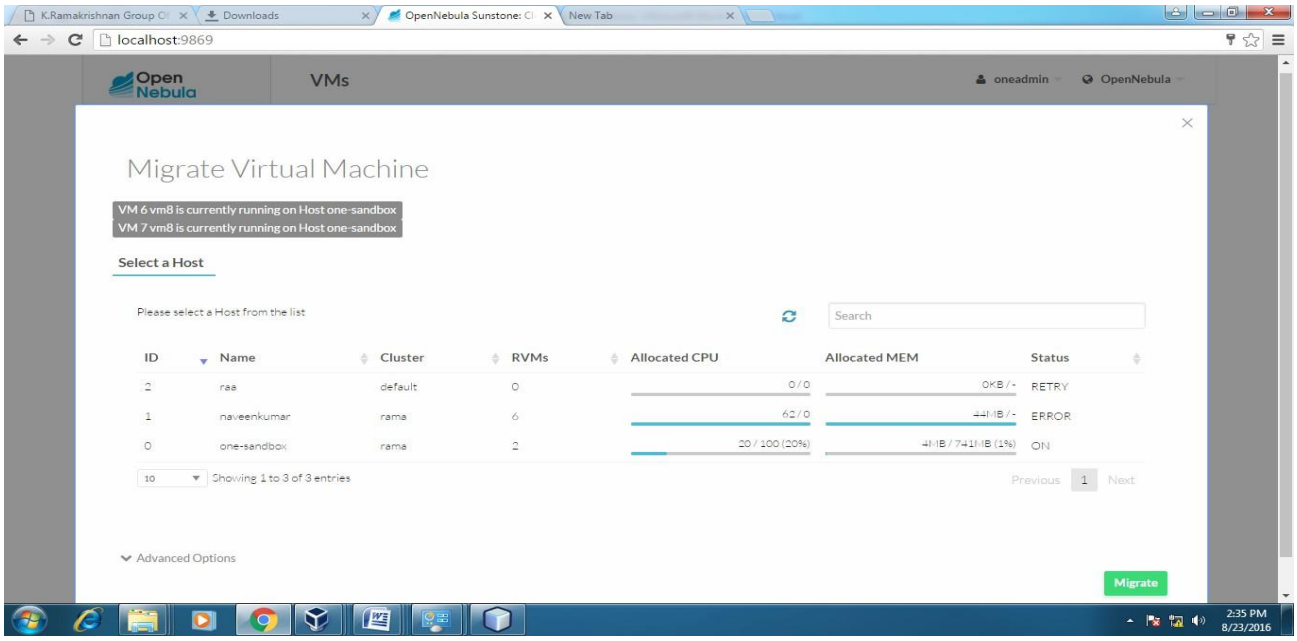
1. Open Browser, type localhost:9869
2. Login using username: oneadmin, password: opennebula
3. Then follow the steps to migrate VMs
    a. Click on infrastructure
    b. Select clusters and enter the cluster name
    c. Then select host tab, and select all host
    d. Then select Vnets tab, and select all vnet
    e. Then select datastores tab, and select all datastores
    f. And then choose host under infrastructure tab
    g. Click on + symbol to add new host, name the host then click on create.
4. on instances, select VMs to migrate then follow the stpes
    a. Click on 8$^{th}$ icon ,the drop down list display
    b. Select migrate on that ,the popup window display
    c. On that select the target host to migrate then click on migrate.

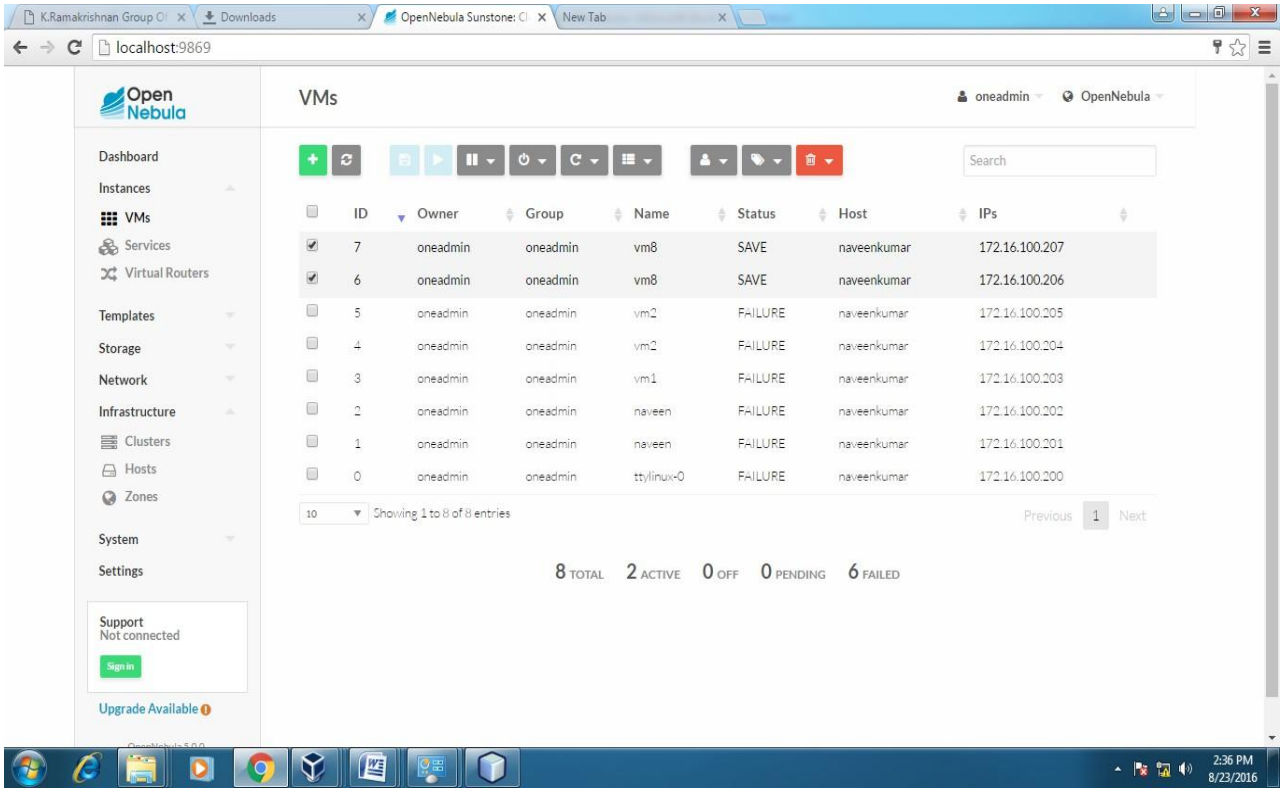
**Before migration**
**Host:SACET**



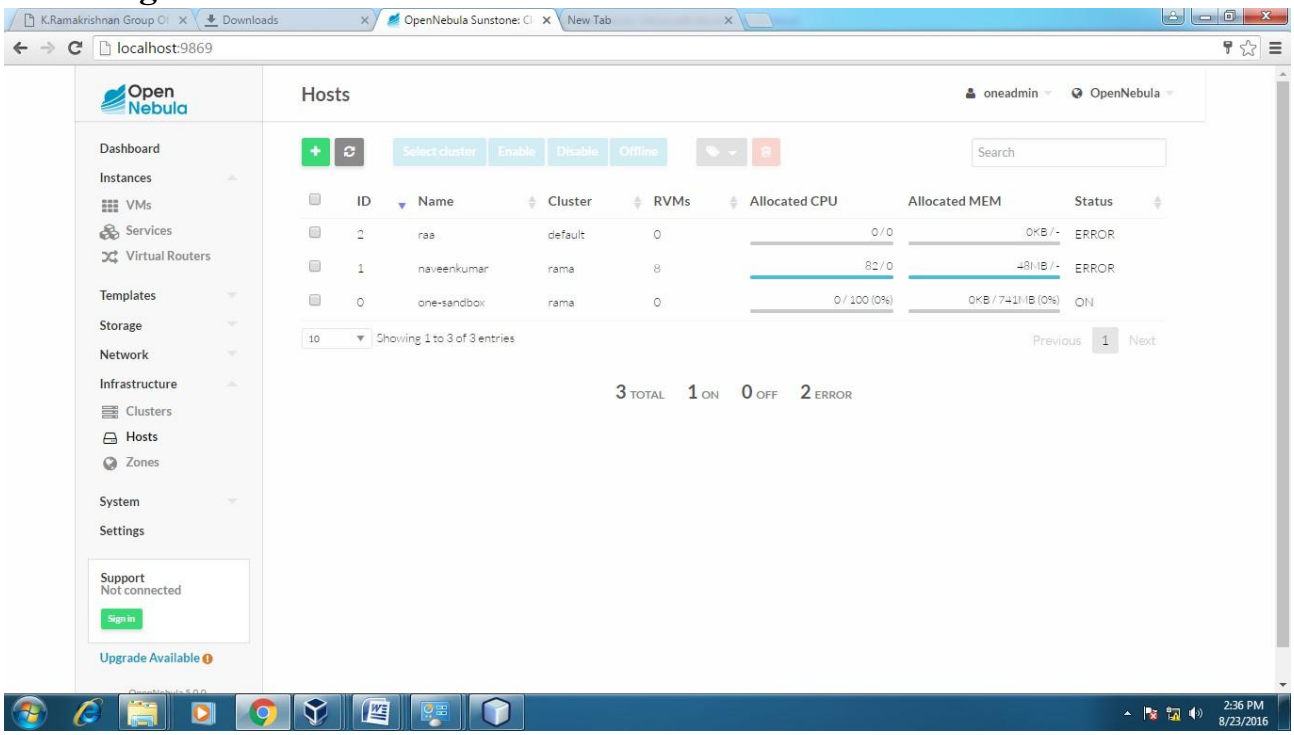

## Host:one-sandbox



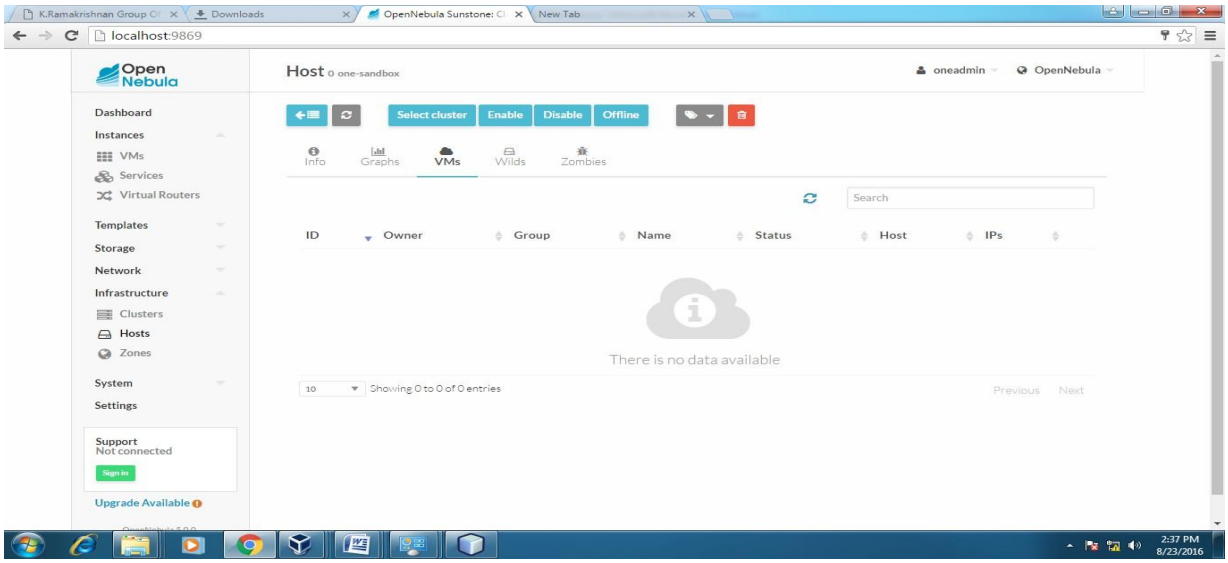

localhost:9869
Open Nebula
VMs
oneadmin
OpenNebula
Migrate Virtual Machine
VM 6 vm8 is currently running on Host one-sandbox
VM 7 vm8 is currently running on Host one-sandbox
Select a Host
Please select a Host from the list
Search
ID
Name
Cluster
RVMs
Allocated CPU
Allocated MEM
Status
2 rae default 0 0 / 0 0KB / - RETRY
1 naveenkumar rama 6 62 / 0 44MB / - ERROR
0 one-sandbox rama 2 20 / 100 (20%) 4MB / 741MB (1%) ON
Showing 1 to 3 of 3 entries
Previous
1
Next
Advanced Options
Migrate
2:35 PM
8/23/2016


localhost:9869
Open Nebula
VMs
oneadmin
OpenNebula
Dashboard
Instances
VMs
Services
Virtual Routers
Templates
Storage
Network
Infrastructure
Clusters
Hosts
Zones
System
Settings
Support
Not connected
Sign in
Upgrade Available
Search
ID Owner Group Name Status Host IPs
7 oneadmin oneadmin vm8 SAVE naveenkumar 172.16.100.207
6 oneadmin oneadmin vm8 SAVE naveenkumar 172.16.100.206
5 oneadmin oneadmin vm2 FAILURE naveenkumar 172.16.100.205
4 oneadmin oneadmin vm2 FAILURE naveenkumar 172.16.100.204
3 oneadmin oneadmin vm1 FAILURE naveenkumar 172.16.100.203
2 oneadmin oneadmin naveen FAILURE naveenkumar 172.16.100.202
1 oneadmin oneadmin naveen FAILURE naveenkumar 172.16.100.201
0 oneadmin oneadmin ttylinux-0 FAILURE naveenkumar 172.16.100.200
Showing 1 to 8 of 8 entries
Previous
1
Next
8 TOTAL 2 ACTIVE 0 OFF 0 PENDING 6 FAILED
2:36 PM
8/23/2016

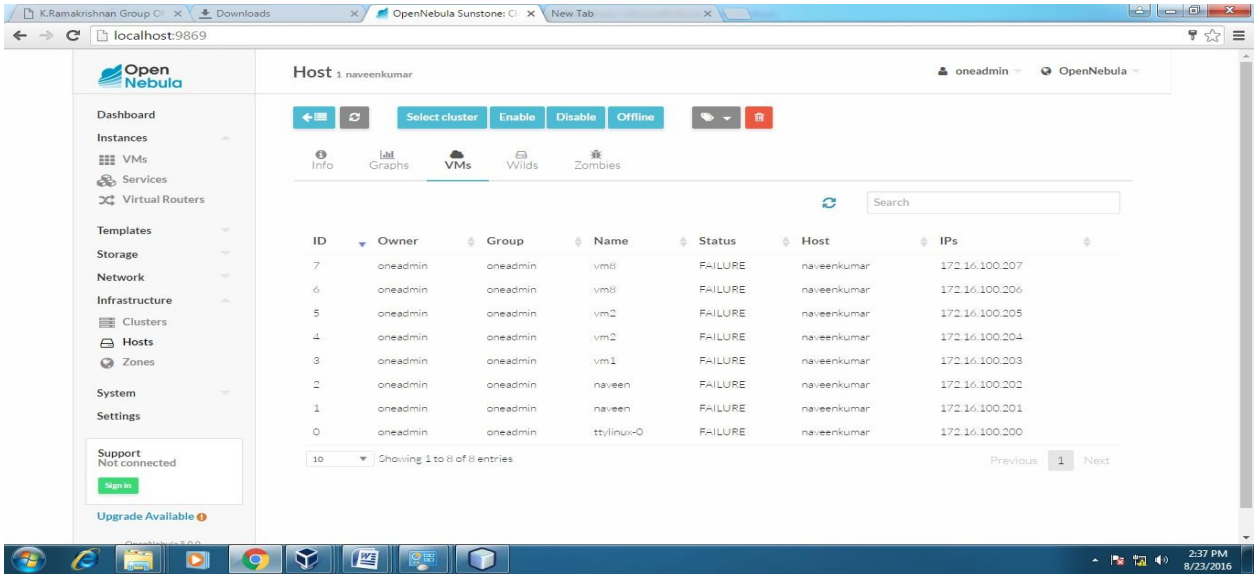
## After Migration:




## Host:one-sandbox



**Host:SACET**




## Applications:

Easily migrate your virtual machine from one pc to another.

**Result:**

Thus the file transfer between VM was successfully completed…..

**EXP NO : 7** **DATE: 03/09/2025**

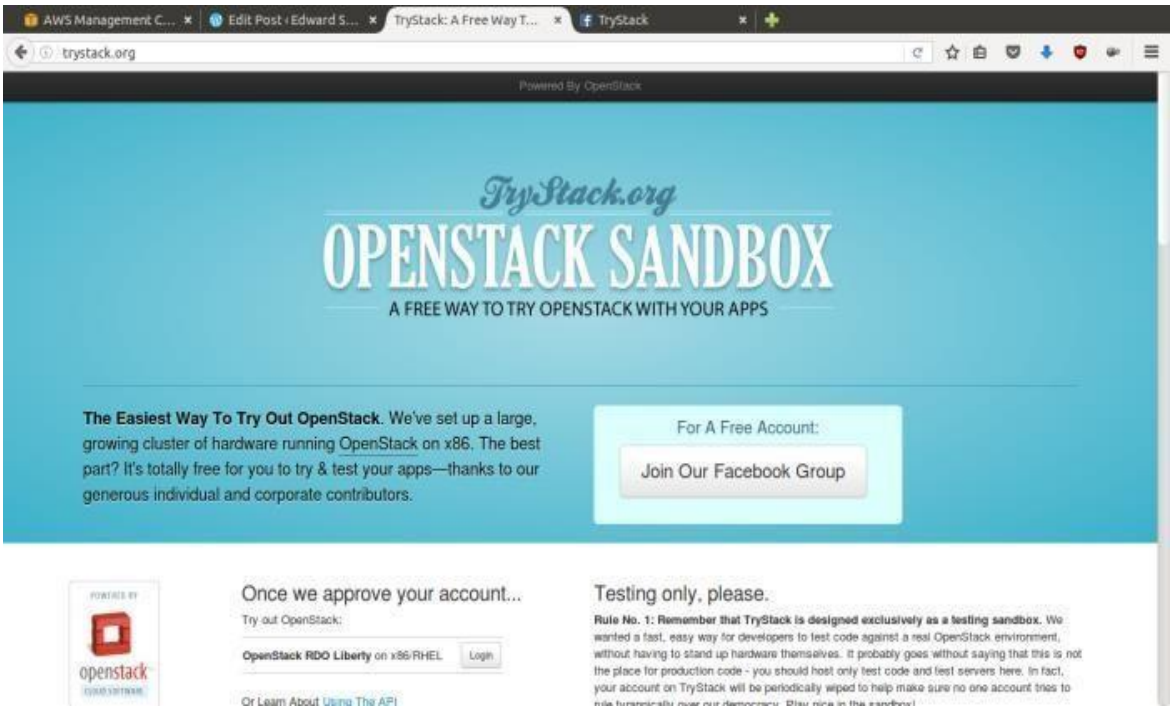
## Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

**Aim:**

To Find a procedure to launch virtual machine using trystack.

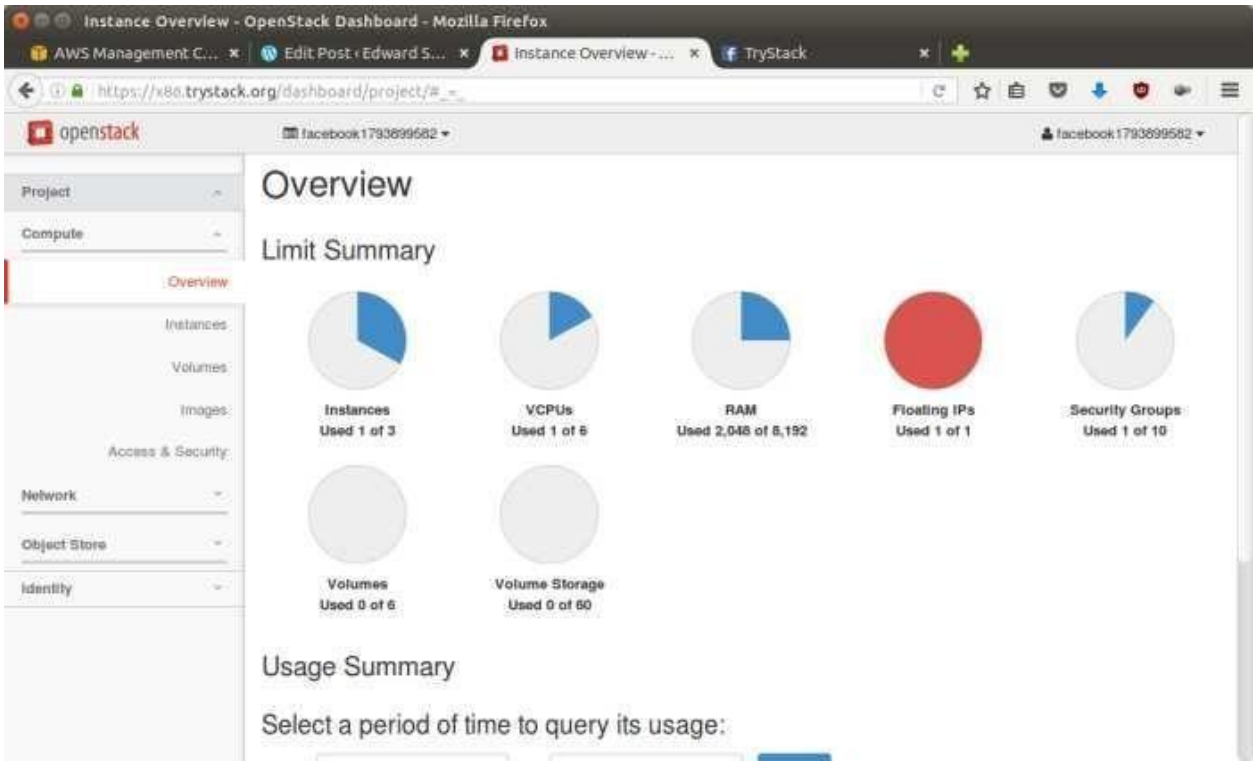
**Steps:**

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




TryStack.org Homepage

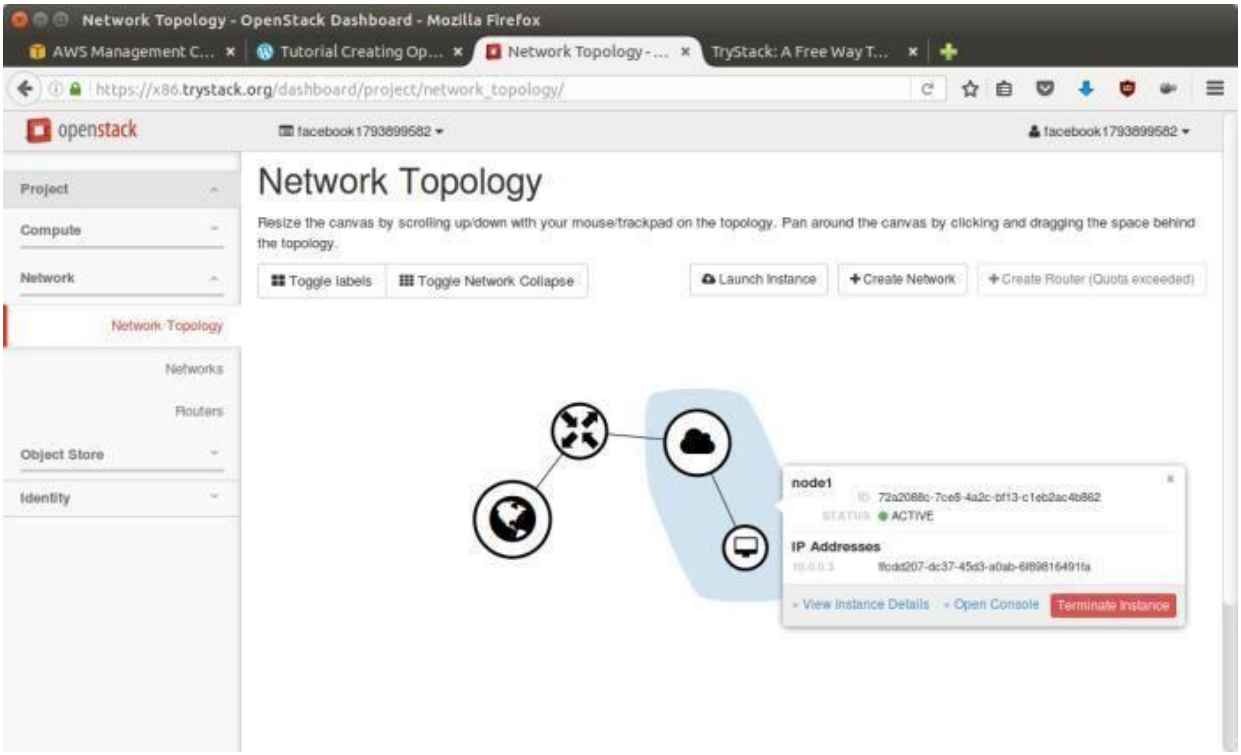
I assume that you already join to the Facebook Group and login to the dashboard. After you log in to the TryStack, you will see the Compute Dashboard like:




OpenStack Compute Dashboard

**Overview: What we will do?**

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



Network topology

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

**Step 1: Create Network**

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

1. Go to **Network > Networks** and then click **Create Network**.
2. In **Network** tab, fill **Network Name** for example internal and then click **Next**.
3. In **Subnet** tab,
    1. Fill **Network Address** with appropriate CIDR, for example 192.168.1.0/24. Use private network CIDR block as the best practice.
    2. Select **IP Version** with appropriate IP version, in this case IPv4.
    3. Click **Next**.
    4. In **Subnet Details** tab, fill **DNS Name Servers** with 8.8.8.8 (Google DNS) and then click **Create**.

**Step 2: Create Instance**

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

1. Go to **Compute > Instances** and then click **Launch Instance**.
2. In **Details** tab,
    1. Fill **Instance Name**
    2. **Flavor**, for example m1.medium.
    3. Fill **Instance Count** with **1**.
    4. Select **Instance Boot Source** with **Boot from Image**.
    5. Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want install Ubuntu 14.04 in your virtual machine.
3. In **Access & Security** tab,
    1. Click [+] button of **Key Pair** to import key pair. This key pair is a public and private key that we will use to connect to the instance from our machine.
    2. In **Import Key Pair** dialog,
        1. Fill **Key Pair Name** with your machine name (for example Edward-Key).
        2. Fill **Public Key** with your **SSH public key** (usually is in ~/.ssh/id_rsa.pub). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate key pair.
        3. Click **Import key pair**.
    3. In **Security Groups**, mark/check **default**.
4. In **Networking** tab
    1. In **Selected Networks**, select network that have been created in Step 1, for example internal.
5. Click Launch.
6. If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu 2.

**Step 3: Create Router**

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

1. Go to **Network > Routers** and then click **Create Router**.
2. Fill **Router Name** for example router1 and then click **Create router**.
3. Click on your **router name link**, for example router1, **Router Details** page.
4. Click **Set Gateway** button in upper right:
    1. Select **External networks** with **external**.
    2. Then **OK**.
5. Click **Add Interface** button.
    1. Select **Subnet** with the network that you have been created in Step 1.
    2. Click **Add interface**.
6. Go to **Network > Network Topology**. You will see the network topology. In the example, there are two network, i.e. external and internal, those are bridged by a router. There are instances those are joined to internal network.

**Step 4: Configure Floating IP Address**

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

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

**Step 5: Configure Access & Security**

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

1. Go to **Compute > Access & Security** and then open **Security Groups** tab.
2. In **default** row, click **Manage Rules**.
3. Click **Add Rule**, choose **ALL ICMP** rule to enable ping into your instance, and then
4. click**Add**.
5. Click **Add Rule**, choose **HTTP** rule to open HTTP port (port 80), and then click **Add**.
6. Click **Add Rule**, choose **SSH** rule to open SSH port (port 22), and then click **Add**.
7. You can open other ports by creating new rules.

**Step 6: SSH to Your Instance**

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

**Result:**

Thus the openstack demo worked successfully.

**EXP NO : 8** **DATE: 10/09/2025**

# Install Hadoop single node cluster and run simple applications like wordcount.

## Aim:

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

## Steps:

### Install Hadoop

**Step 1:** Click here to download the Java 8 Package. Save this file in ou home directory.

**Step 2:** Extract the Java Tar File.

***Command*:** tar -xvf jdk-8u101-linux-i586.tar.gz

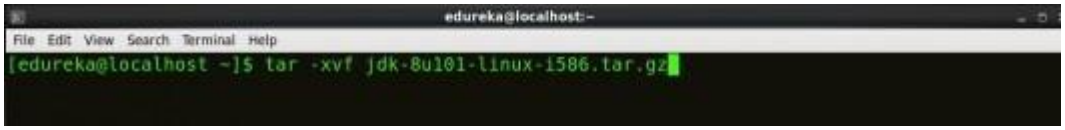



*Fig: Hadoop Installation – Extracting Java Files*

**Step 3: Download the Hadoop 2.7.3 Package.**

***Command*:** wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop- 2.7.3.tar.gz

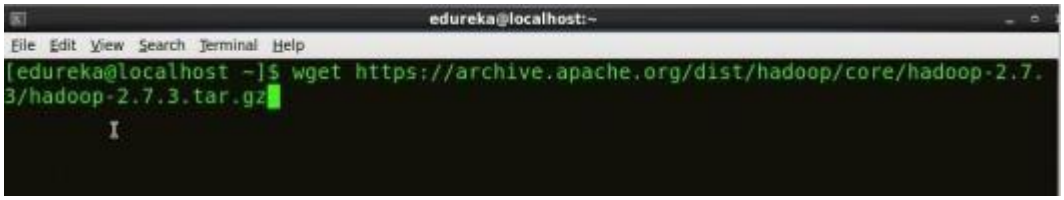



*Fig: Hadoop Installation – Downloading Hadoop*

**Step 4:** Extract the Hadoop tar File.

***Command***: tar -xvf hadoop-2.7.3.tar.gz

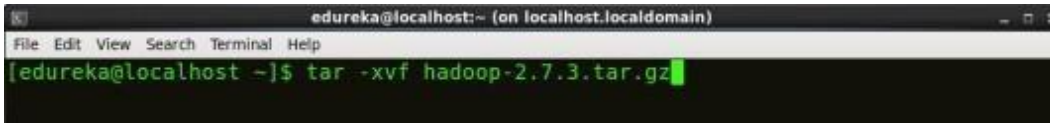
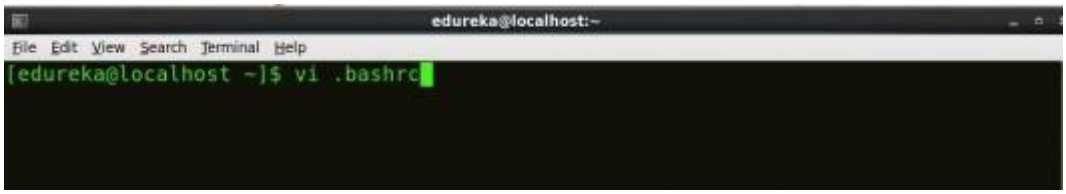


*Fig: Hadoop Installation – Extracting Hadoop Files*

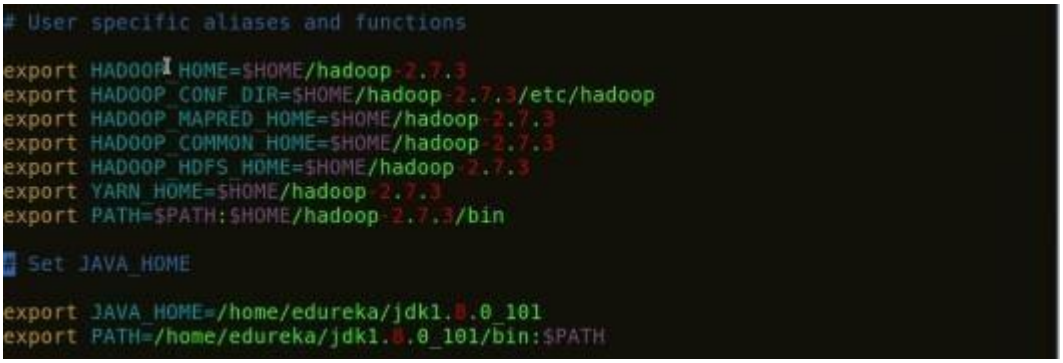
**Step 5:** Add the Hadoop and Java paths in the bash file (.bashrc). Open**. bashrc** file. Now, add Hadoop and Java Path as shown below.

***Command*:** vi .bashrc

*Fig: Hadoop Installation – Setting Environment Variable*




Then, save the bash file and close it.




For applying all these changes to the current Terminal, execute the source command.
***Command*:** source .bashrc

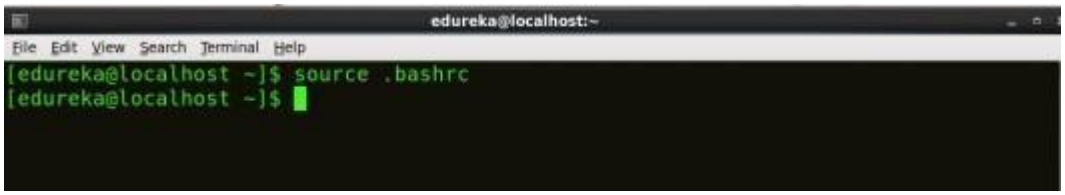



*Fig: Hadoop Installation – Refreshing environment variables*

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

***Command*:** java -version

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

***Command*:** hadoop version

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

*Fig: Hadoop Installation – Checking Hadoop Version*

**Step 6:** Edit the **Hadoop Configuration files**.

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

***Command:*** ls

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

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

**Step 7:** Open *core-site.xml* and edit the property mentioned below inside configuration tag:

*core-site.xml* informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

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

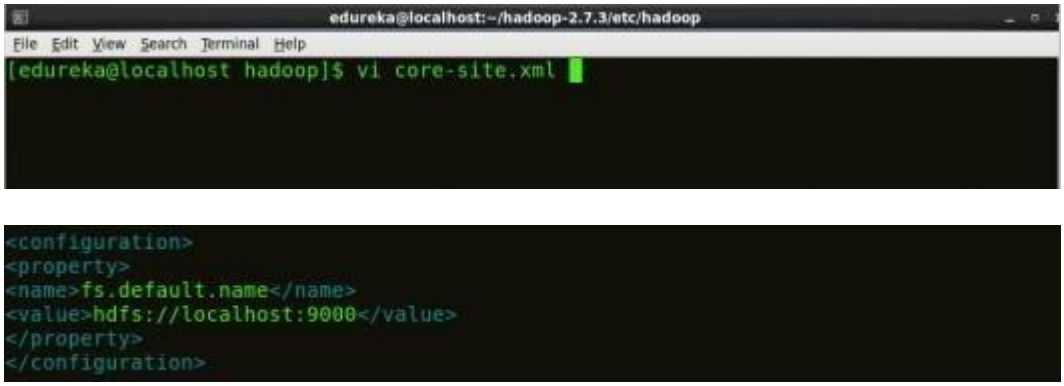



*Fig: Hadoop Installation – Configuring core-site.xml*

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

**Step 8:** Edit *hdfs-site.xml* and edit the property mentioned below inside **configuration tag:**

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

***Command*:** vi hdfs-site.xml

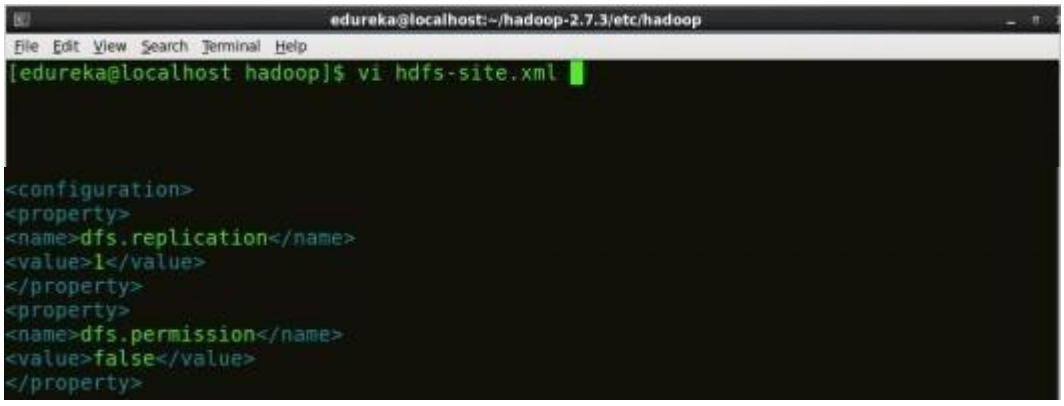


*Fig: Hadoop Installation – Configuring hdfs-site.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
</configuration>
```

**Step 9:** Edit the *mapred-site.xml* file and edit the property mentioned below **inside configuration tag:**

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

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

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

***Command*:** vi mapred-site.xml.

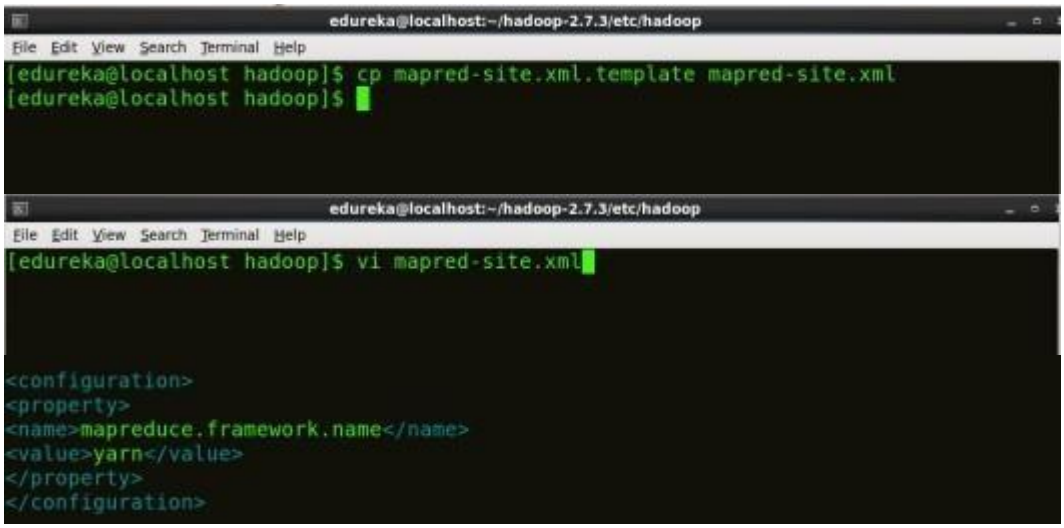



*Fig: Hadoop Installation – Configuring mapred-site.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

**Step 10:** Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:

*yarn-site.xml* contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

***Command*:** vi yarn-site.xml

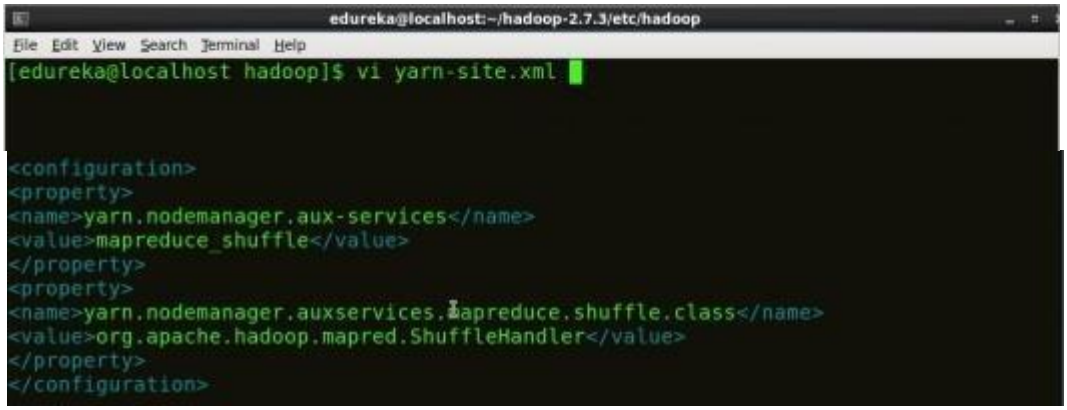



*Fig: Hadoop Installation – Configuring yarn-site.xml*

**Step 11:** Edit *hadoop-env.sh* and add the Java Path as mentioned below:

```
<?xml version="1.0">
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</
name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

*hadoop-env.sh* contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

***Command*:** vi hadoop–env.sh

*Fig: Hadoop Installation – Configuring hadoop-env.sh*

**Step 12:** Go to Hadoop home directory and format the

NameNode.

***Command*:** cd

***Command*:** cd hadoop-2.7.3

***Command*:** bin/hadoop namenode -format

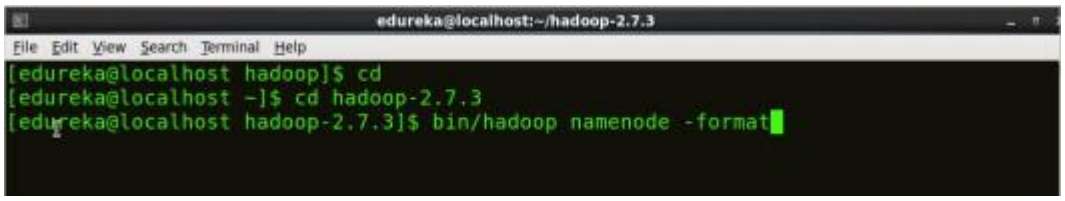



*Fig: Hadoop Installation – Formatting NameNode*

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

**Step 13:** Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.

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

Either you can start all daemons with a single command or do it individually.

***Command:*** ./start-all.sh

The above command is a combination of ***start-dfs.sh, start-yarn.sh*** & ***mr-jobhistory- daemon.sh***

Or you can run all the services individually as below:

**Start NameNode:**

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

***Command:*** ./hadoop-daemon.sh start namenode

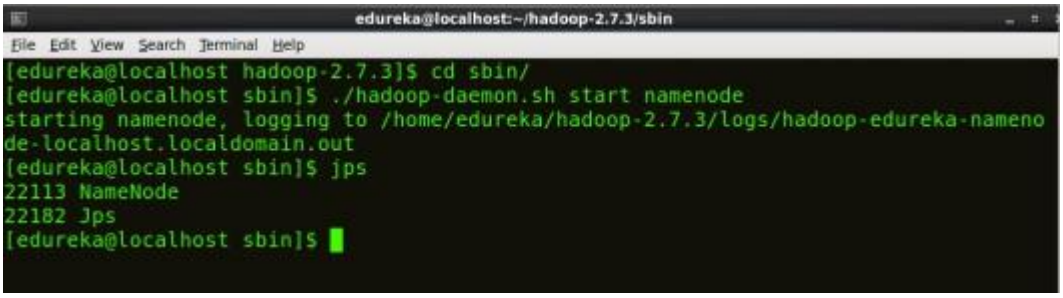



*Fig: Hadoop Installation – Starting NameNode*

**Start DataNode:**

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

***Command:*** ./hadoop-daemon.sh start datanode

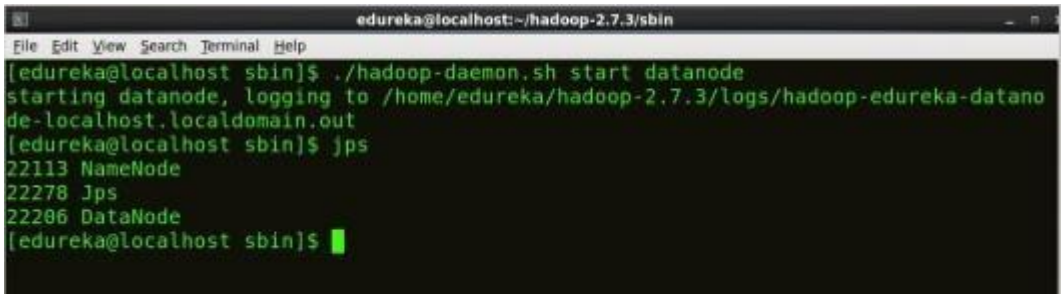



*Fig: Hadoop Installation – Starting DataNode*

**Start ResourceManager:**

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each

NodeManagers and the each application's ApplicationMaster.

***Command:*** ./yarn-daemon.sh start resourcemanager

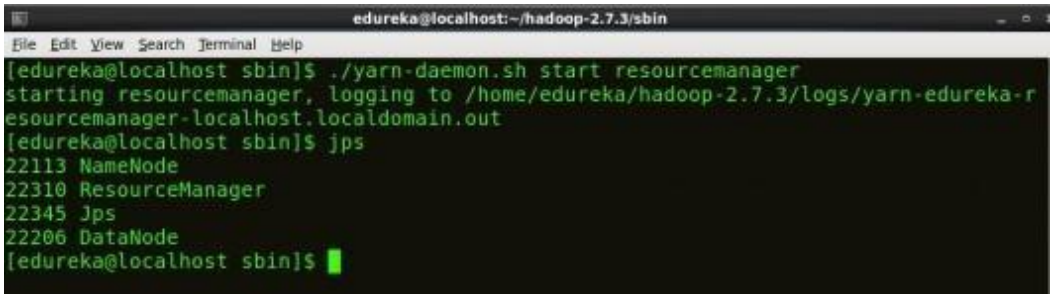



*Fig: Hadoop Installation – Starting ResourceManager*

**Start NodeManager:**

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

***Command:*** ./yarn-daemon.sh start nodemanager

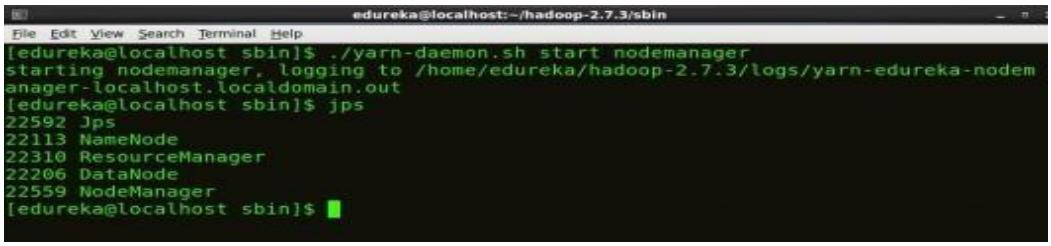


*Fig: Hadoop Installation – Starting NodeManager*

**Start JobHistoryServer:**

JobHistoryServer is responsible for servicing all job history related requests from client.

***Command*:** ./mr-jobhistory-daemon.sh start historyserver

**Step 14:** To check that all the Hadoop services are up and running,

run the below command.

***Command:*** jps

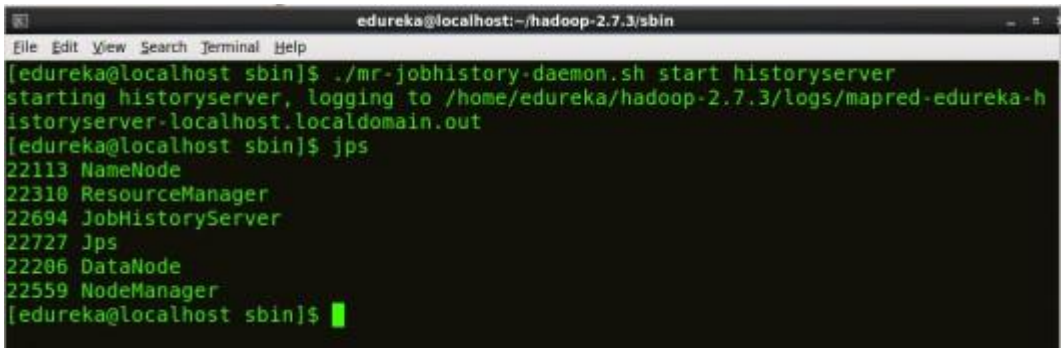



*Fig: Hadoop Installation – Checking Daemons*

**Step 15:** Now open the Mozilla browser and go to **localhost**:**50070/dfshealth.html** to check the NameNode interface.

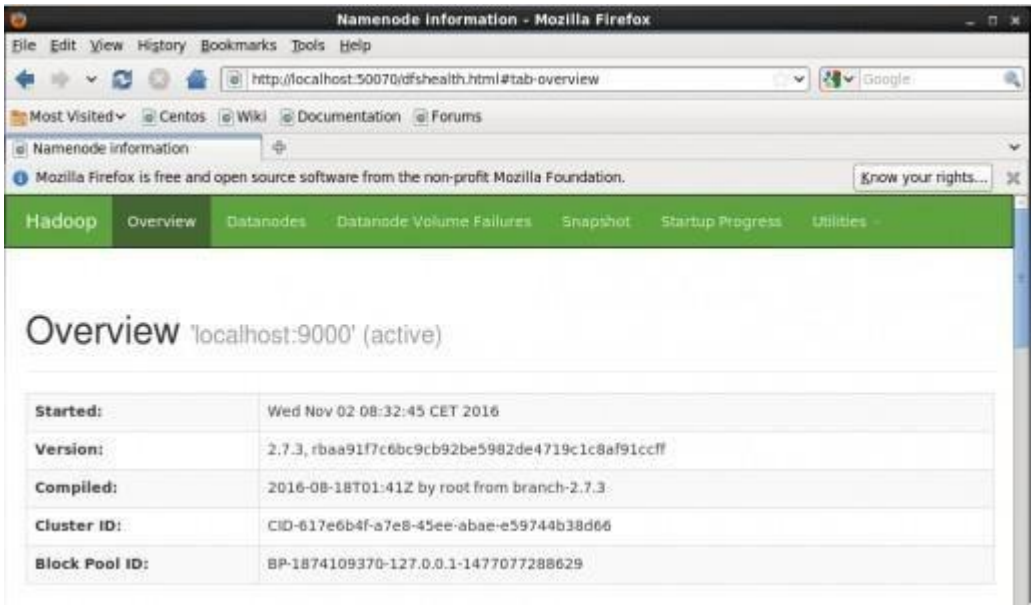



*Fig: Hadoop Installation – Starting WebUI*

Congratulations, you have successfully installed a single node Hadoop cluster

**Result:**

Thus the Hadoop one cluster was installed and simple applications executed successfully.