CCS335 LAB Manual

cloud computing (St.Joseph's College of Engineering)

ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL– R-2021

III YEAR- V SEMESTER

CCS335- CLOUD COMPUTING LABORATORY

2023-2024 ODD SEMESTER

PREPARED BY

Mrs.A.Francis Thivya, AP/CSE

LIST OF EXPERIMENTS

1. Install Virtualbox /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 Simple Programs.
3. Install Google App Engine. Create *hello world* app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Install Hadoop single node cluster and run simple applications like word count.
8. Creating and Executing Your First Container Using Docker.
9. Run a Container from Docker Hub.

TABLE OF CONTENTS

S.NO.	DATE	EXCERCISE TITLE	MARKS	SIGN.
		Install Virtualbox / VMware		
1.		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 Simple Programs		
3.		Install Google App Engine. Create *hello world* app and other simple web applications using python/java.		
4.		Use GAE launcher to launch the web applications.		
5.		Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.		
6.		Find a procedure to transfer the files from one virtual machine to another virtual machine.		
7.		Install Hadoop single node cluster and run simple applications like word count.		
8.		Creating and Executing Your First Container Using Docker.		
9.		Run a Container from Docker Hub		

EX NO. : 1 **Install Virtualbox / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.**

DATE:

Aim:

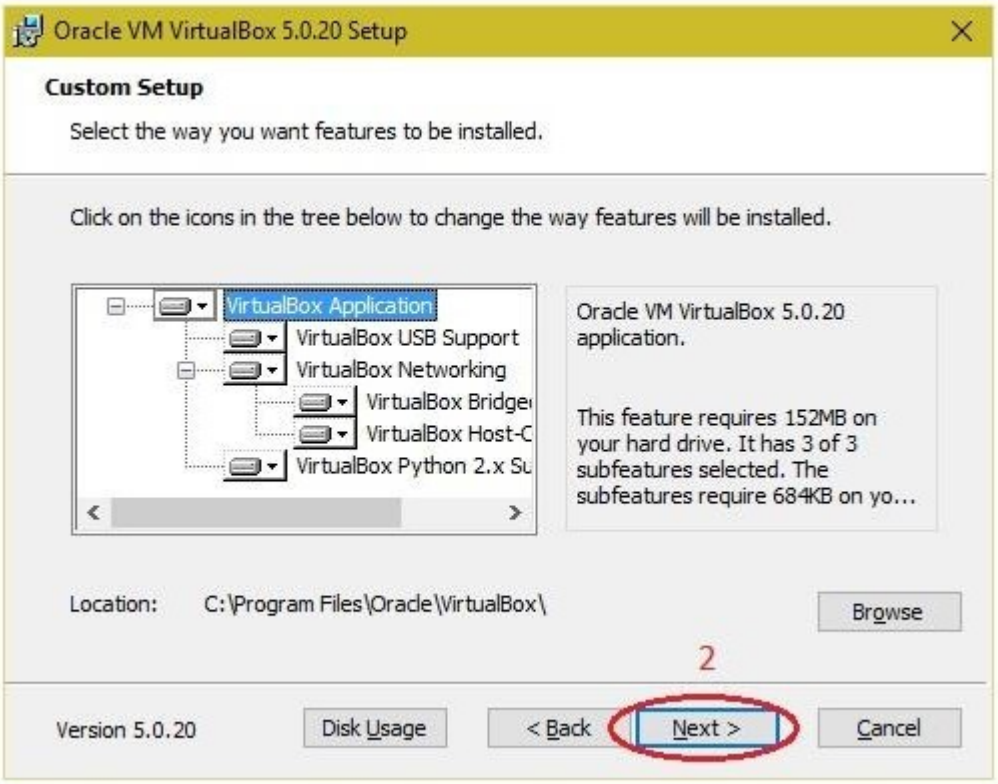
To Install Virtualbox / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

PROCEDURE:

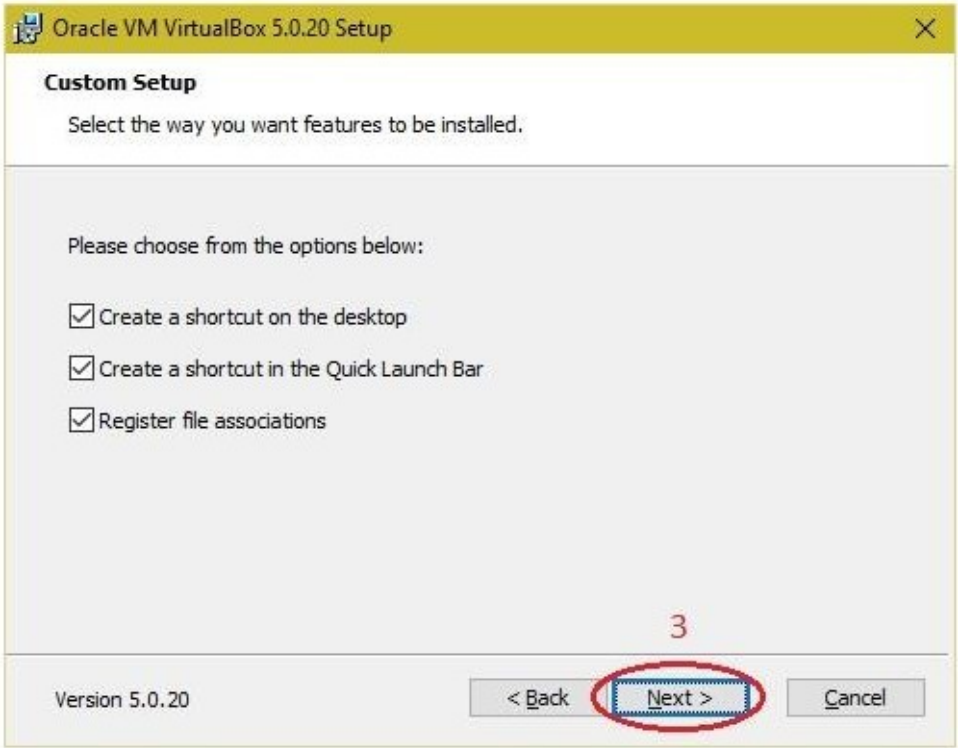
Steps to install Virtual Box:

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

2. Click the next button..

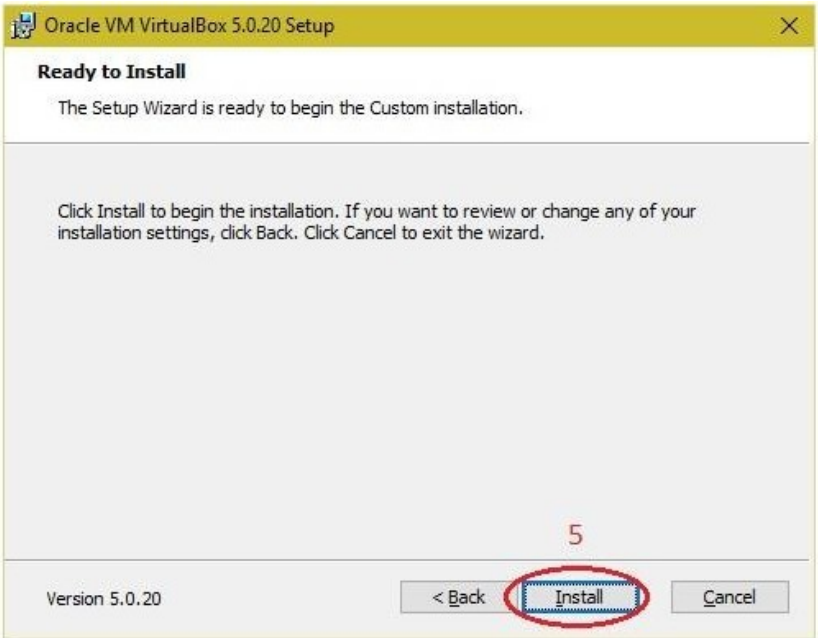


3. Click the next button



4. Click the YES button..

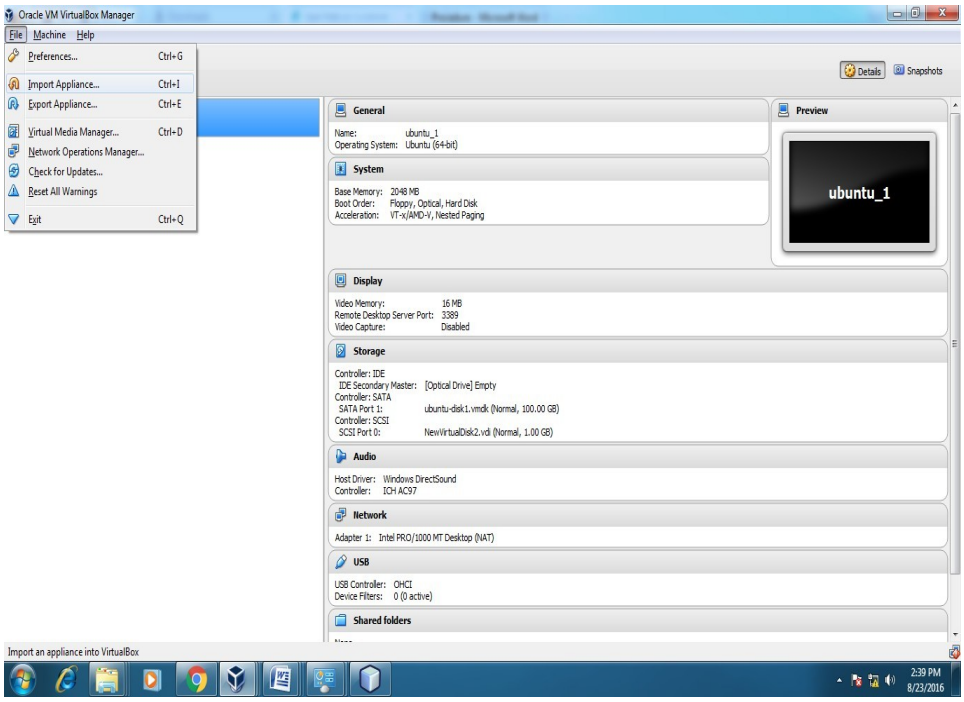
5. Click the install button…

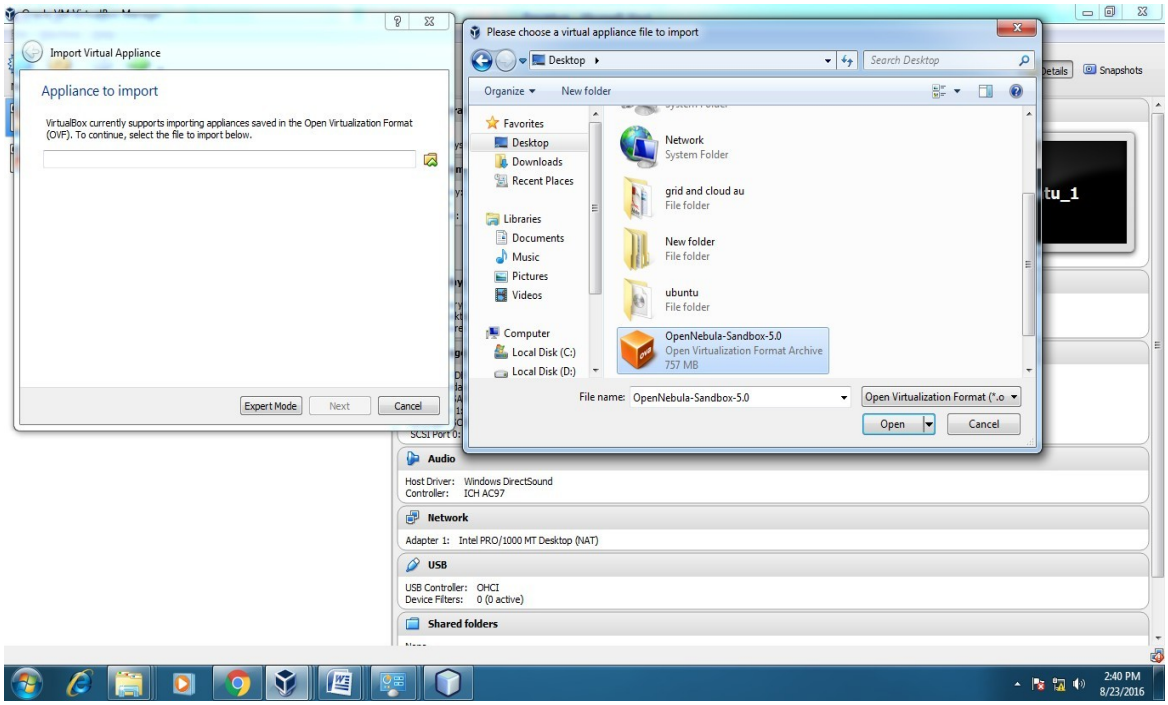


6. Then installation was completed..the show virtual box icon on desktop screen….

Steps to import Open nebula sandbox:

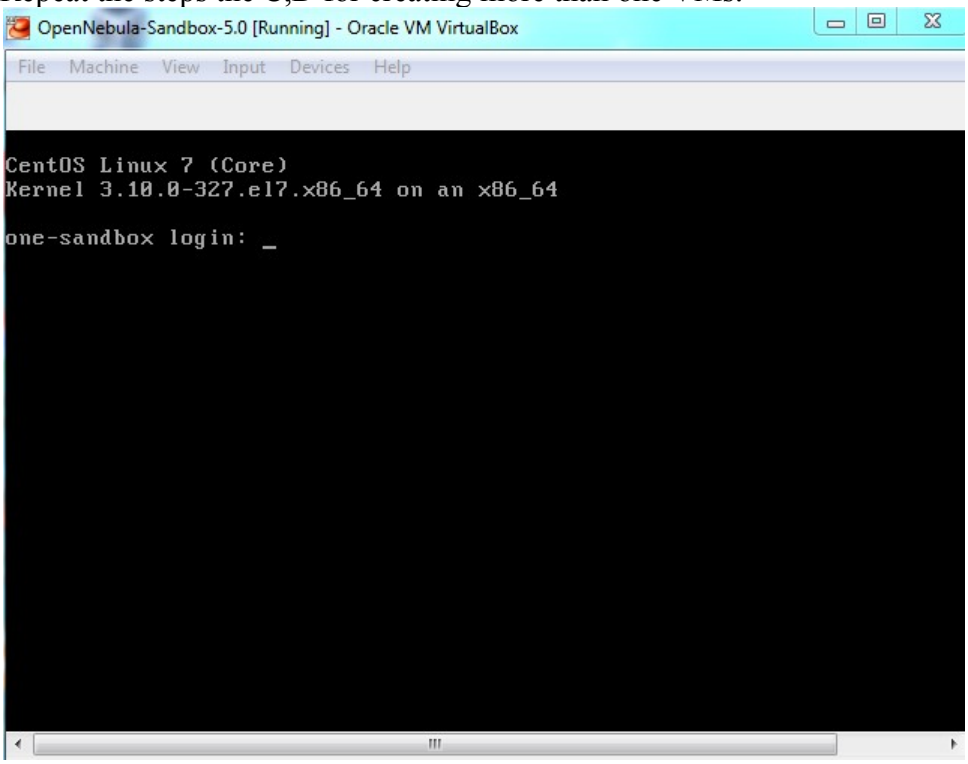
1. Open Virtual box
2. File ☾ import Appliance
3. Browse OpenNebula-Sandbox-5.0.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the Open Nebula
6. Login using username: root, password:opennebula

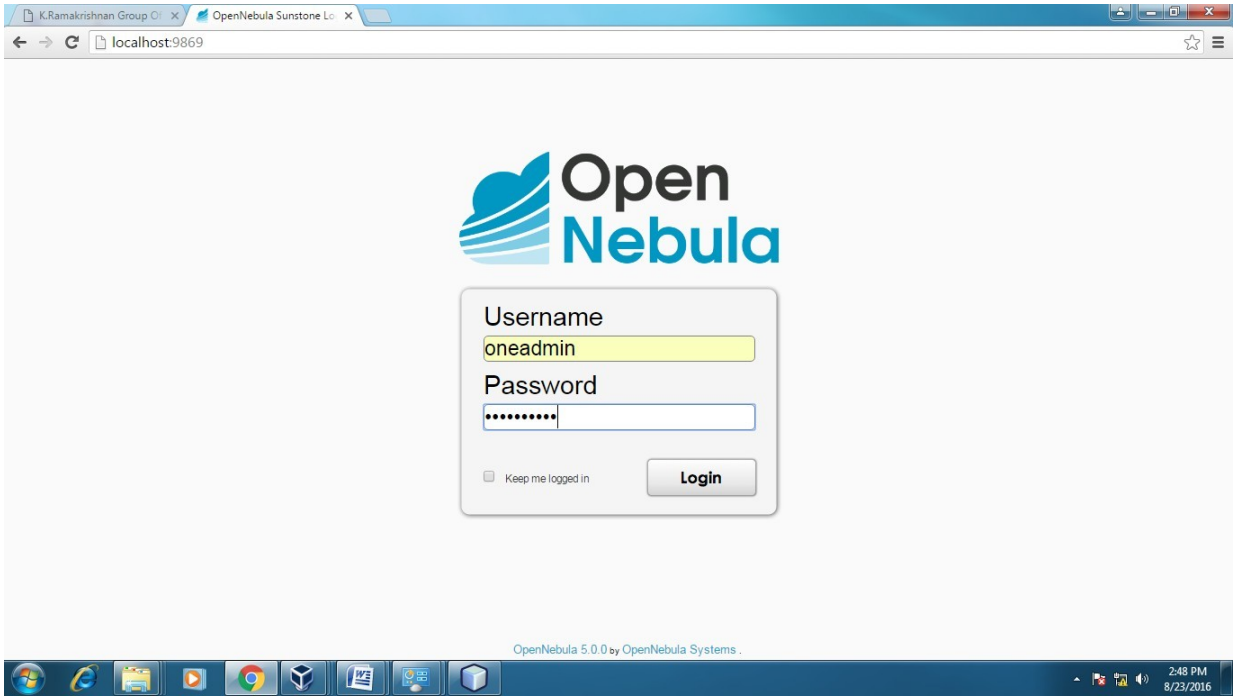


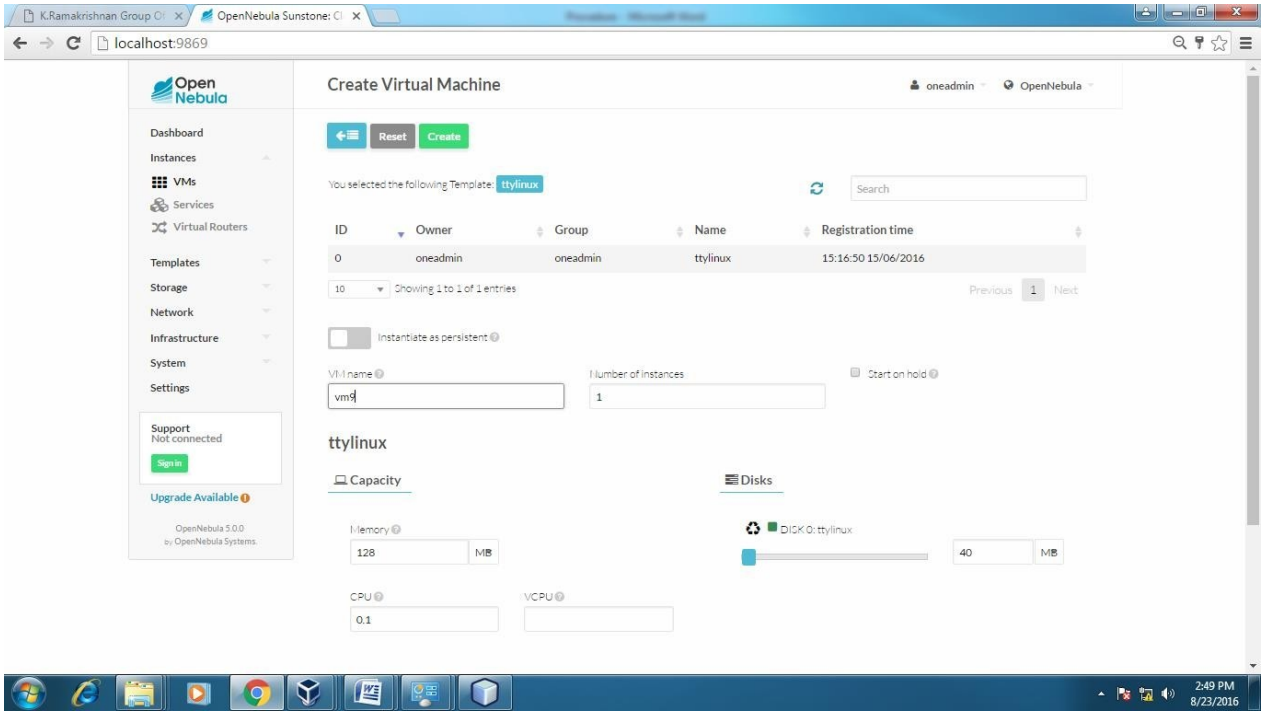


Steps to create Virtual Machine through opennebula

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.

EX.NO.:2 **Install**

a C compiler in the virtual machine created using

DATE: **virtual box and execute Simple Programs**

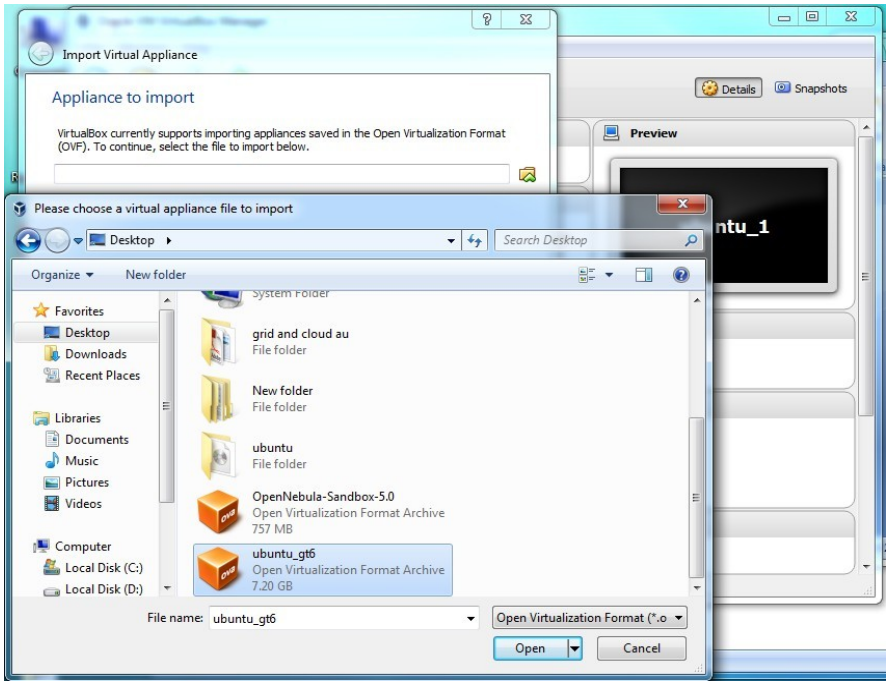
Aim:

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

PROCEDURE:

Steps to import .ova file:

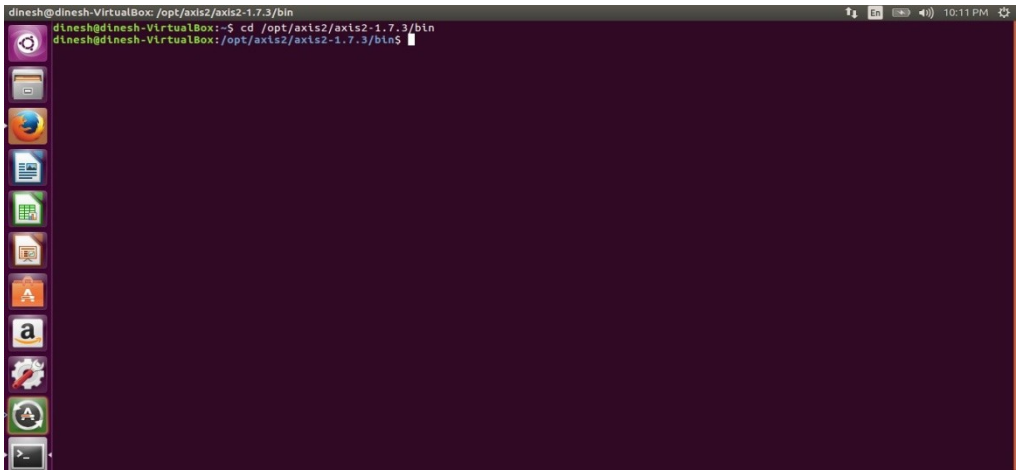
1. Open Virtual box
2. File ☾ import Appliance
3. Browse ubuntu_gt6.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the ubuntu_gt6
6. Login using username: dinesh, password:99425.



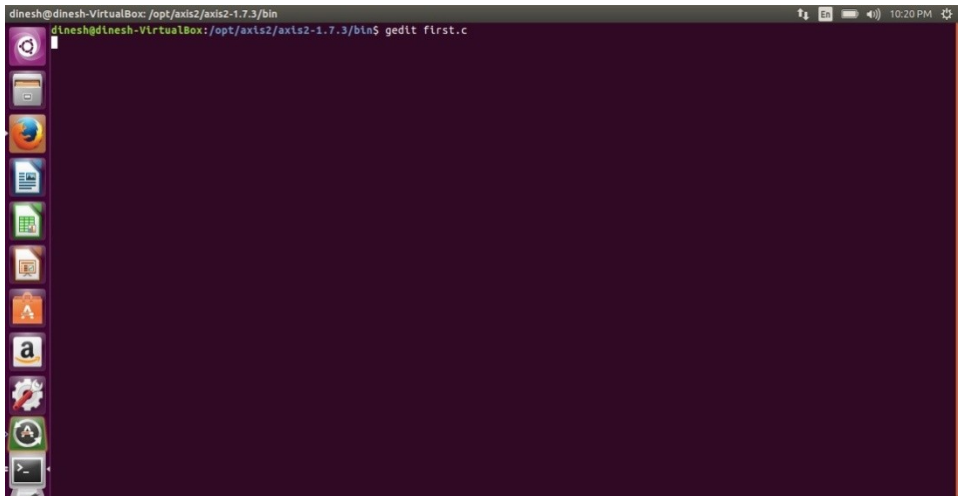
Steps to run c program:

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

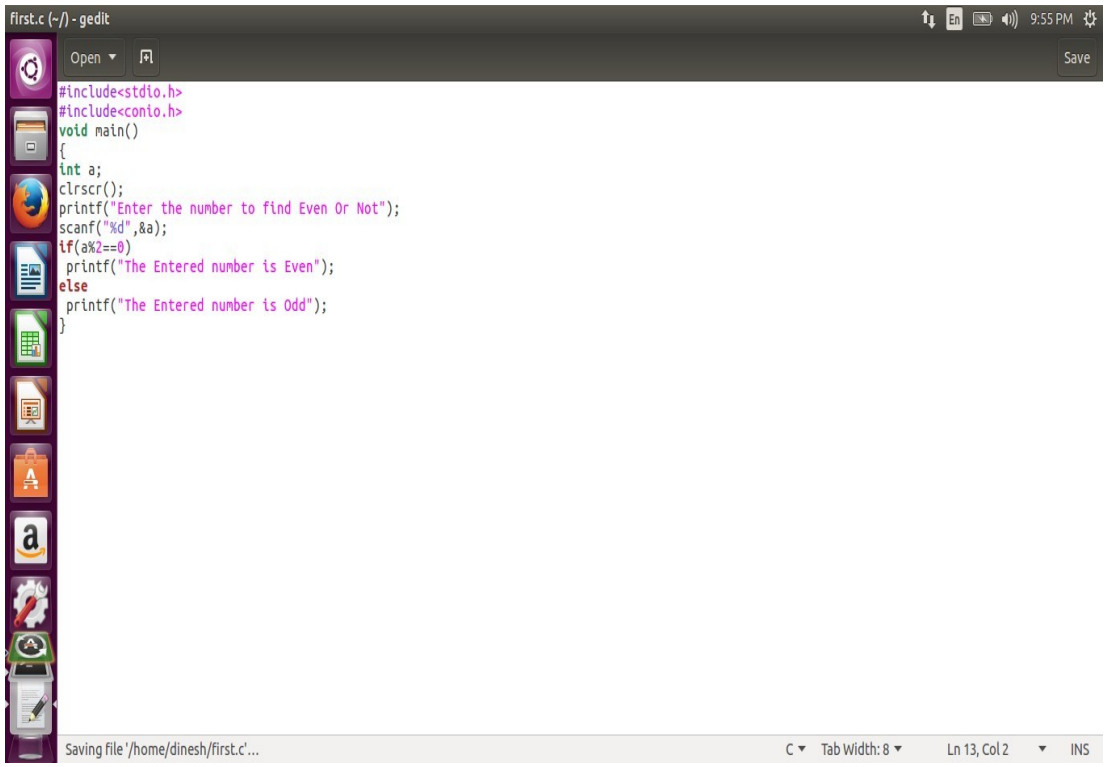
1. **Type cd /opt/axis2/axis2-1.7.3/bin then press enter**



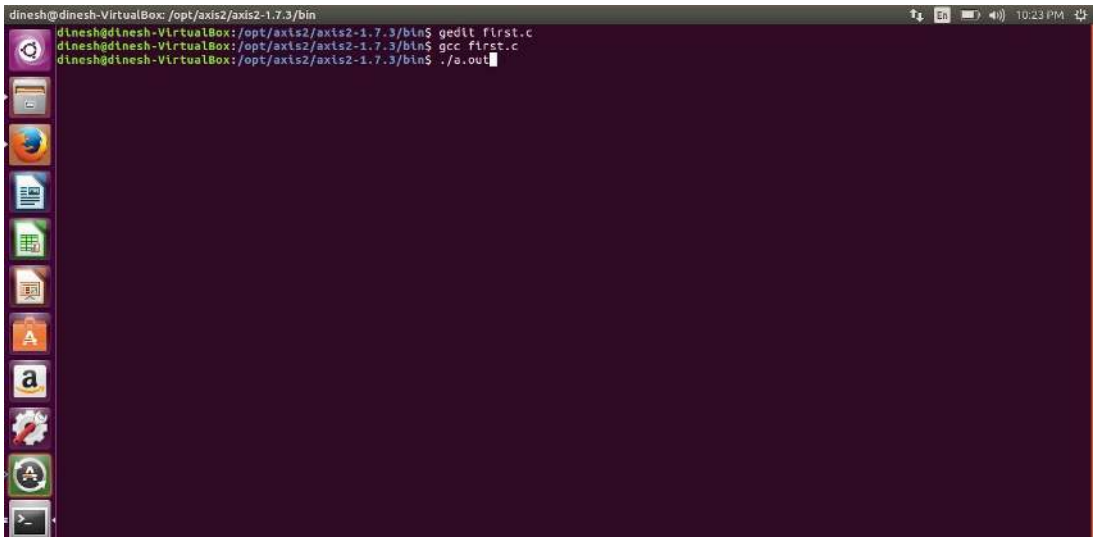
2. **Type gedit first.c**



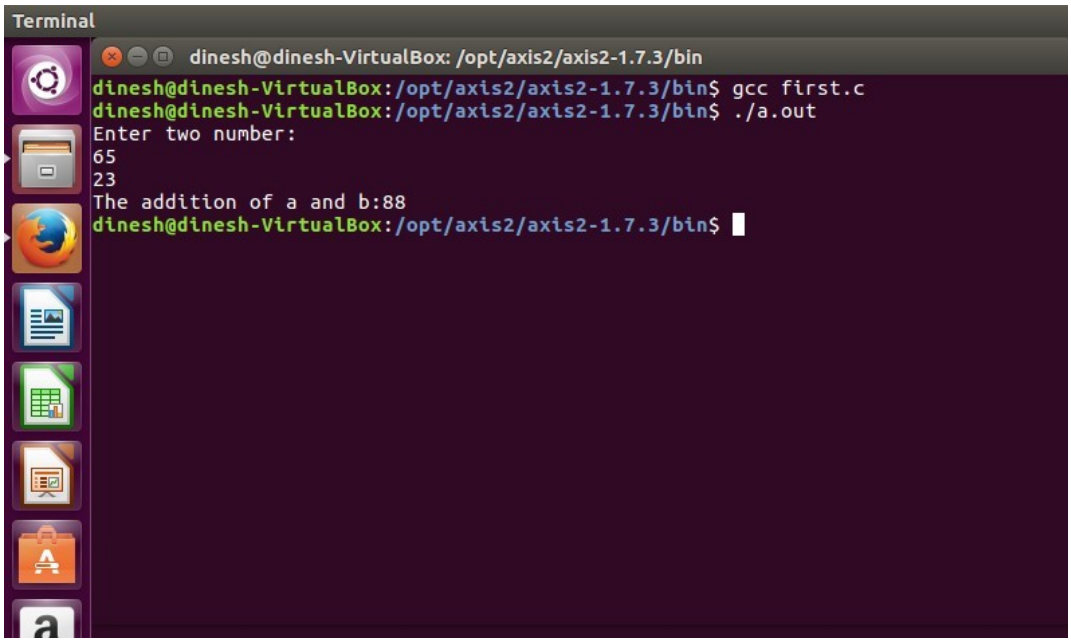
3. Type the c program



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.

EX NO.:3 **Install Google App Engine. Create *hello world* app and other simple**

DATE: **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

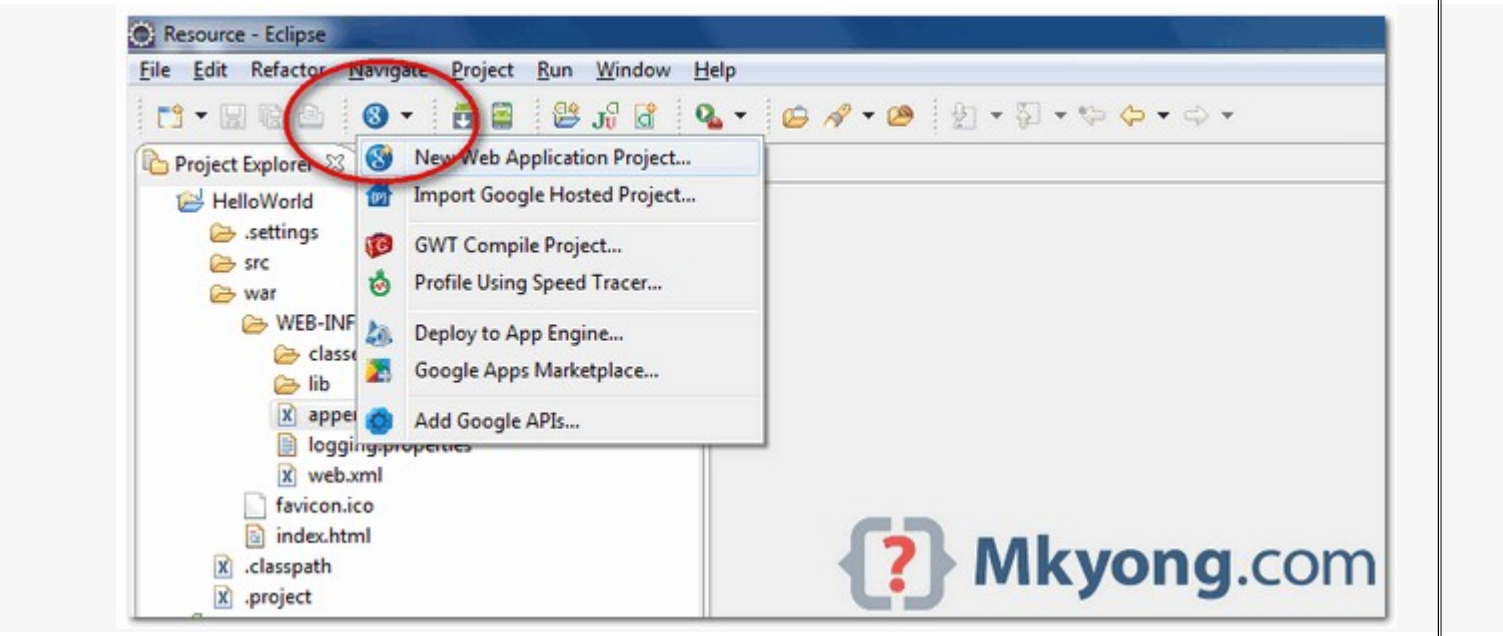
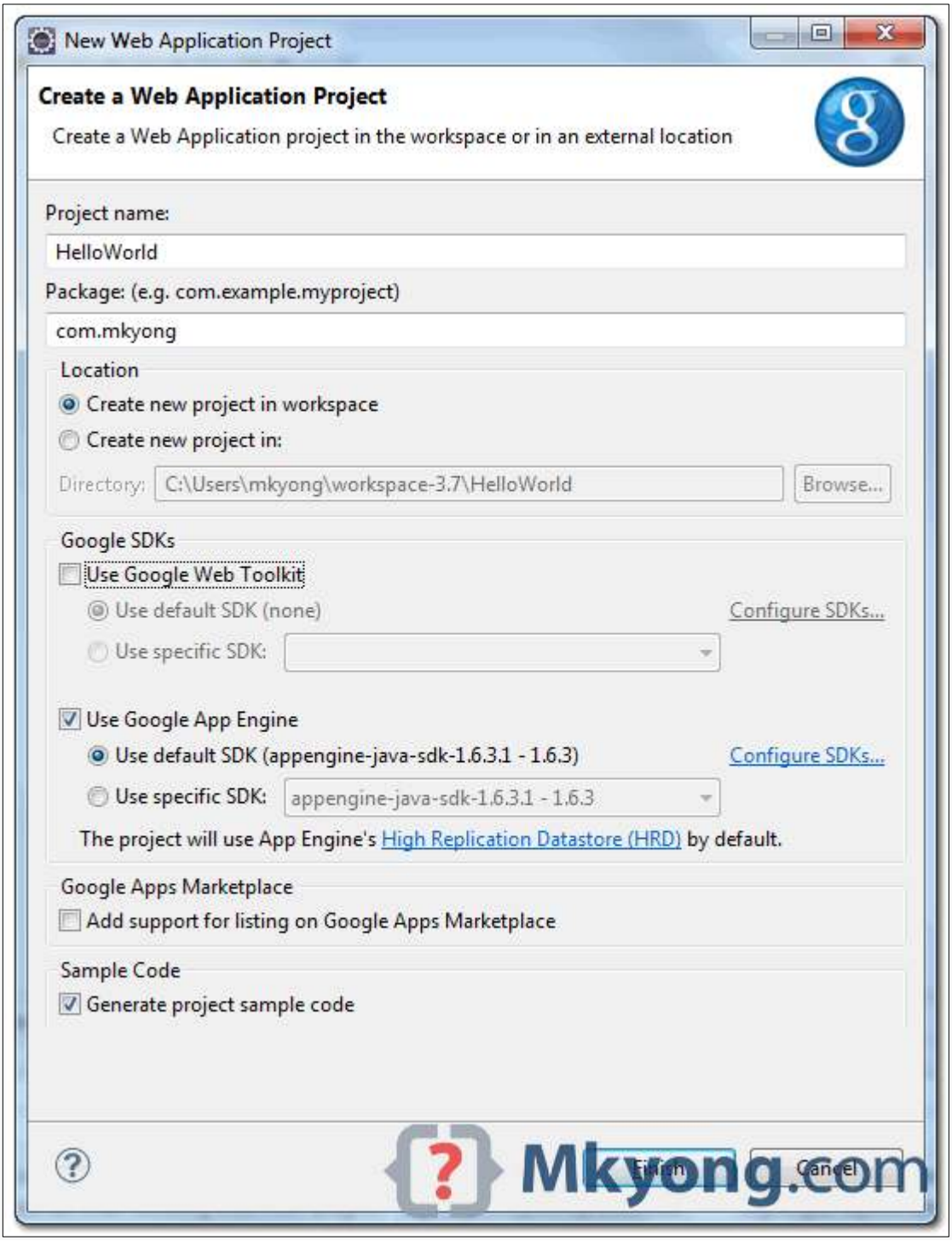


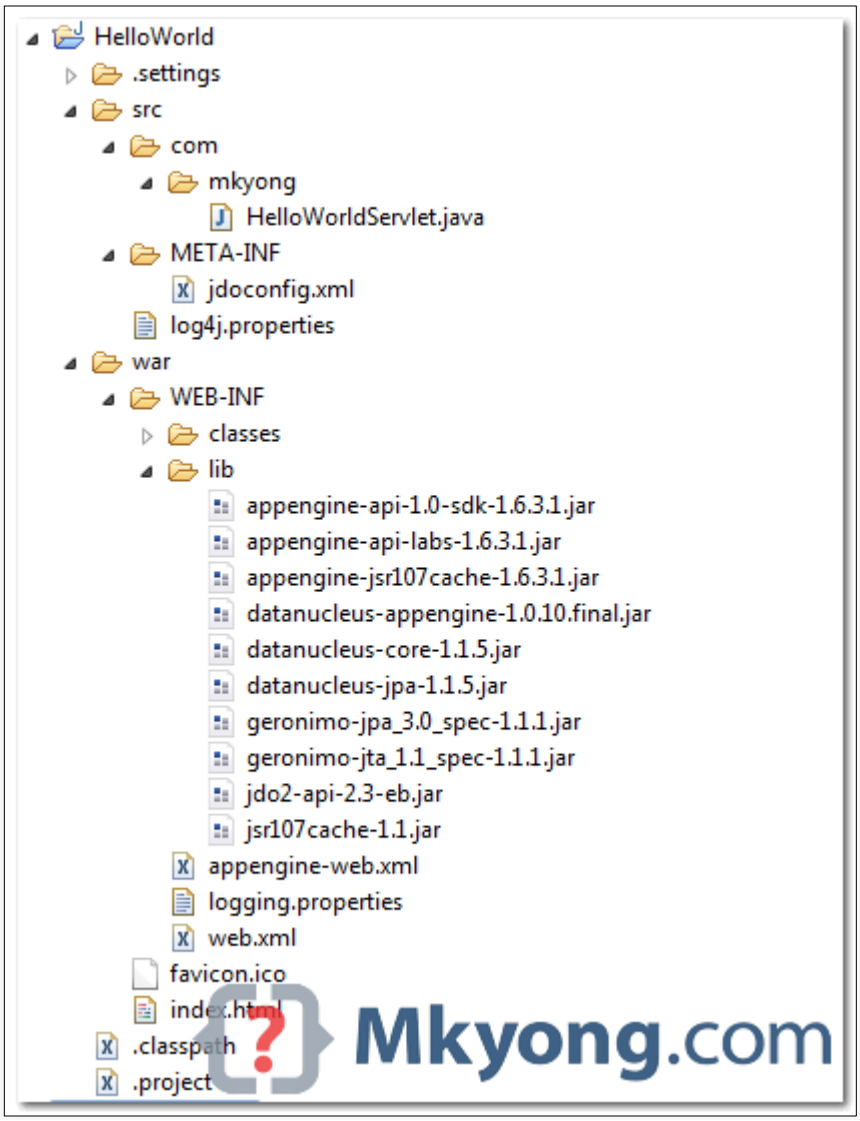
*Figure – Deselect the "**Google Web ToolKit**", and link your GAE Java SDK via the "**configure SDK**" link.*



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**".

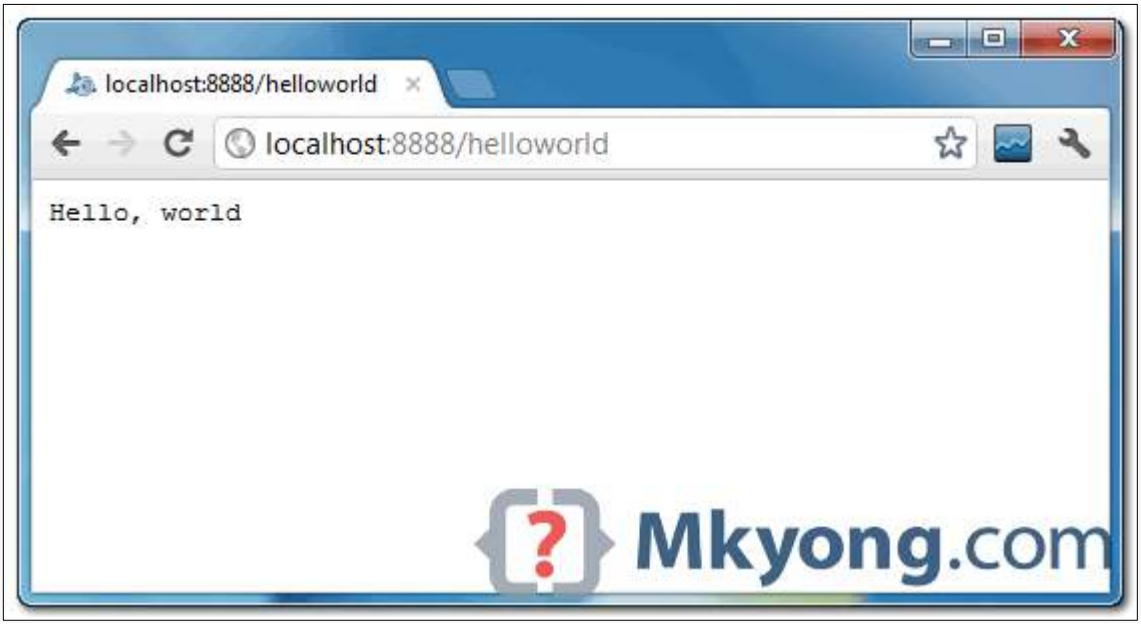
Eclipse console :

```
//...
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.

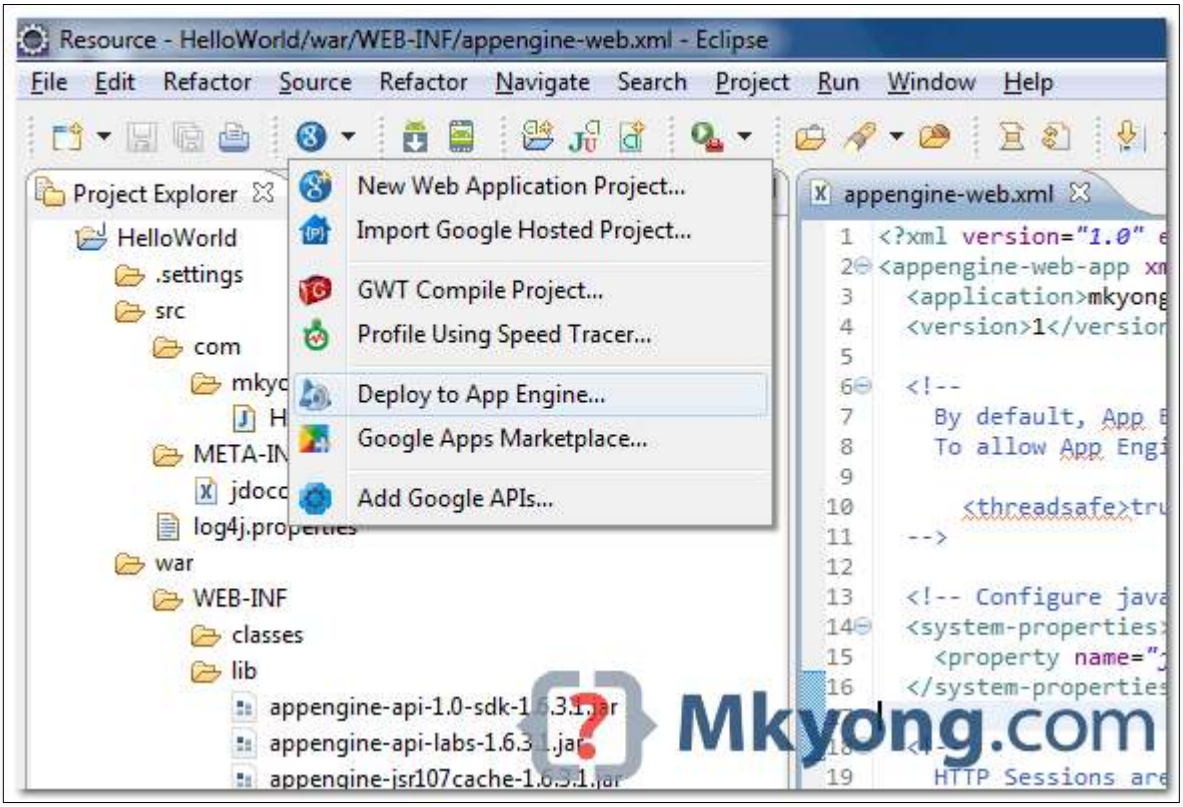


Figure 1.2 – Sign in with your Google account and click on the Deploy button.

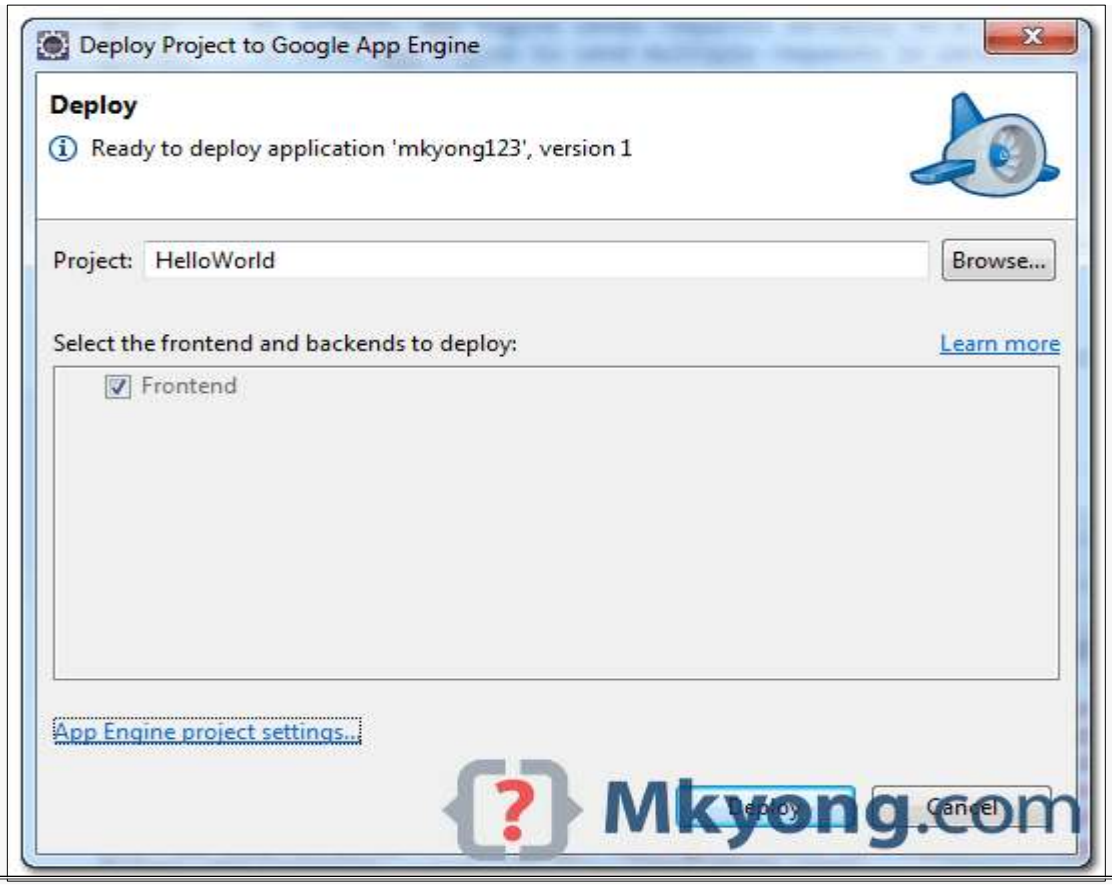


Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – http://mkyong123.appspot.com/

Result:

Thus the simple application was created successfully.

EX. NO.:4 **Simulate a cloud scenario using CloudSim and run a scheduling**

DATE: **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)
```

5. 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
```

6. The third step is to create a broker:

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

7. 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())
```

8. 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...
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>null
Broker is
starting... Entities
started.
: Broker: Cloud Resource List received with 1
resource(s) 0.0: Broker: Trying to Create VM #0 in
Datacenter_0
: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
: Broker: All Cloudlets executed.
Finishing.......400.1: Broker: Destroying
VM #0
Broker is shutting down...
Simulation: No more future events
Cloud Information Service: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down
Simulation completed.
Simulation completed.

========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	400	400.1	0.1

*****Datacenter:
Datacenter_0***** User id
Debt
3 35.6

CloudSimExample1 finished!

RESULT:

The simulation was successfully executed.

EX.NO.:5 **Use GAE launcher to launch the web applications.**
DATE:

Aim:

To Use GAE launcher to launch the web applications.

Steps:

Making your First Application

Now you need to create a simple application. We could use the "+"option to have the launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.

Make a folder for your Google App Engine applications. I am going to make the Folder on my Desktop called "**apps**" – the path to 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 would be:

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:

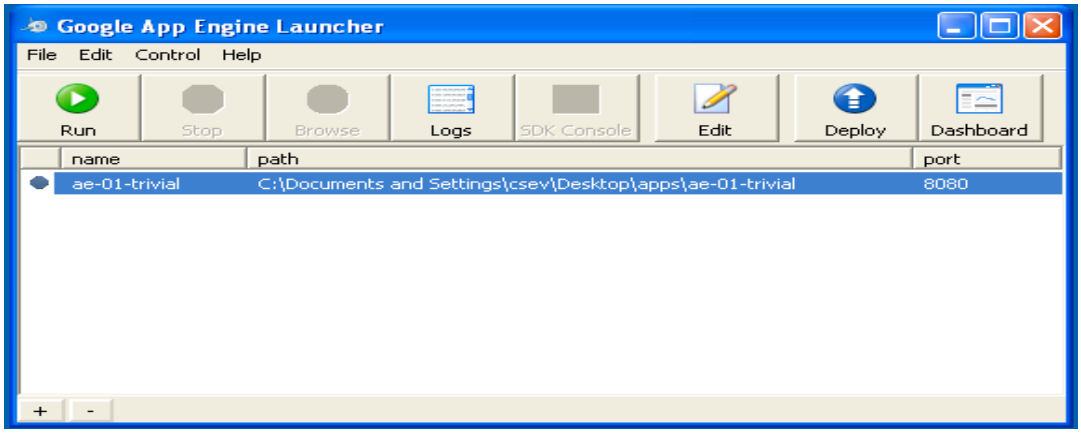
```
application: ae-01-trivial
version: 1
runtime: python api_version: 1
handlers:- url: /.*
script: index.py
```

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 you have selected your application and press Run. After a few moments your application will start and the launcher will show 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:

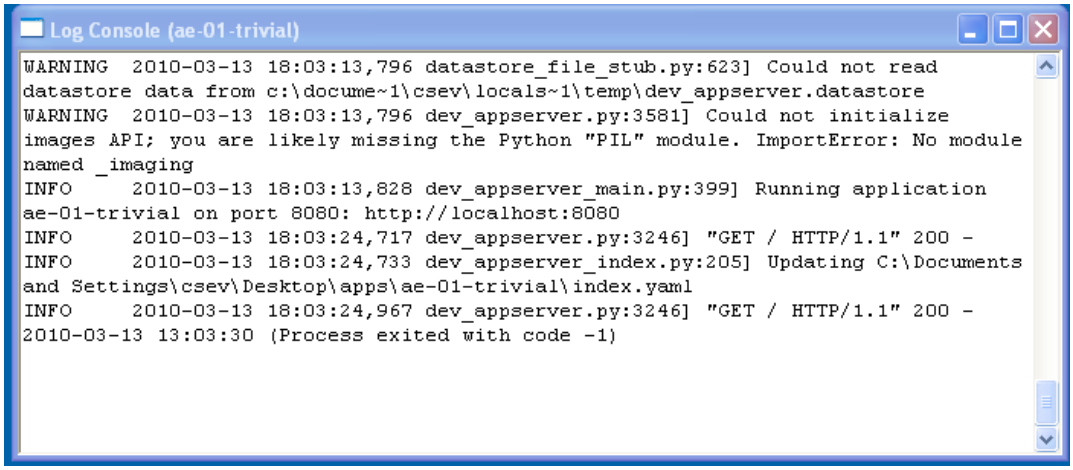
Just for fun, edit the index.pytochange the name "Chuck" to you row

nname and press Refresh in the browser to verify your updates.

Watching the Log

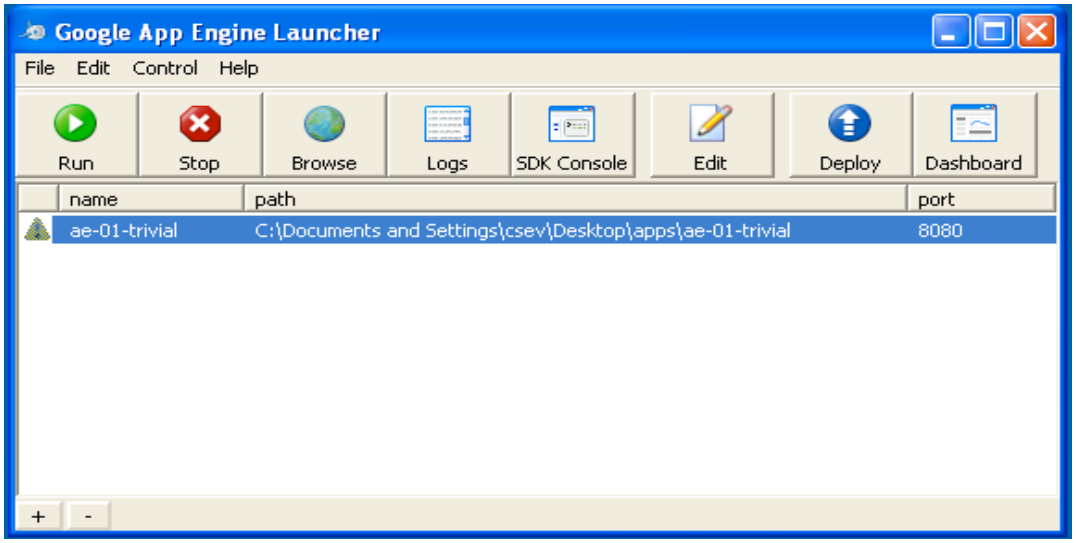
You can watch the internal log of the actions that the web server is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the Logs button to bring up a log window:

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

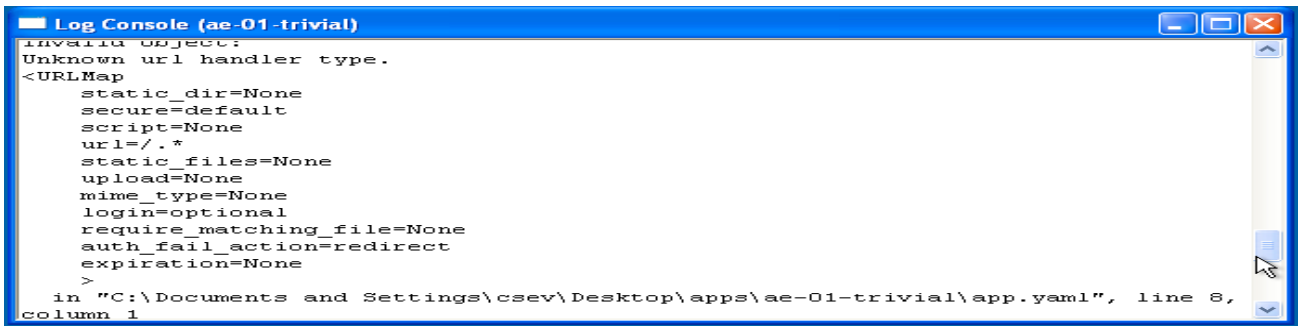


Dealing With Errors

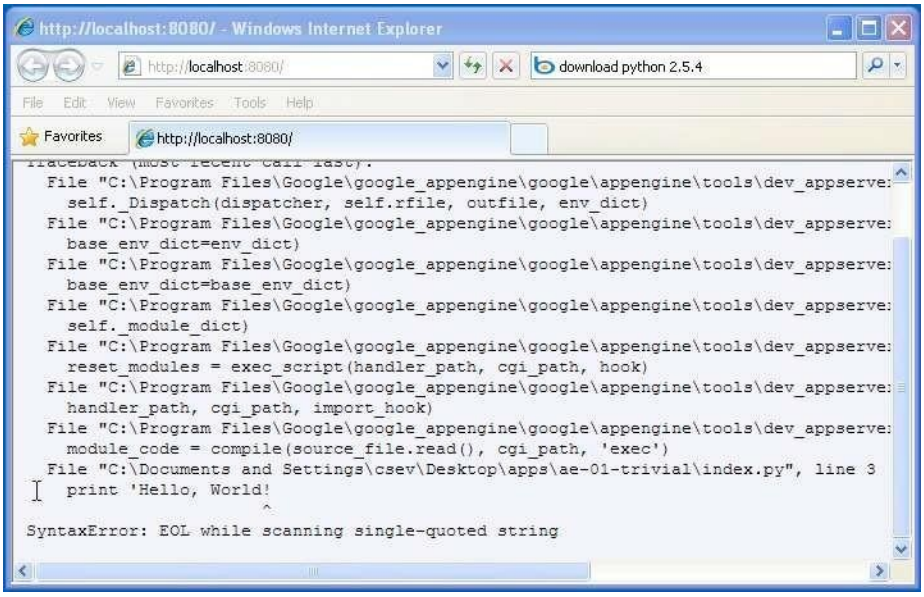
With two files to edit, there are two general categories of errors that you may encounter. If youmake a mistake onthe app.yamlfile, the App Engine willnotstart and your launcher will show a yellow icon near your application:



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).
Ifyoumake asyntaxerror in the index.pyfile, a Pythontrace backerrorwillappear in yourbrowser.



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.

EX.NO:6 **Find a procedure to transfer the files from one virtual machine to another virtual machine.**

DATE:

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!

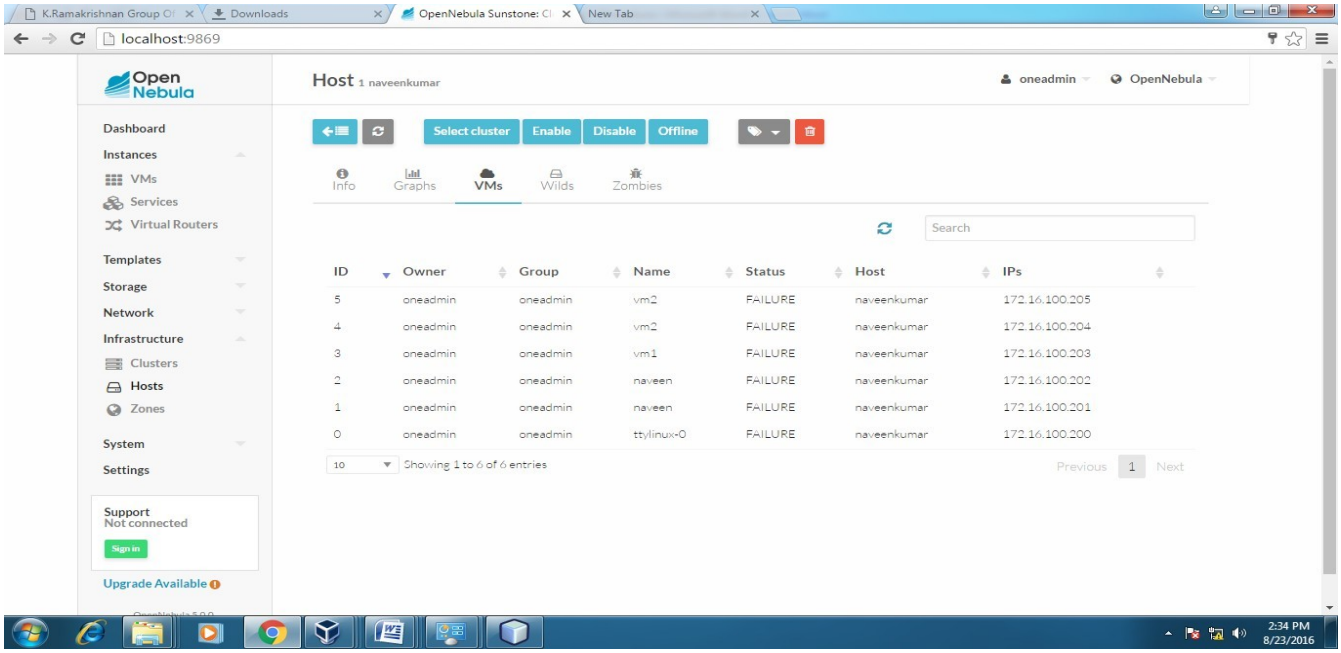
PROCEDURE:

Steps:

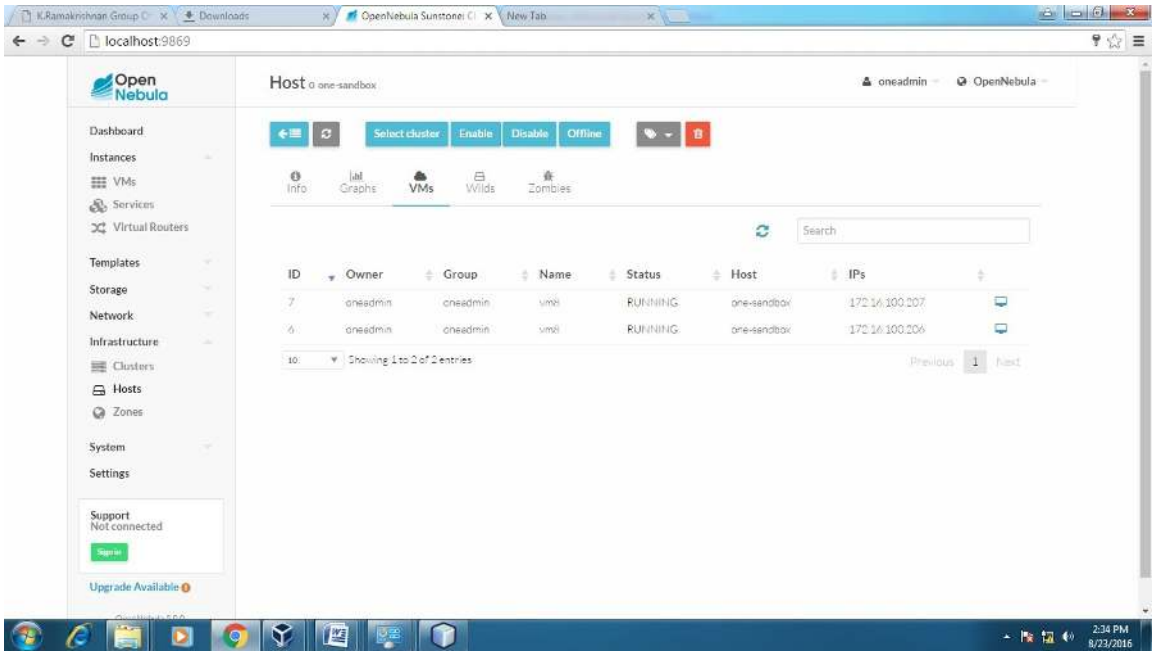
1. Open Browser, type localhost:9869
2. Login using username: oneadmin, password: opennebula
3. Then follow the steps to migrate VMs
   a. Click on infrastructure
   b. Select clusters and enter the cluster name
   c. Then select host tab, and select all host
   d. Then select Vnets tab, and select all 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 8th 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.

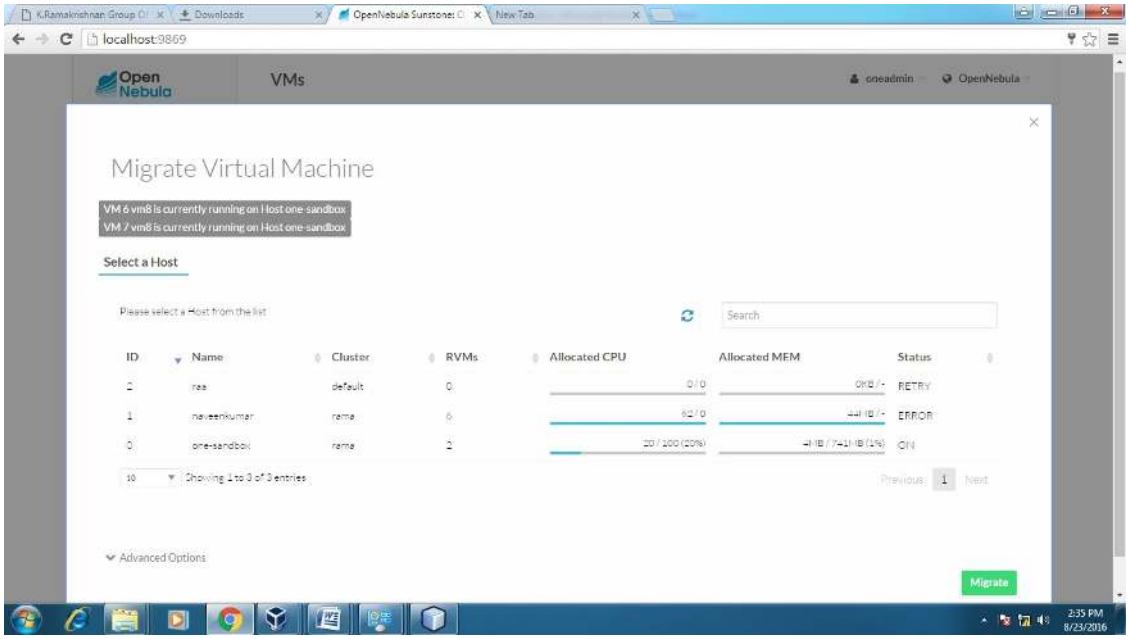
Before migration

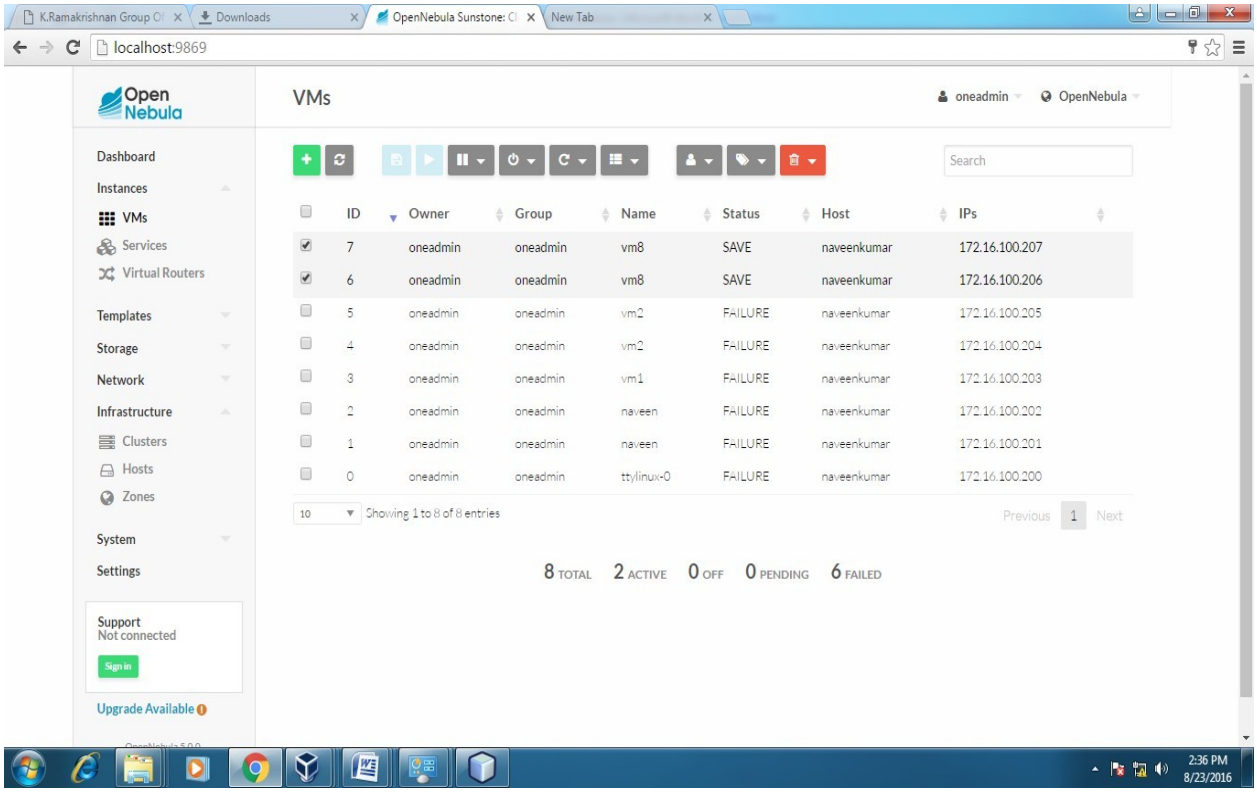
Host:SACET



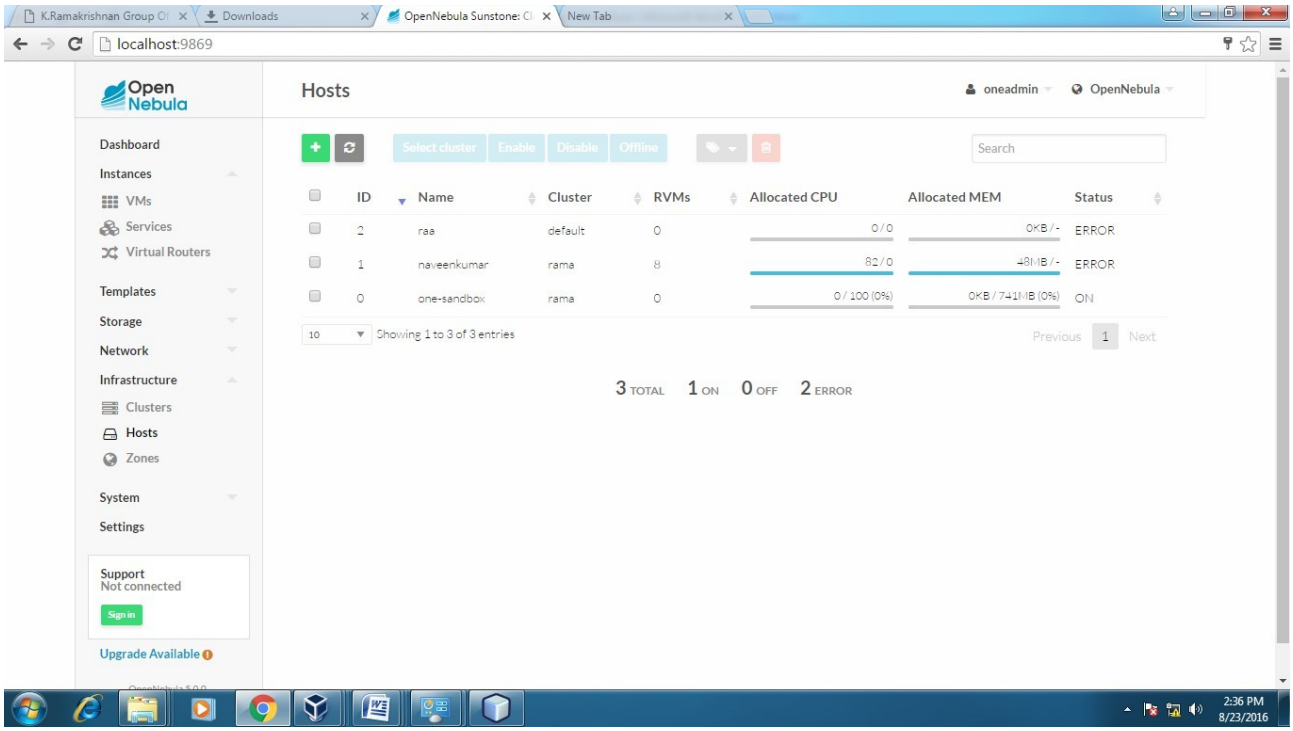
Host:one-sandbox



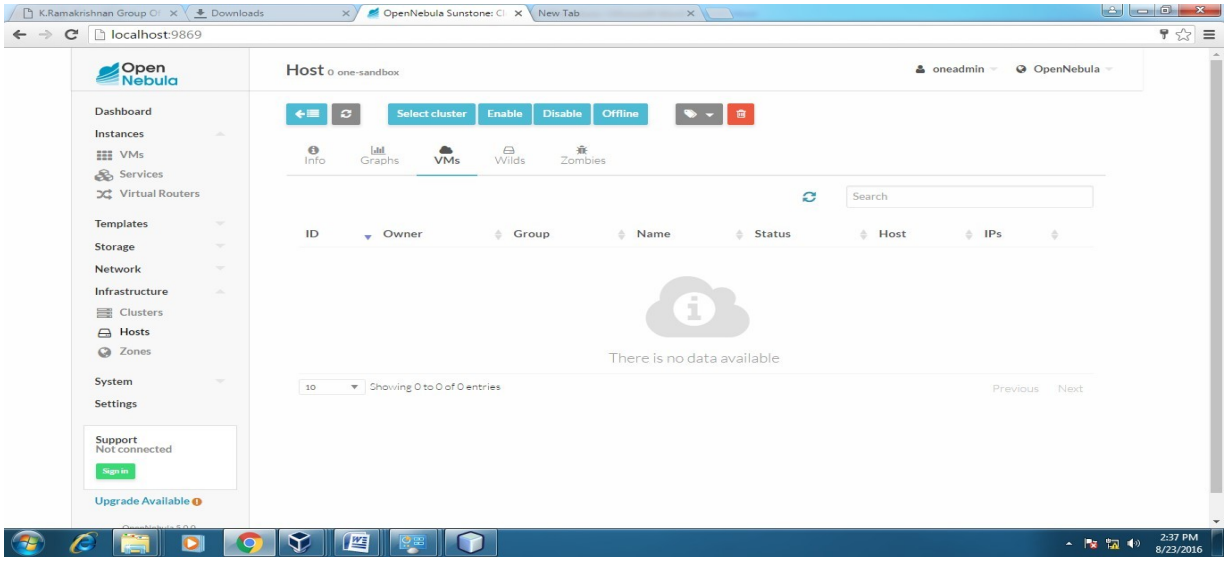




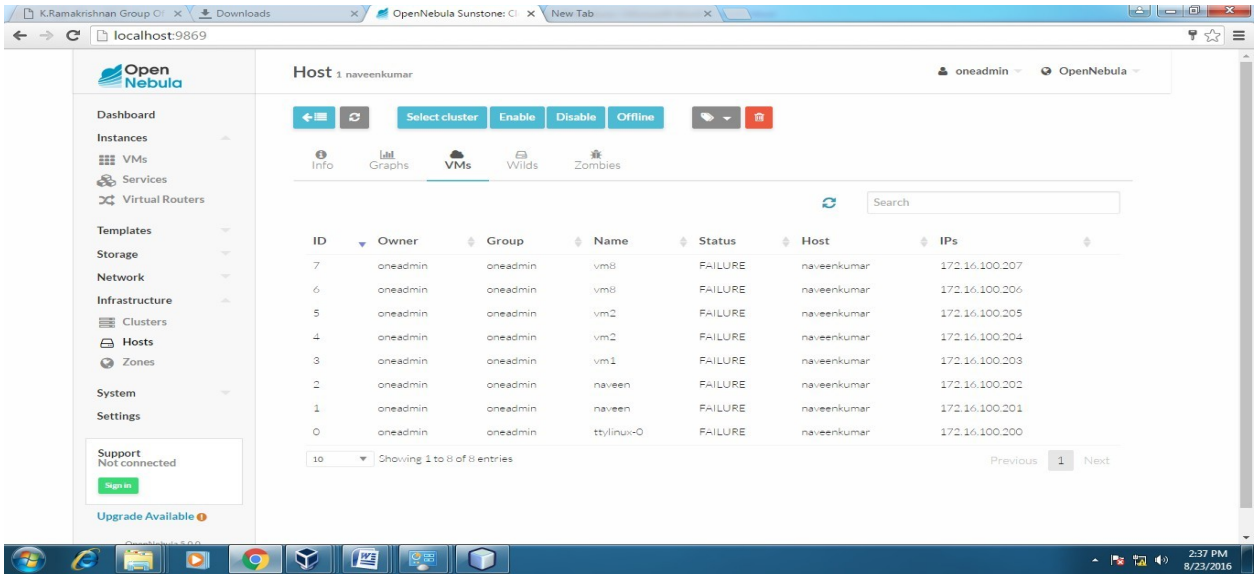
After Migration:



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…..

EX NO.:8 **Install Hadoop single node cluster and run simple applications like wordcount.**

DATE :

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 your home directory.

Step 2: Extract the Java Tar File.

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

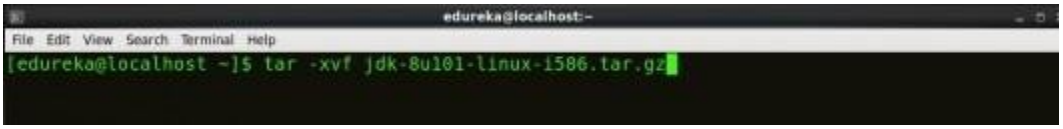


Fig: Hadoop Installation – Extracting Java Files

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

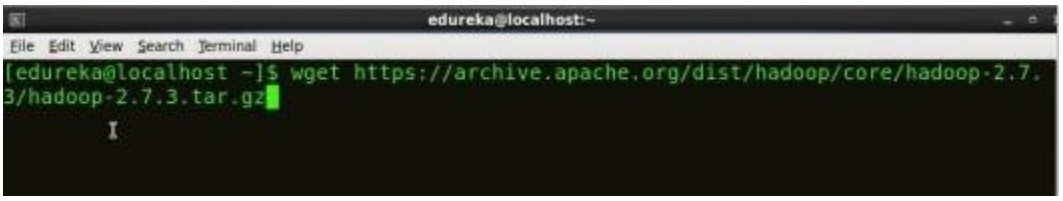


Fig: Hadoop Installation – Downloading Hadoop

Step 4: Extract the Hadoop tar File.

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

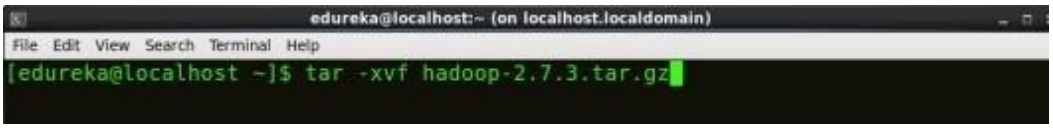
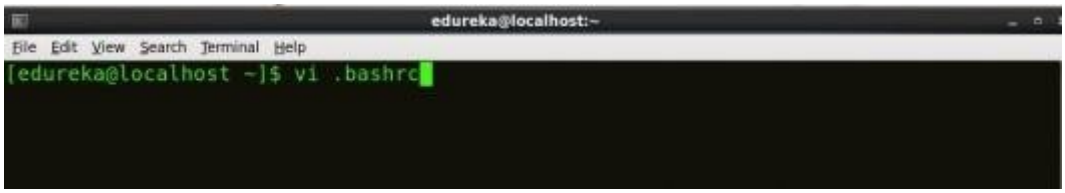


Fig: Hadoop Installation – Extracting Hadoop Files

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



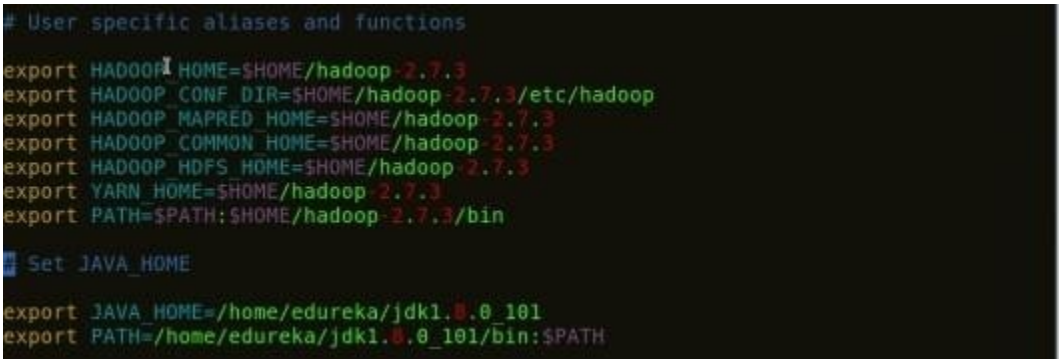


Fig: Hadoop Installation – Setting Environment Variable

Then, save the bash file and close it.

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

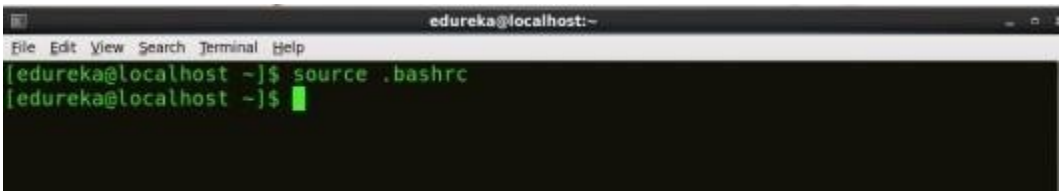
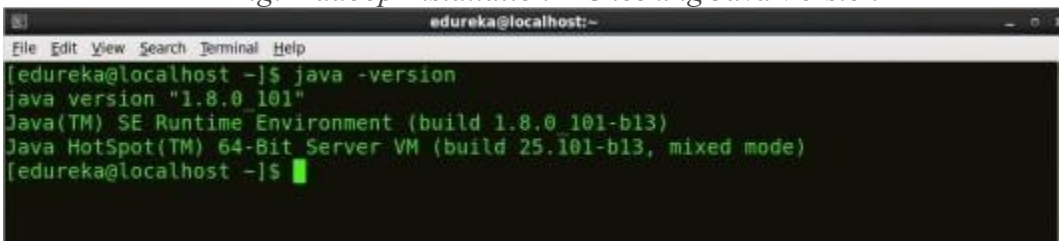


Fig: Hadoop Installation – Refreshing environment variables

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

Fig: Hadoop Installation – Checking Java Version



***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

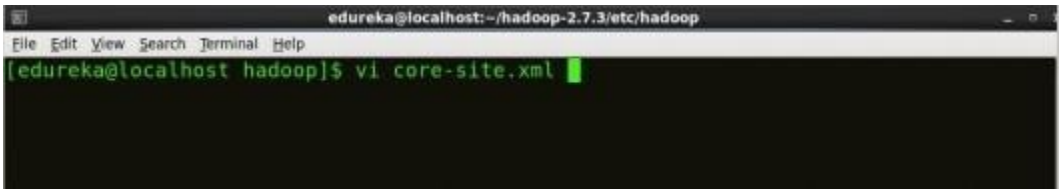
All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you can see in the snapshot below:

```
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]$
```

Fig: Hadoop Installation – Hadoop Configuration Files

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



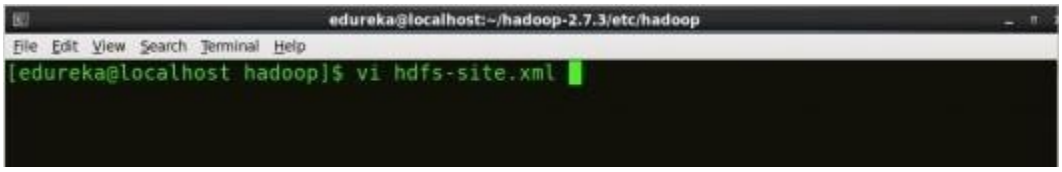

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

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

Fig: Hadoop Installation – Configuring core-site.xml

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




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

Fig: Hadoop Installation – Configuring hdfs-site.xml

```
<?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>
```

```
9                <name>dfs.permission</name>
10                 <value>false</value>
11               </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.

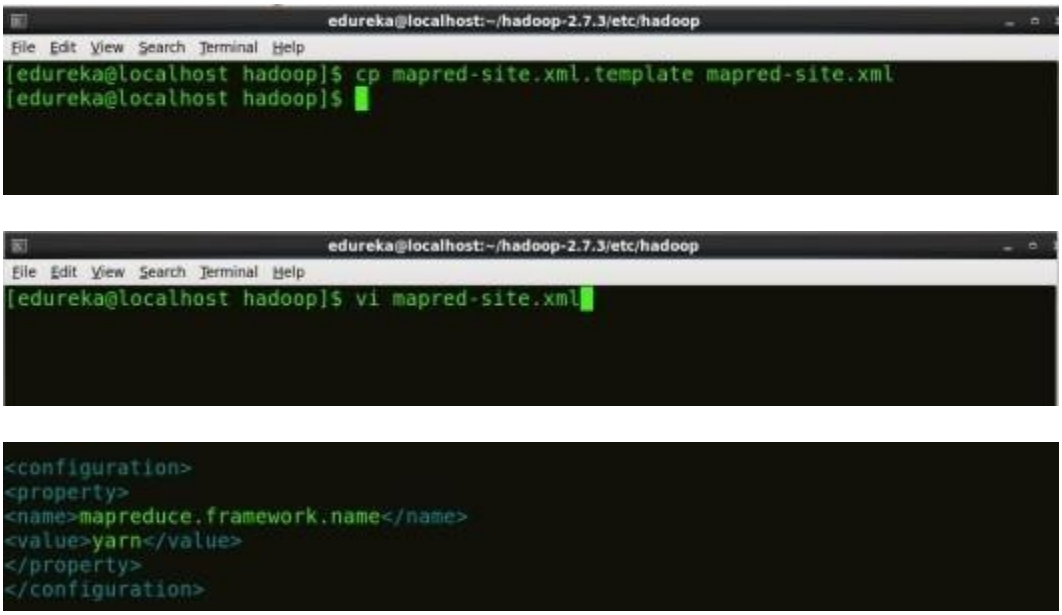


Fig: Hadoop Installation – Configuring mapred-site.xml

```
<?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

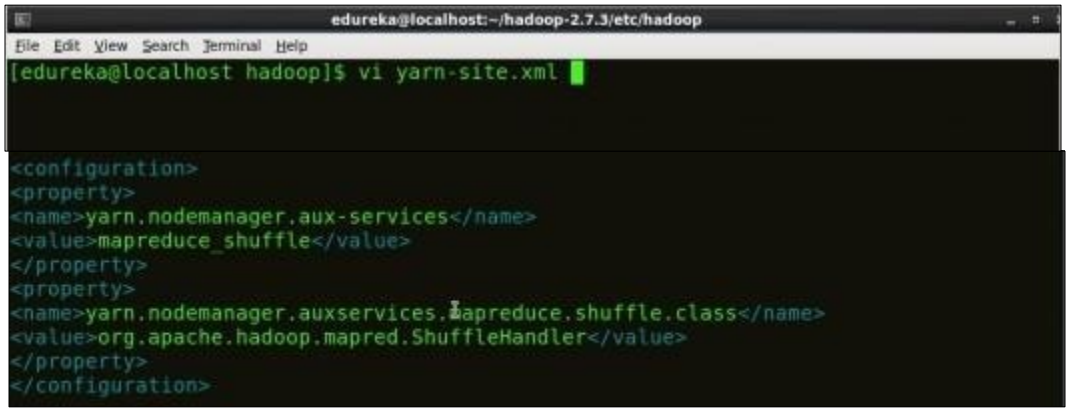


Fig: Hadoop Installation – Configuring yarn-site.xml

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

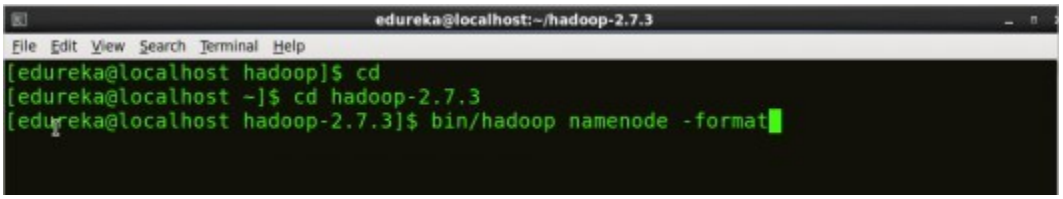


Fig: Hadoop Installation – Formatting NameNode

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

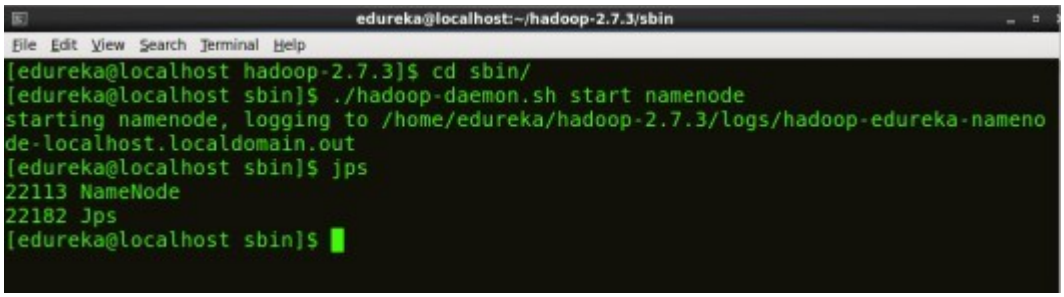


Fig: Hadoop Installation – Starting NameNode

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

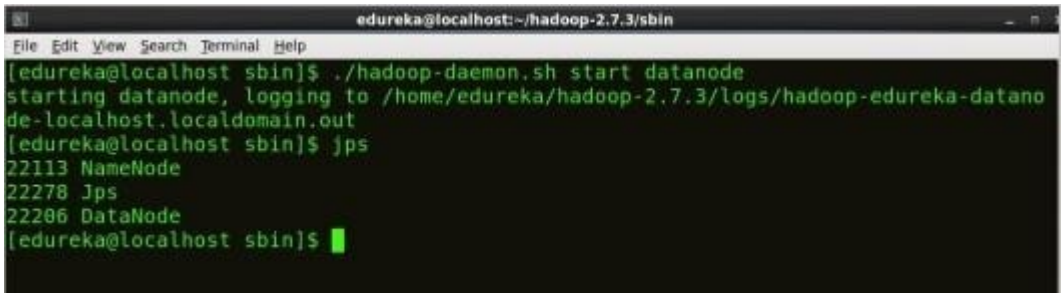


Fig: Hadoop Installation – Starting DataNode

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

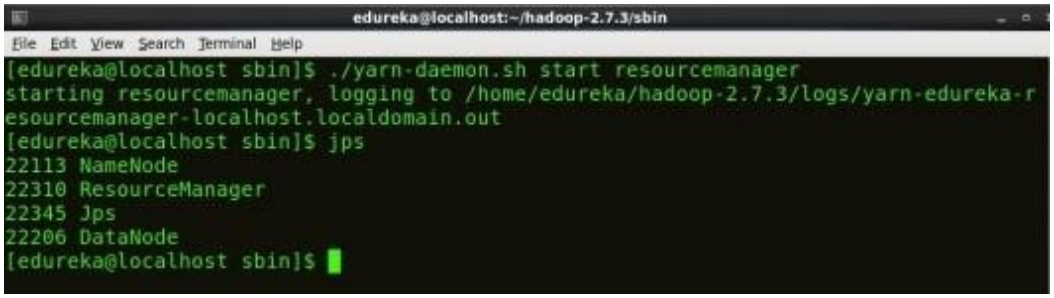
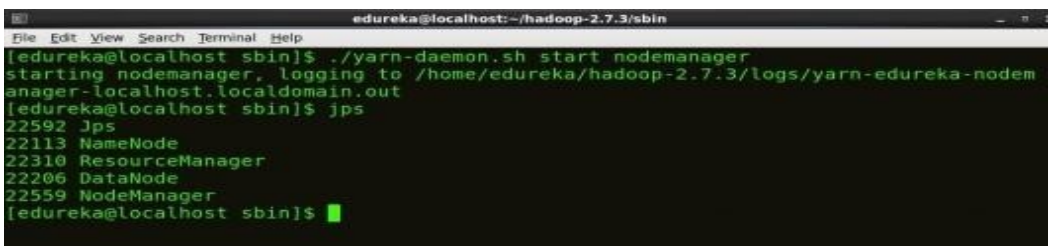


Fig: Hadoop Installation – Starting ResourceManager

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



See Batch Details

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

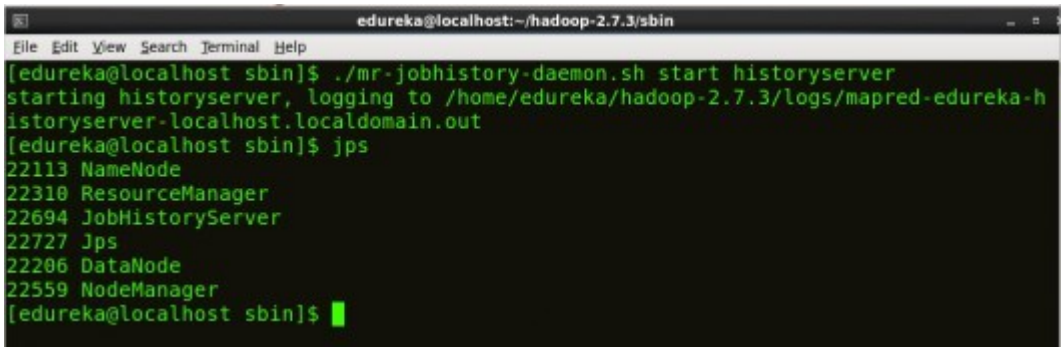


Fig: Hadoop Installation – Checking Daemons

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

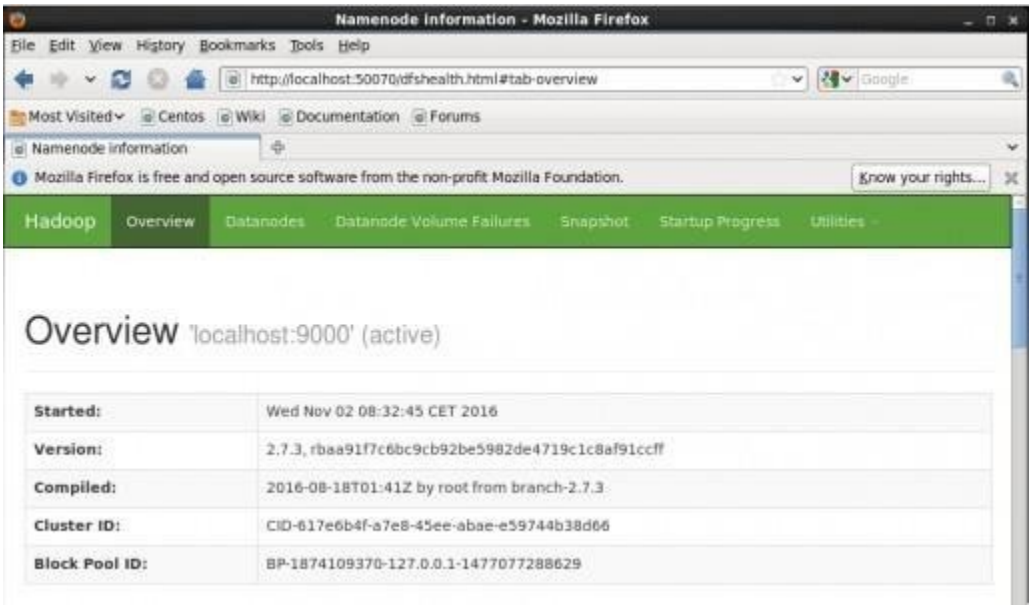


Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster

Result:

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

Ex. No:9 ***Run a Container from Docker Hub***

AIM:

To write a program to run a container from Docker hub.

PROCEDURE:

Run a container from docker hub Run

docker -h,

$ docker -h
Flag shorthand -h has been deprecated, please use --help Usage:

docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

...

Management Commands:
builder Manage builds
config Manage Docker configs
container Manage containers engine
Manage the docker engine
image Manage images
network Manage networks
node Manage Swarm nodes
plugin Manage plugins
secret Manage Docker secrets
service Manage services
stack Manage Docker stacks
swarm Manage Swarm
system Manage Docker
trust Manage trust on Docker images
volume Manage volumes

The Docker command line can be used to manage several features of the Docker Engine. In this lab, we will mainly focus on the container command.

If podman is installed, you can run the alternative command for comparison. sudo

podman -h

You can additionally review the version of your Docker installation,

docker version

Client:
Version: 19.03.6
...

Server: Docker Engine - Community
Engine
Version: 19.03.5
...

sudo podman version --events-backend=none
Version: 2.1.1
API Version: 2.0.0
Go Version: go1.15.2
Built: Thu Jan 1 00:00:00 1970
OS/Arch: linux/amd64

Step 1: Run your first container

We are going to use the Docker CLI to run our first container. Open

a terminal on your local computer

Run docker container run -t ubuntu top

Use the docker container run command to run a container with the ubuntu image using the top command. The -t flags allocate a pseudo-TTY which we need for the top to work correctly.

$ docker container run -it ubuntu top Unable to find image 'ubuntu:latest' locally latest: Pulling from library/ubuntu aafe6b5e13de: Pull complete 0a2b43a72660: Pull complete 18bdd1e546d2: Pull complete 8198342c3e05: Pull complete f56970a44fd4: Pull complete Digest: sha256:f3a61450ae43896c4332bda5e78b453f4a93179045f20c8181043b26b5e79028 Status: Downloaded newer image for ubuntu:latest

The docker run command will result first in a docker pull to download the ubuntu image onto your host. Once it is downloaded, it will start the container. The output for the running container should look like this:

top - 20:32:46 up 3 days, 17:40, 0 users, load average: 0.00, 0.01, 0.00
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 2046768 total, 173308 free, 117248 used, 1756212 buff/cache
KiB Swap: 1048572 total, 1048572 free, 0 used. 1548356 avail Mem

```
  PID USER    PR  NI    VIRT    RES    SHR S  %CPU %MEM    TIME+
    COMMAND 1 root 20   0   36636   3072   2640 R   0.3  0.2   0:00.04 top
```

Inspect the container with docker container exec

The docker container exec command is a way to "enter" a running container's namespaces with a new process.

Open a new terminal. On cognitiveclass.ai, select Terminal > New Terminal.

Using play-with-docker.com, to open a new terminal connected to node1, click "Add New Instance" on the lefthand side, then ssh from node2 into node1 using the IP that is listed by 'node1 '. For example:

```
[node2] (local) root@192.168.0.17 ~
$ ssh 192.168.0.18
[node1] (local) root@192.168.0.18 ~
$
```

In the new terminal, use the docker container ls command to get the ID of the running container you just created.

```
$ docker container ls
CONTAINER ID    IMAGE          COMMAND            CREATED         STATUS                PORTS
NAMES
b3ad2a23fab3        ubuntu              "top"               29 minutes ago      Up 29 minutes
goofy_nobel
$ docker container exec -it b3ad2a23fab3 bash
root@b3ad2a23fab3:/#
```

And Voila! We just used the docker container exec command to "enter" our container's namespaces with our bash process. Using docker container exec with bash is a common pattern to inspect a docker container.

Notice the change in the prefix of your terminal. e.g. root@b3ad2a23fab3:/. This is an indication that we are running bash "inside" of our container.

From the same termina, run ps -ef to inspect the running processes.

```
root@b3ad2a23fab3:/# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 20:34 ?        00:00:00 top
root        17     0  0 21:06 ?        00:00:00 bash
root        27    17  0 21:14 ?        00:00:00 ps -ef
```

You should see only the top process, bash process and our ps process.

```
root@b3ad2a23fab3:/# exit
exit
$ ps -ef
```

```
# Lots of processes!
docker ps -a

docker rm <CONTAINER ID>
```

Step 2: Run Multiple Containers

Explore the Docker Hub

The Docker Hub is the public central registry for Docker images, which contains community and official images.

Run an Nginx server

Let's run a container using the official Nginx image from the Docker Hub.

```
$ docker container run --detach --publish 8080:80 --name nginx nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
36a46ebd5019: Pull complete 57168433389f:
Pull complete 332ec8285c50: Pull complete
Digest: sha256:c15f1fb8fd55c60c72f940a76da76a5fccce2fefa0dd9b17967b9e40b0355316 Status:
Downloaded newer image for nginx:latest

5e1bf0e6b926bd73a66f98b3cbe23d04189c16a43d55dd46b8486359f6fdf048 Nginx is a
```

lightweight web server. You can access it on port 8080 on your localhost.

Access the nginx server on localhost:8080. curl

localhost:8080

will return the HTML home page of Nginx,

```
<!DOCTYPE html>
<html>
<head>


<title>Welcome to nginx!</title>
<style>
  body
  {
    width: 35em; margin:
    0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }

</style>
</head>
<body>
```

<h1>Welcome to nginx!</h1>

If you are using play-with-docker, look for the 8080 link near the top of the page, or if you run a Docker client with access to a local browser,

Run a mongo DB server

Now, run a mongoDB server. We will use the official mongoDB image from the Docker Hub. Instead of using the latest tag (which is the default if no tag is specified), we will use a specific version of the mongo image.

```
$ docker container run --detach --publish 8081:27017 --name mongo mongo:4.4 Unable
to find image mongo:4.4 locally
4.4: Pulling from library/mongo
d13d02fa248d: Already exists
bc8e2652ce92: Pull complete
3cc856886986: Pull complete
c319e9ec4517: Pull complete
b4cbf8808f94: Pull complete
cb98a53e6676: Pull complete
f0485050cd8a: Pull complete
ac36cdc414b3: Pull complete
61814e3c487b: Pull complete
523a9f1da6b9: Pull complete
3b4beaef77a2: Pull complete
Digest: sha256:d13c897516e497e898c229e2467f4953314b63e48d4990d3215d876ef9d1fc7c Status:
Downloaded newer image for mongo:4.4
d8f614a4969fb1229f538e171850512f10f490cb1a96fca27e4aa89ac082eba5
```

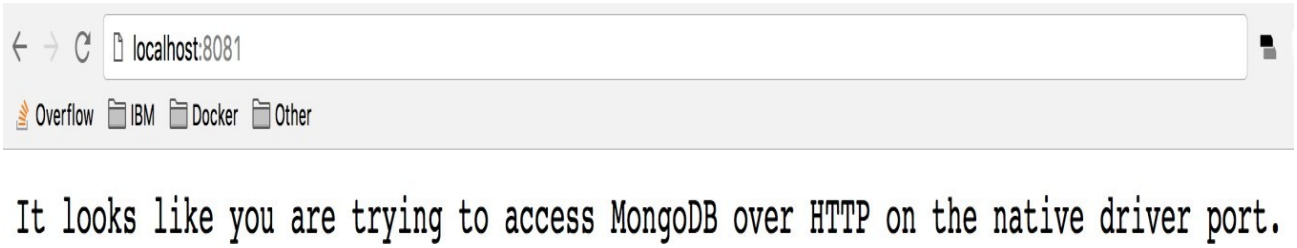
Access localhost:8081 to see some output from mongo.

curl localhost:8081

which will return a warning from MongoDB,

It looks like you are trying to access MongoDB over HTTP on the native driver port. If

you are using play-with-docker, look for the 8080 link near the top of the page.



Check your running containers with docker container ls

$ docker container ls

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d6777df89fea nginx "nginx -g 'daemon ..." Less than a second ago Up 2 seconds
0.0.0.0:8080-
>80/tcp nginx
ead80a0db505 mongo "docker-entrypoint..." 17 seconds ago Up 19 seconds 0.0.0.0:8081-
>27017/tcp mongo
af549dccd5cf ubuntu "top" 5 minutes ago Up 5 minutes priceless_kepler
```

Step 3: Clean Up

First get a list of the containers running using docker container ls.

```
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d6777df89fea nginx "nginx -g 'daemon ..." 3 minutes ago Up 3 minutes
0.0.0.0:8080-
>80/tcp nginx
ead80a0db505 mongo "docker-entrypoint..." 3 minutes ago
Up 3 minutes 0.0.0.0:8081->27017/tcp mongo
af549dccd5cf ubuntu "top" 8 minutes ago
Up 8 minutes priceless_kepler
```

Next, run docker container stop [container id] for each container in the list. You can also use the names of the containers that you specified before.

```
$ docker container stop
d67 ead af5 d67
e
a
d

a
f
5
```

1. ***Remove the stopped containers***

   docker system prune is a really handy command to clean up your system. It will remove any stopped containers, unused volumes and networks, and dangling images.

   ```
   $ docker system
   prune WARNING!
   This will remove:
       - all stopped containers
       - all volumes not used by at least one container
       - all networks not used by at least one container
       - all dangling images
   Are you sure you want to
   continue? [y/N] y Deleted
   Containers:
   7872fd96ea4695795c41150a06067d605f69702dbcb9ce49492c9029f0e1b44b
   ```

60abd5ee65b1e2732ddc02b971a86e22de1c1c446dab165462a08b037ef7835c
31617fdd8e5f584c51ce182757e24a1c9620257027665c20be75aa3ab6591740

Total reclaimed space: 12B

1. Define Cloud Computing with example.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

2. What is the working principle of Cloud Computing?

The cloud is a collection of computers and servers that are publicly accessible via the Internet. This hardware is typically owned and operated by a third party on a consolidated basis in one or more data center locations. The machines can run any combination of operating systems.

3. What are the advantages and disadvantages of Cloud Computing?

Advantages

Lower-Cost Computers for Users
Improved Performance
Lower IT Infrastructure Costs
Fewer Maintenance Issues
Lower Software Costs
Instant Software Updates
Increased Computing Power
Unlimited Storage Capacity
Increased Data Safety
Improved Compatibility Between Operating
Systems Improved Document Format Compatibility
Easier Group Collaboration
Universal Access to Documents
Latest Version Availability
Removes the Tether to Specific Devices

Disadvantages

Requires a Constant Internet Connection
Doesn't Work Well with Low-Speed Connections
Can Be Slow
Features Might Be Limited
Stored Data Might Not Be
Secure
If the Cloud Loses Your Data, You're Screwed

4. What is distributed system?

A *distributed system* is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal.

Three significant characteristics of distributed systems are:

- ✓ Concurrency of components
- ✓ Lack of a global clock
- ✓ Independent failure of components
- ✓ What is cluster?
- ✓ Acomputingclusterconsistsofinterconnectedstand-alonecomputerswhichworkcooperativelyasasingleintegratedcomputingresource.Inthepast,clusteredcomputersystemshavedemonstrated

5. What is grid computing?

Grid Computing enables virtuals organizations to share geographically distributed resources as they pursue common goals, assuming the absence of central location, central control, omniscience, and an existing trust relationship.

(or)

- ✓ Gridtechnologydemandsnewdistributedcomputingmodels,software/middlewaresupport,networkprotocols,andhardwareinfrastructures.
- ✓ NationalgridprojectsarefollowedbyindustrialgridplatformdevelopmentbyIBM,Microsoft,Sun,HP,Dell,Cisco,EMC,PlatformComputing,andothers. Newgridserviceproviders(GSPs)andnewgridapplicationshaveemergedrapidly,similartothegrowthofInternetandwebservicesinthepasttwodecades.
- ✓ gridsystemsareclassifiedinessentiallytwocategories:computationalordatagridsandP2Pgrids.

6. What are the business areas needs in Grid computing?

- ✓ Life Sciences
- ✓ Financial services
- ✓ Higher Education
- ✓ Engineering Services
- ✓ Government
- ✓ Collaborative games

7. List out the Grid Applications:

- ✓ Application partitioning that involves breaking the problem into discrete pieces
- ✓ Discovery and scheduling of tasks and workflow
- ✓ Data communications distributing the problem data where and when it is required
- ✓ Provisioning and distributing application codes to specific system nodes
- ✓ Autonomic features such as self-configuration, self-optimization, self-recovery and self-management

8. List some grid computing toolkits and frameworks?

- ✓ Globus Toolkit Globus Resource Allocation Manager(GRAM)
- ✓ Grid Security Infrastructure(GSI)
- ✓ Information Services
- ✓ Legion, Condor and Condor-G
- ✓ NIMROD, UNICORE, NMI.

9. What are Desktop Grids?

These are grids that leverage the compute resources of desktop computers.
Because of the true (but unfortunate) ubiquity of Microsoft® Windows® operating system in corporations, desktop grids are assumed to apply to the Windows environment.
The Mac OS™ environment is supported by a limited number of vendors.

10. What are Server Grids?

- ✓ Some corporations, while adopting Grid Computing , keep it limited to server resources that are within the purview of the IT department.
- ✓ Special servers, in some cases, are bought solely for the purpose of creating an internal "utility grid" with resources made available to various departments.
- ✓ No desktops are included in server grids. These usually run some flavor of the Unix/Linux operating system.

11. Define Opennebula.

OpenNebula is an open source management tool that helps virtualized data centers oversee private clouds, public clouds and hybrid clouds.....OpenNebula is vendor neutral, as well as platform- and API-agnostic. It
can use KVM, Xen or VMware hypervisors.

12. Define Eclipse.

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment.

13. Define Netbeans.

NetBeans is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages. NetBeans is also referred to as a platform of modular components used for developing Java desktop applications.

14. Define Apache Tomcat.

Apache Tomcat (or Jakarta Tomcat or simply Tomcat) is an open source servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run."

15. What is private cloud?

The *private cloud* is built within the domain of an intranet owned by a single organization. Therefore, they are client owned and managed. Their access is limited to the owning clients and their partners. Their deployment was not meant to sell capacity over the Internet through publicly accessible interfaces. Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.

16. What is public cloud?

A *public cloud* is built over the Internet, which can be accessed by any user who has paid for the service. Public clouds are owned by service providers. They are accessed by subscription. Many companies have built public clouds, namely Google App Engine, Amazon AWS, Microsoft Azure, IBM Blue Cloud, and Salesforce Force.com. These are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.

17. What is hybrid cloud?

A *hybrid cloud* is built with both public and private clouds, Private clouds can also support a *hybrid cloud* model by supplementing local infrastructure with computing capacity from an external public cloud. For example, the *research compute cloud* (RC2) is a private cloud built by IBM.

18. What is a Community Cloud ?

A community cloud in computing is a collaborative effort in which infrastructure is shared between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. This is controlled and used by a group of organizations that have shared interest. The costs are spread over fewer users than a public cloud (but more than a private cloud

19. Define IaaS?

The IaaS layer offers storage and infrastructure resources that is needed to deliver the Cloud services. It only comprises of the infrastructure or physical resource. Top IaaS Cloud Computing Companies: Amazon (EC2), Rackspace, GoGrid, Microsoft, Terremark and Google.

20. Define PaaS?

PaaS provides the combination of both, infrastructure and application. Hence, organizations using PaaS don't have to worry for infrastructure nor for services. Top PaaS Cloud Computing Companies: Salesforce.com, Google, Concur Technologies, Ariba, Unisys and Cisco..

21. Define SaaS?

In the SaaS layer, the Cloud service provider hosts the software upon their servers. It can be defined as a in model in which applications and softwares are hosted upon the server and made available to customers over a network. Top SaaS Cloud Computing Companies: Amazon Web Services, AppScale, CA Technologies, Engine Yard, Salesforce and Windows Azure.

22. What is meant by virtualization?

Virtualizationisacomputerarchitecturetechnologybywhichmultiplevirtualmachines (VMs)are multipl exedin the same hardwar emachine.Theideaof VMs canbe dated back to the 1960s. The purpose of a VM is to enhance resource sharing by many users and improve computer performance interms of resource utilization and application flexibility.

23. What are the implementation levels of virtualization?

The virtualization types are following

1. OS-level virtualization
2. ISA level virtualization
3. User-ApplicationLevel virtualization
4. hardware level virtualization
5. library level virtualization

24. List the requirements of VMM?

There are three requirements for a VMM.

First, a VMM should provide an environment for programs which is essentially identical to the original machine.

Second, programs run in this environment should show, at worst, only minor decreases in speed.

Third, a VMM should be in complete control of the system resources.

25. Explain Host OS and Guest OS?

A comparison of the differences between a host system, a guest system, and a virtual machine within a virtual infrastructure.

A host system (host operating system) would be the primary & first installed operating system. If you are using a bare metal Virtualization platform like Hyper-V or ESX, there really isn't a host operating system besides the Hypervisor. If you are using a Type-2 Hypervisor like VMware Server or Virtual Server, the host operating system is whatever operating system those applications are installed into.

A guest system (guest operating system) is a virtual guest or virtual machine (VM) that is installed under the host operating system. The guests are the VMs that you run in your virtualization platform.

26. Write the steps for live VM migration?

The five steps for live VM migration is

Stage 0: *Pre-Migration*
Active VM on Host A
Alternate physical host may be preselected for migration
Block devices mirrored and free resources maintained
Stage 1: *Reservation*
Initialize a container on the target
host Stage 2: *Iterative pre-copy*
Enable shadow paging
Copy dirty pages in successive
rounds. Stage 3: *Stop and copy*
Suspend VM on host A
Generate ARP to redirect traffic to Host B
Synchronize all remaining VM state to Host B
Stage 4: *Commitment*
VM state on Host A is released
Stage 5: *Activation*
VM starts on Host B
Connects to local
devices

Resumes normal operation

27..Define Globus Toolkit: Grid Computing Middleware

- ✓ Globus is open source grid software that addresses the most challenging problmes in distributed resources sharing.
- ✓ The Globus Toolkit includes software services and libraries for distributed security, resource management, monitoring and discovery, and data management.

28. Define Blocks in HDFS

- ✓ A disk has a block size, which is the minimum amount of data that it can read or write. Filesystems for a single disk build on this by dealing with data in blocks, which are an integral multiple of the disk block size. Filesystem blocks are typically a few kilobytes in size, while disk blocks are normally 512 bytes. This is generally transparent to the filesystem user who is simply reading or writing a file—of whatever length.

29. Define Namenodes and Datanodes

- ✓ An HDFS cluster has two types of node operating in a master-worker pattern:
  - ➢ a *namenode* (the master) and
  - ➢ a number of *datanodes*(workers).
- ✓ The namenode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree. This information is stored persistently on the local disk in the form of two files: the namespace image and the edit log.
- ✓ The namenode also knows the datanodes on which all the blocks for a given file are located, however, it does not store block locations persistently, since this information is reconstructed from datanodes when the system starts.

30. Define HADOOP.

Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation.

31. Define HDFS.

Hadoop Distributed File System (HDFS) is a Java-based file system that provides scalable and reliable data storage that is designed to span large clusters of commodity servers. HDFS, MapReduce, and YARN form the core of Apache™ Hadoop®.

32. Write about HADOOP.

Hadoop was created by Doug Cutting and Mike Cafarella in 2005. Cutting, who was working at Yahoo! at the time, named it after his son's toy elephant. It was originally developed to support distribution for the Nutch search engine project.

33. Definition of *Grid Portal*:

A *Grid Portal* provides an efficient infrastructure to put Grid-empowered applications on corporate Intranet/Internet.

34. Define GAE.

Google App Engine (often referred to as GAE or simply App Engine) is a Platform as a Service and cloud computing platform for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers. App Engine offers automatic scaling for web applications—as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.

35. What is Cloudsim?

CloudSim is a simulation toolkit that supports the modeling and simulation of the core functionality of cloud, like job/task queue, processing of events, creation of cloud entities(datacenter, datacenter brokers,

etc), communication between different entities, implementation of broker policies, etc. This toolkit allows to:

- Test application services in a repeatable and controllable environment.
- Tune the system bottlenecks before deploying apps in an actual cloud.
- Experiment with different workload mix and resource performance scenarios on simulated infrastructure for developing and testing adaptive application provisioning techniques

36. Core features of CloudSim are:

- The Support of modeling and simulation of large scale computing environment as federated cloud data centers, virtualized server hosts, with customizable policies for provisioning host resources to virtual machines and energy-aware computational resources
- It is a self-contained platform for modeling cloud's service brokers, provisioning, and allocation policies.
- It supports the simulation of network connections among simulated system elements.
- Support for simulation of federated cloud environment, that inter-networks resources from both private and public domains.
- Availability of a virtualization engine that aids in the creation and management of multiple independent and co-hosted virtual services on a data center node.
- Flexibility to switch between space shared and time shared allocation of processing cores to virtualized services.

37. Uses of Cloudsim.

- Load Balancing of resources and tasks
- Task scheduling and its migrations
- Optimizing the Virtual machine allocation and placement policies
- Energy-aware Consolidations or Migrations of virtual machines
- Optimizing schemes for Network latencies for various cloud scenarios

38. Define OpenStack.

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.A dashboard is also available, giving administrators control while empowering their users to provision resources through a web interface.

39. Define Trystack.

TryStack is a great way to take OpenStack for a spin without having to commit to a full deployment.

This free service lets you test what the cloud can do for you, offering networking, storage and compute instances, without having to go all in with your own hardware.

It's a labor of love spearheaded by three Red Hat OpenStack experts Will Foster, Kambiz Aghaiepour and Dan Radez.

TryStack's set-up must bear the load of anyone who wants to use it, but instead of an equally boundless budget and paid staff, it was originally powered by donated equipment and volunteers from Cisco, Dell, Equinix, NetApp, Rackspace and Red Hat who pulled together for this OpenStack Foundation project.

40. Define Hadoop.

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.