

## INDEX

| E.NO | DATE | EXPERIMENT NAME  | Marks | Sign |
|------|------|--|-------|------|
| 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 <i>hello world</i> app and other simple webapplications 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.   |       |      |

## **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 windowsOS on top of windows7 or 8.

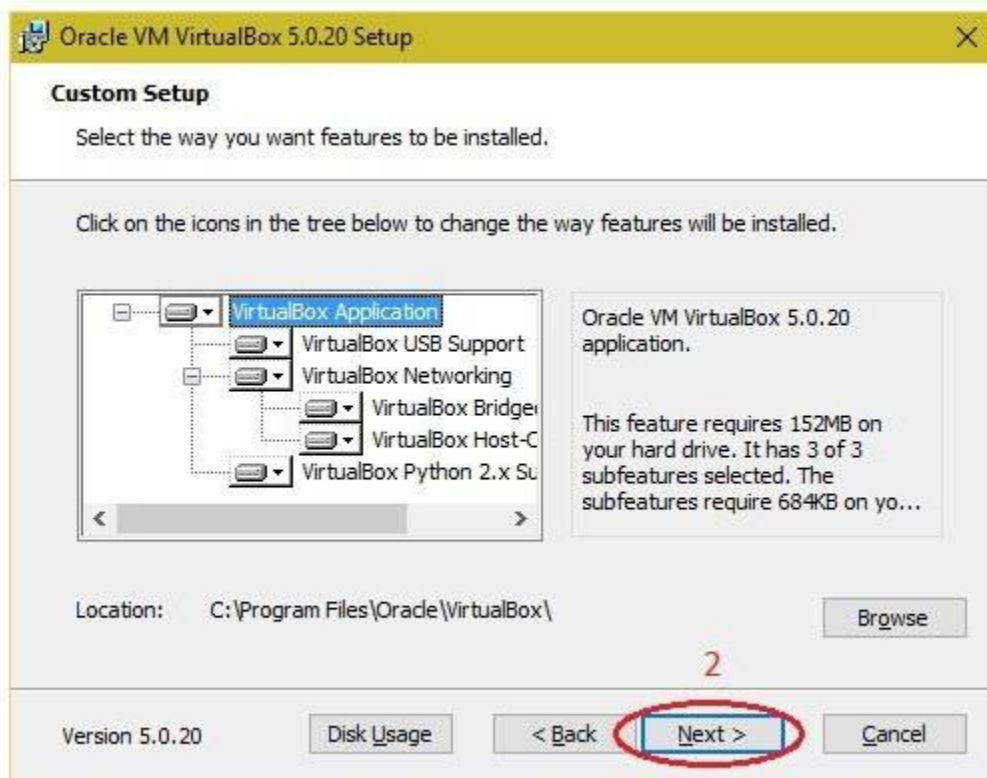
### **PROCEDURE:**

#### **Steps to install Virtual Box:**

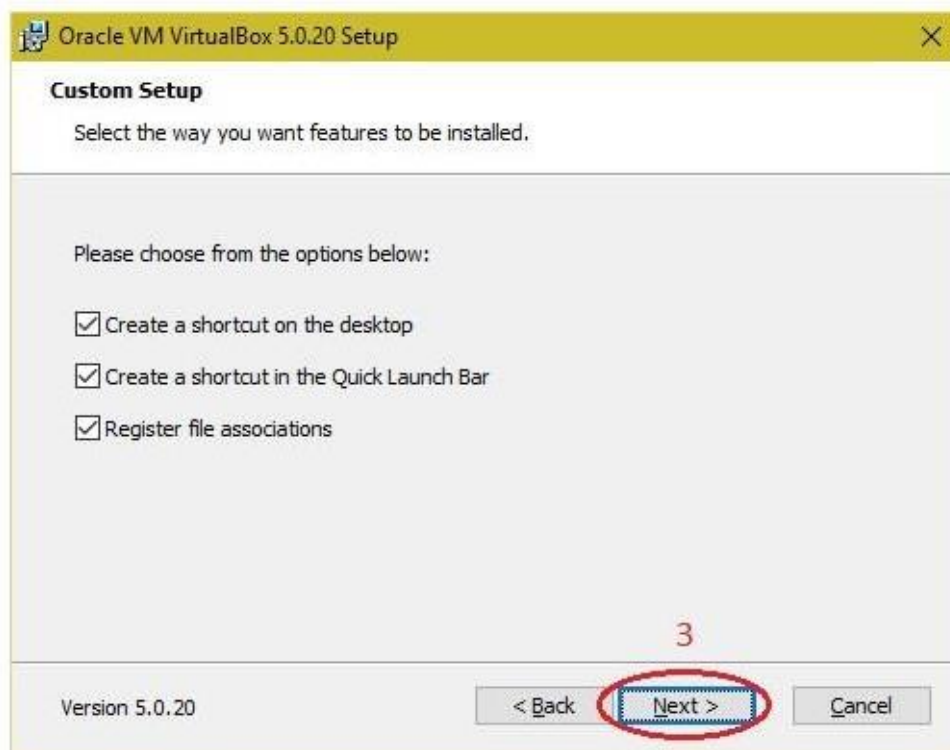
1. Download the Virtual box exe and click the exe file...and select next button..



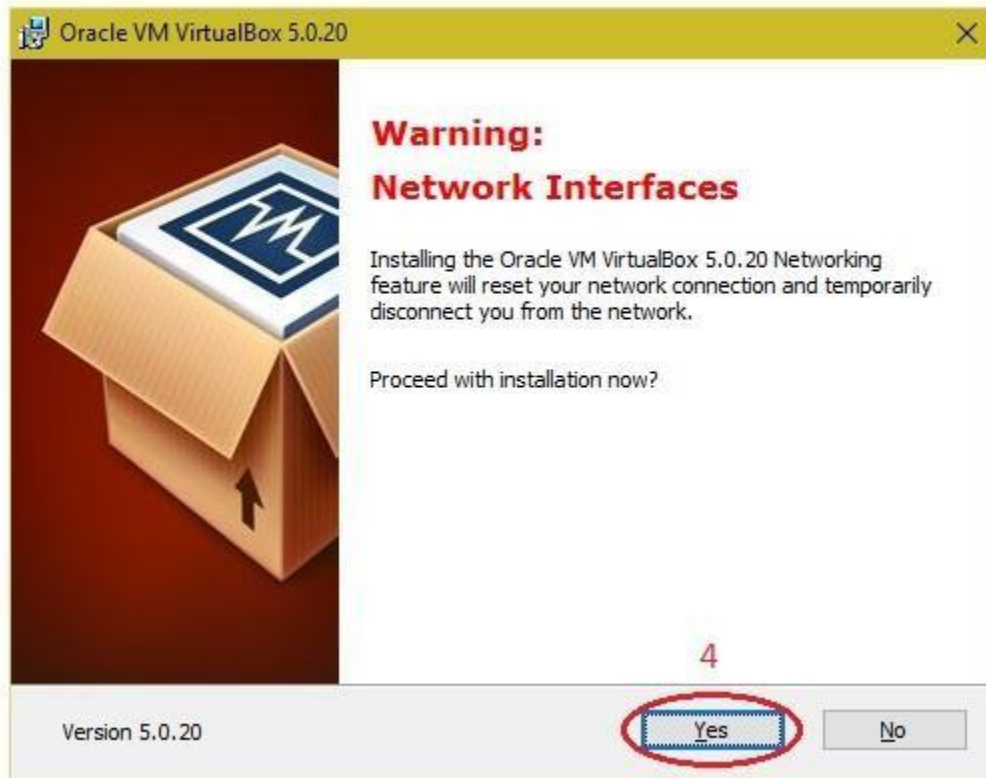
2. Click the next button..



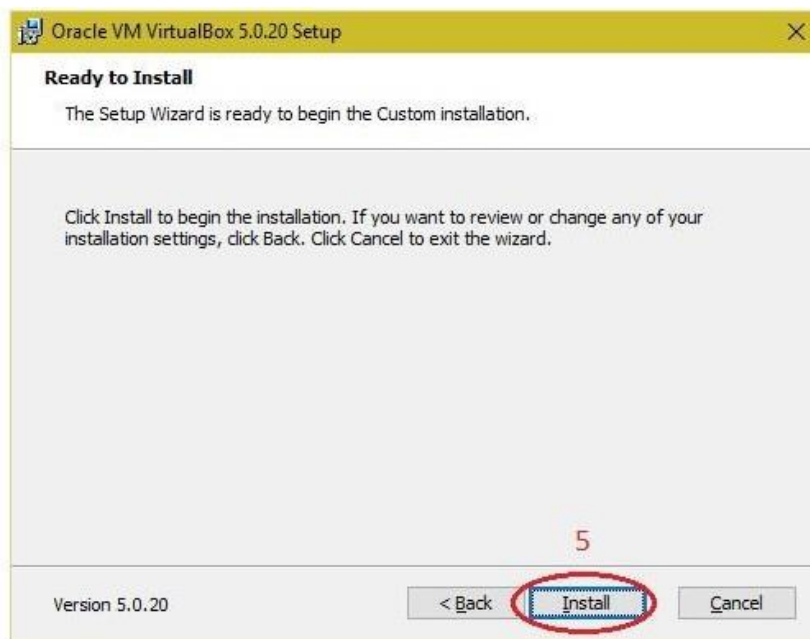
3. Click the next button



4. Click the YES button..



5. Click the install button...



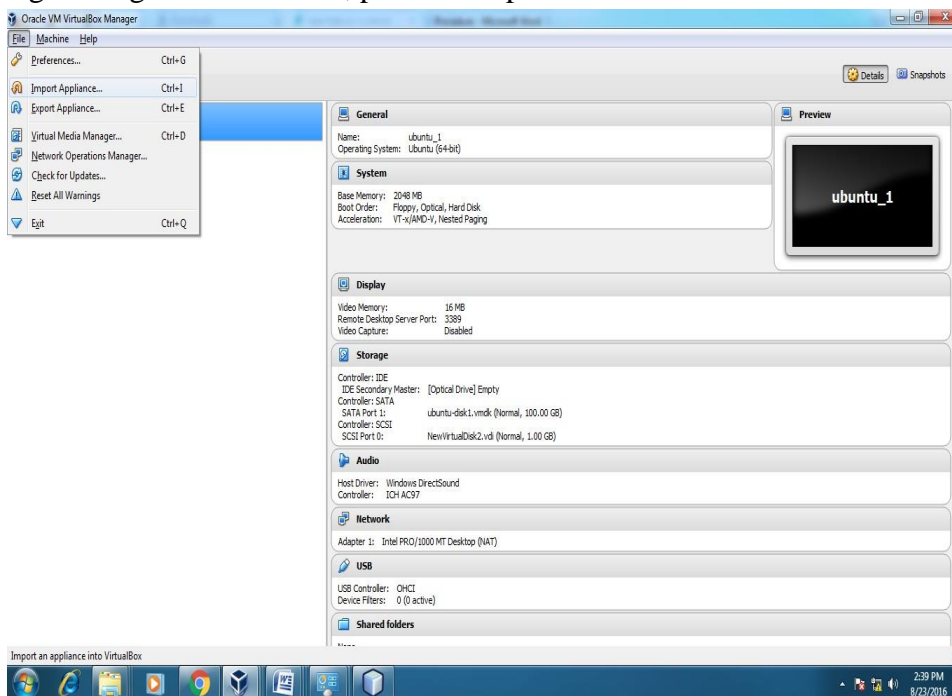
6. Then installation was completed..the show virtual box icon on desktop screen....

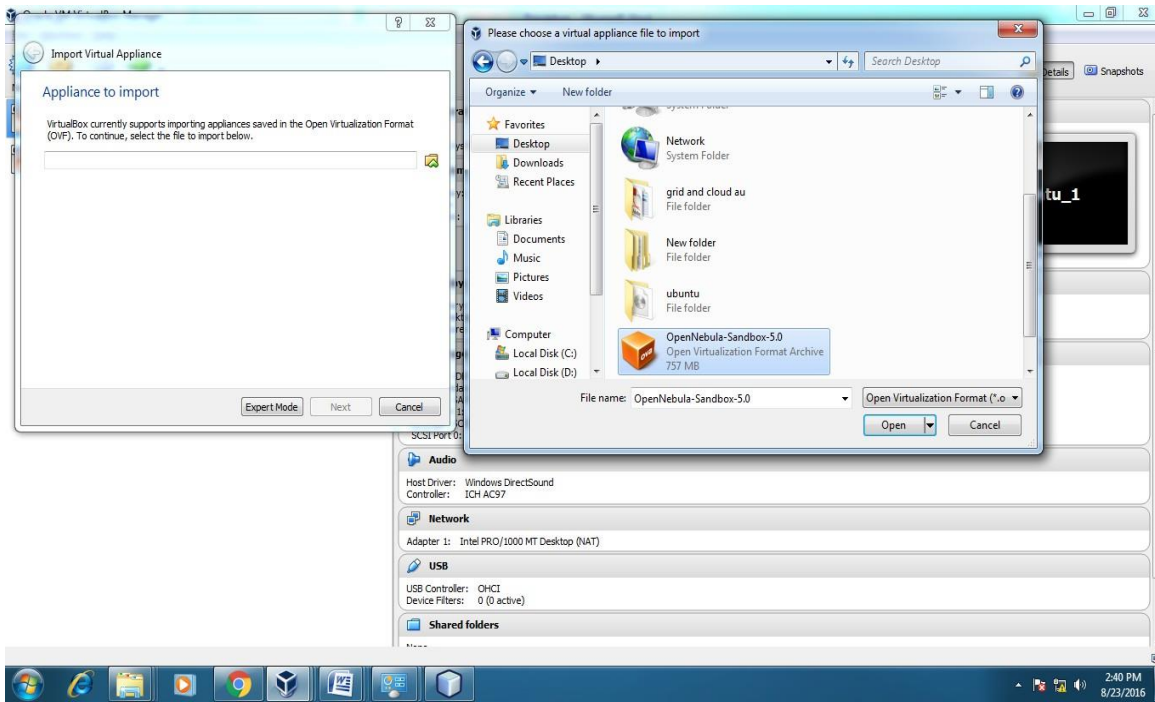


# VirtualBox

## Steps to import Open nebula sandbox:

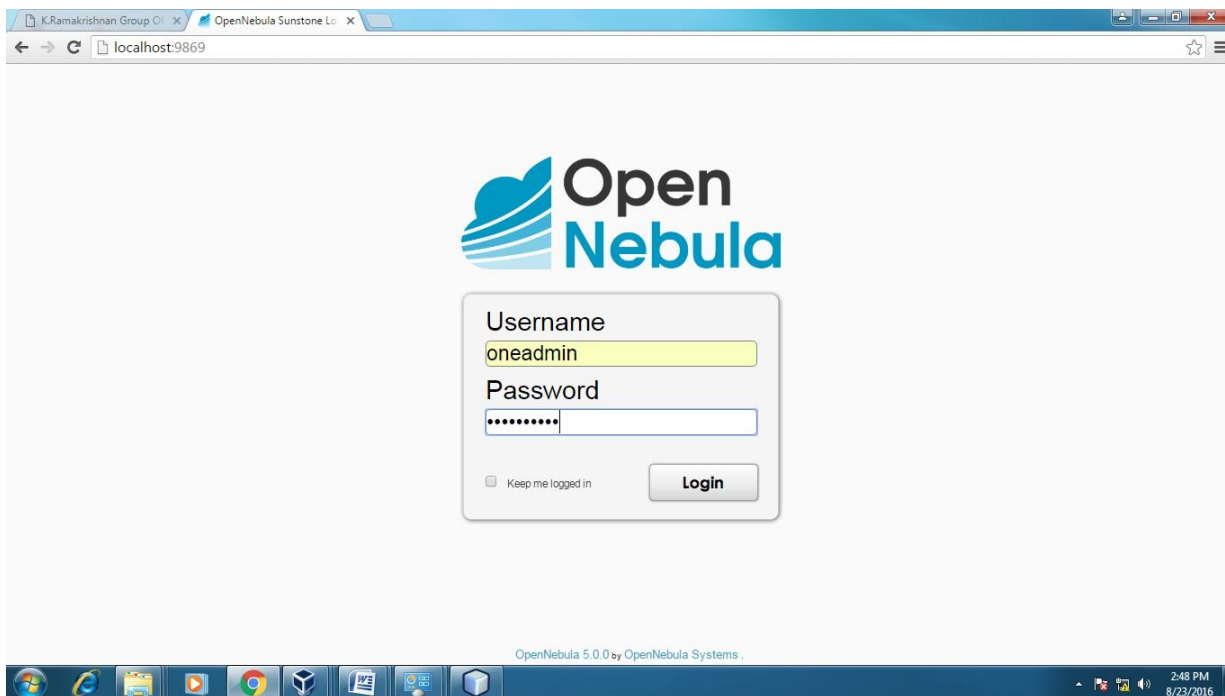
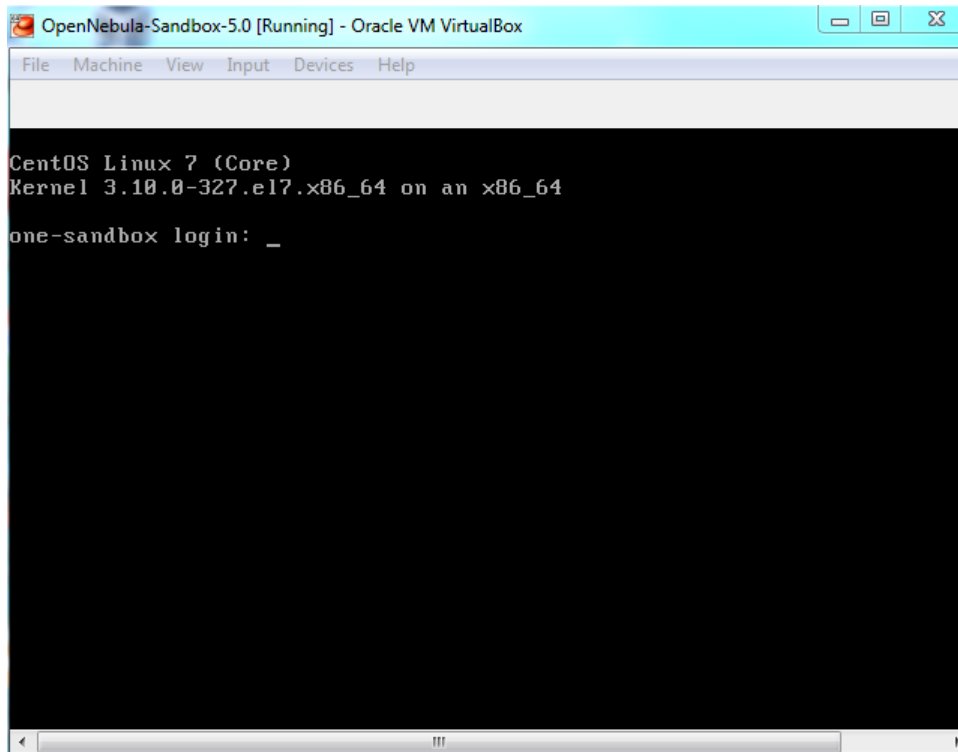
1. Open Virtual box
2. File ☐ import Appliance
3. Browse OpenNebula-Sandbox-5.0.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the Open Nebula
6. Login using username: root, password:opennebula





### **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 Virtual machine
  - a. Expand the + symbol
  - b. Select user oneadmin
  - c. Then enter the VM name, no. of instance, cpu.
  - d. Then click on create button.
  - e. Repeat the steps the C,D for creating more than one VMs.



## RESULT:

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

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

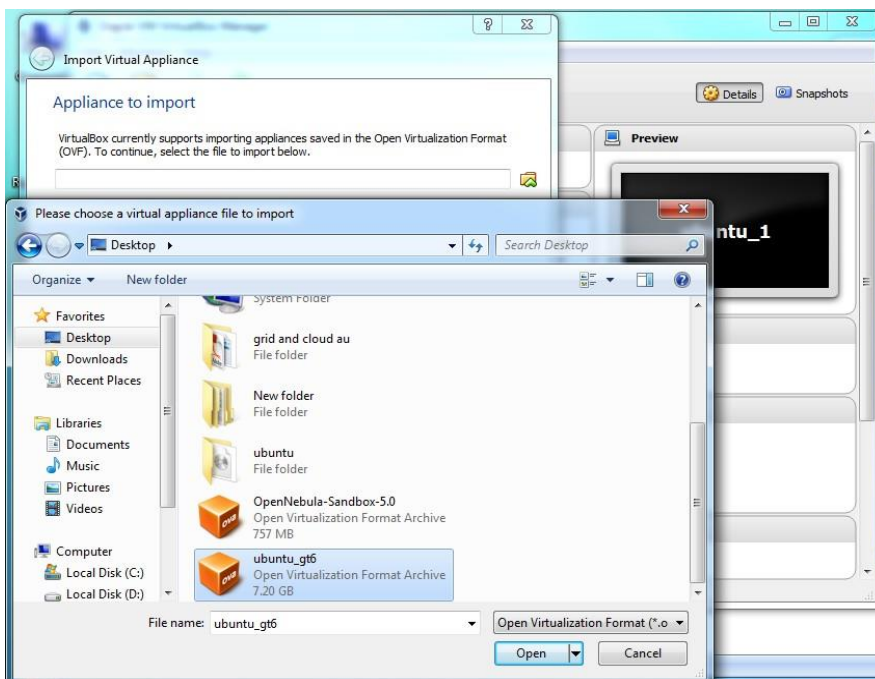
### Aim:

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.

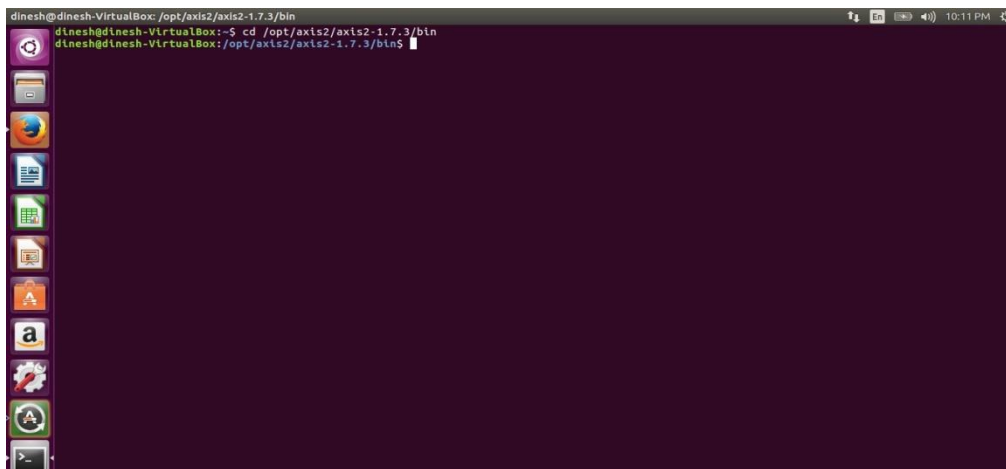


#### Steps to run c program:

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`

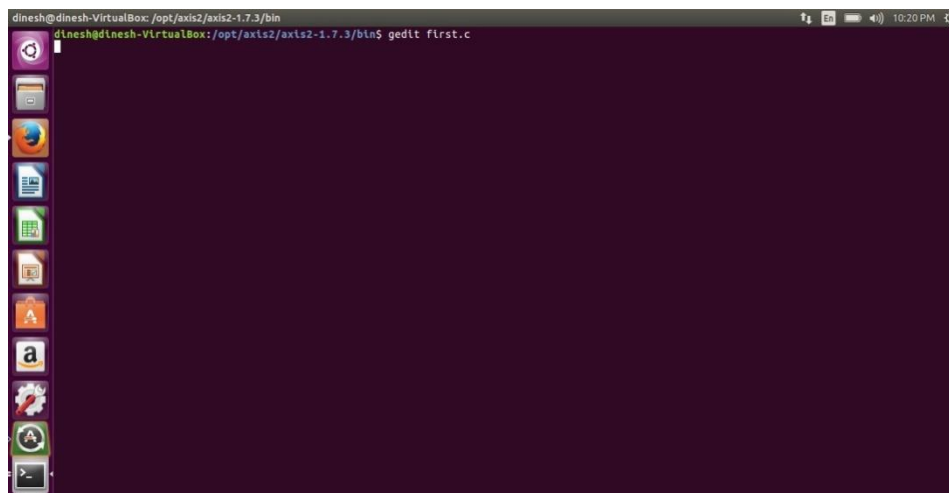


1. Type `cd /opt/axis2/axis2-1.7.3/bin` then press enter



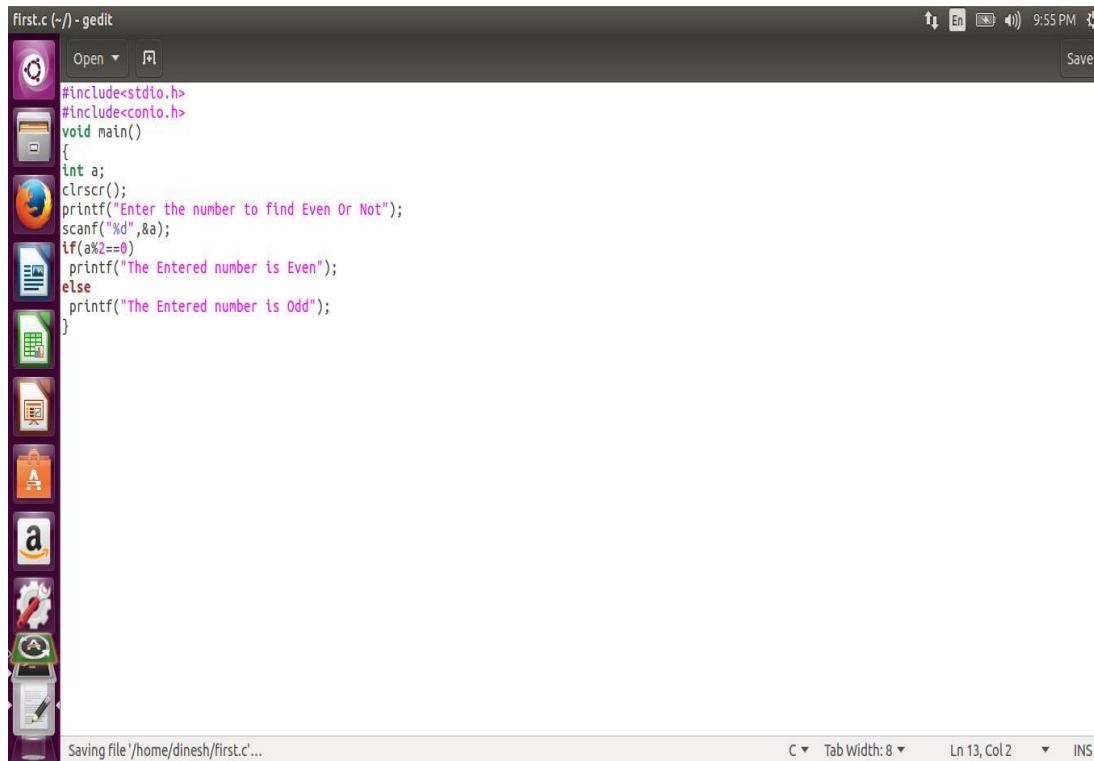
A terminal window titled 'dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin' is shown. The window has a dark purple background. The command prompt shows the user 'dinesh' at the host 'dinesh-VirtualBox' in the directory '/opt/axis2/axis2-1.7.3/bin'. The command `cd /opt/axis2/axis2-1.7.3/bin` has been entered and executed, as indicated by the prompt changing to `dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$`. A vertical dock with various application icons is visible on the left side of the window.

2. Type `gedit first.c`



A terminal window titled 'dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin' is shown. The window has a dark purple background. The command prompt shows the user 'dinesh' at the host 'dinesh-VirtualBox' in the directory '/opt/axis2/axis2-1.7.3/bin'. The command `gedit first.c` has been entered and executed, as indicated by the prompt changing to `dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ gedit first.c`. A vertical dock with various application icons is visible on the left side of the window.

### 3. Type the c program

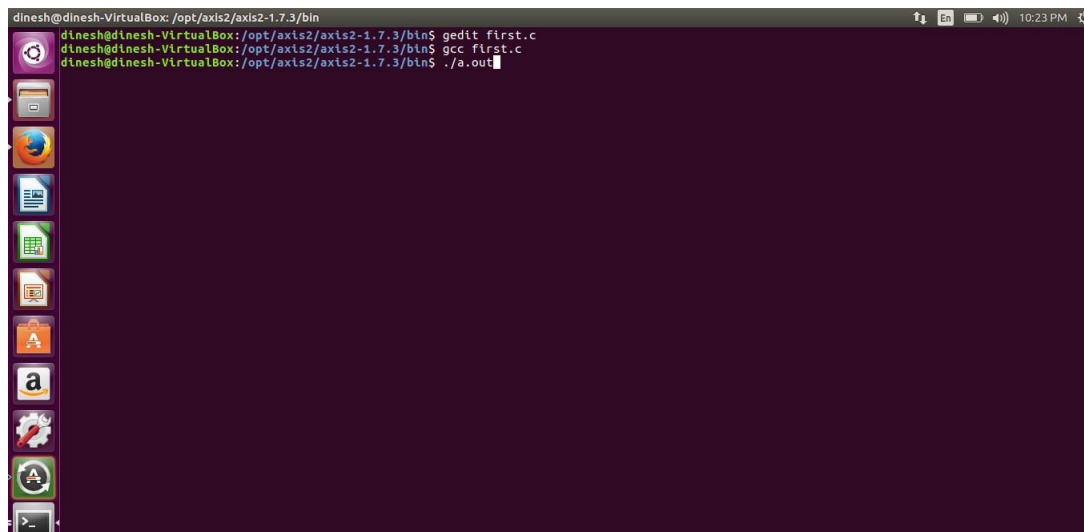


The screenshot shows a gedit editor window titled 'first.c (~/) - gedit'. The editor contains the following C code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a;
    clrscr();
    printf("Enter the number to find Even Or Not");
    scanf("%d",&a);
    if(a%2==0)
        printf("The Entered number is Even");
    else
        printf("The Entered number is Odd");
}
```

The status bar at the bottom indicates 'Saving file '/home/dinesh/first.c...' and 'C Tab Width: 8 Ln 13, Col 2 INS'.

### 4. Running the C program

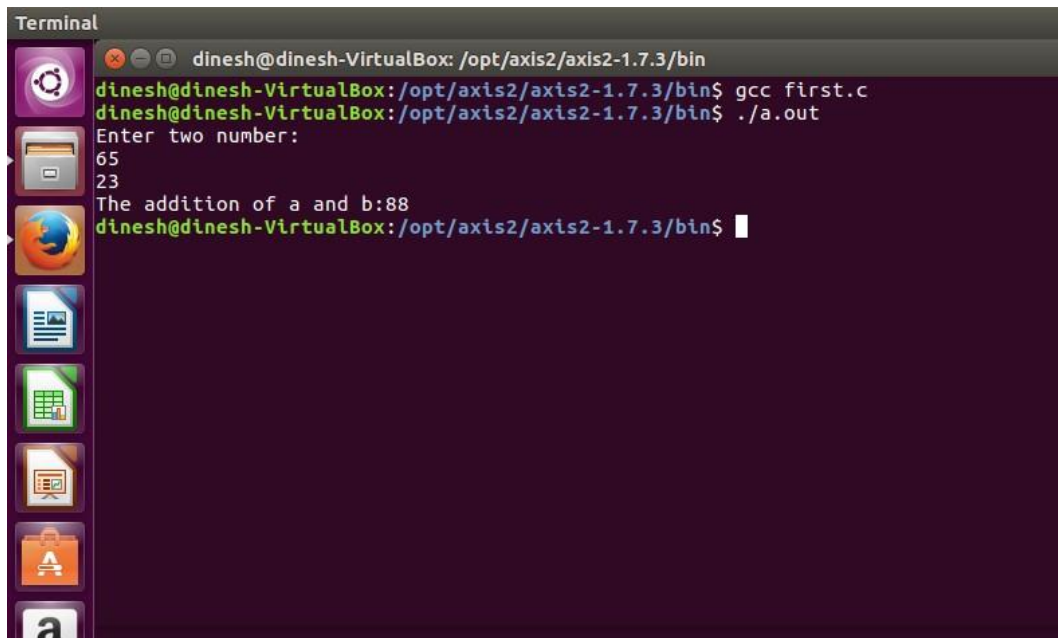


The screenshot shows a terminal window with the following commands and output:

```
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin$ gedit first.c
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin$ gcc first.c
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin$ ./a.out
```

The terminal window shows the user navigating to the directory where the program was saved, compiling it with gcc, and then running the resulting executable (a.out).

5. Display the output:

A screenshot of a Linux terminal window titled "Terminal". The window shows a user named "dinesh" at a machine named "dinesh-VirtualBox". The current directory is "/opt/axis2/axis2-1.7.3/bin". The user has compiled a C program named "first.c" using "gcc" and then executed it with "./a.out". The program prompts the user to "Enter two number:", and the user has entered "65" and "23". The program then outputs "The addition of a and b:88". The terminal window has a sidebar on the left with various application icons.

```
Terminal
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ gcc first.c
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$ ./a.out
Enter two number:
65
23
The addition of a and b:88
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$
```

#### **APPLICATIONS:**

Simply running all programs in grid environment.

#### **RESULT:**

Thus the C compiler in the virtual machine created using virtual box and simple C programs executed successfully.

## Ex.No:3      Install Google App Engine. Create *hello world* app and other simple webapplications using python/java.

### Aim:

To Install Google App Engine. Create *hello world* app and other simple webapplications 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 [GoogleApp 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

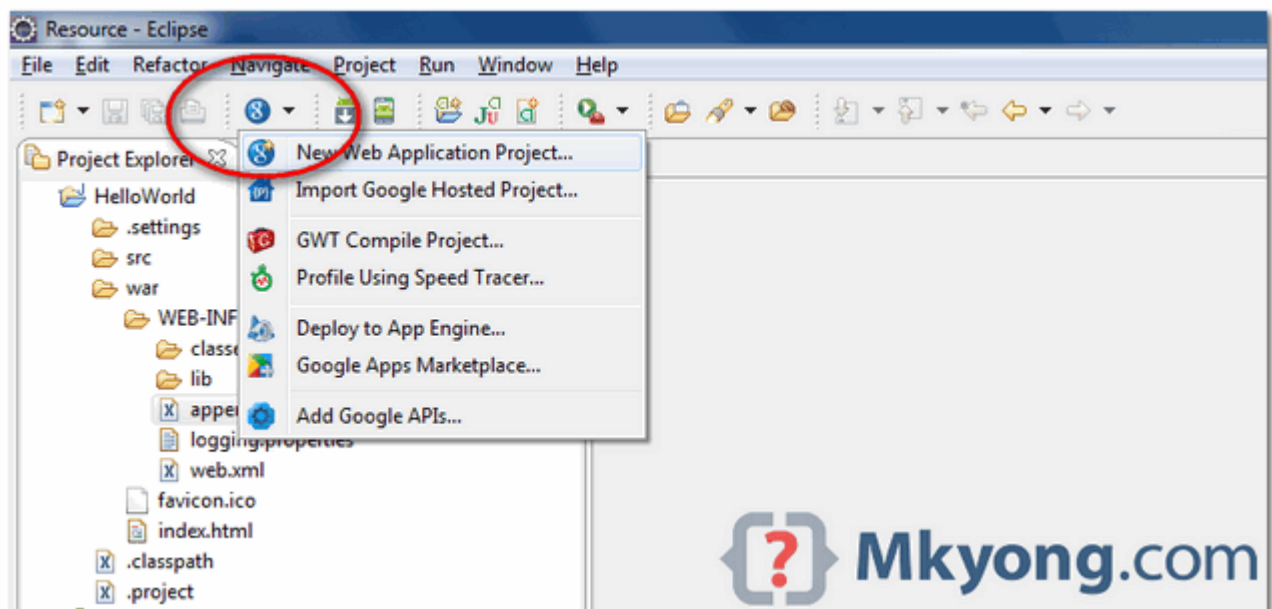
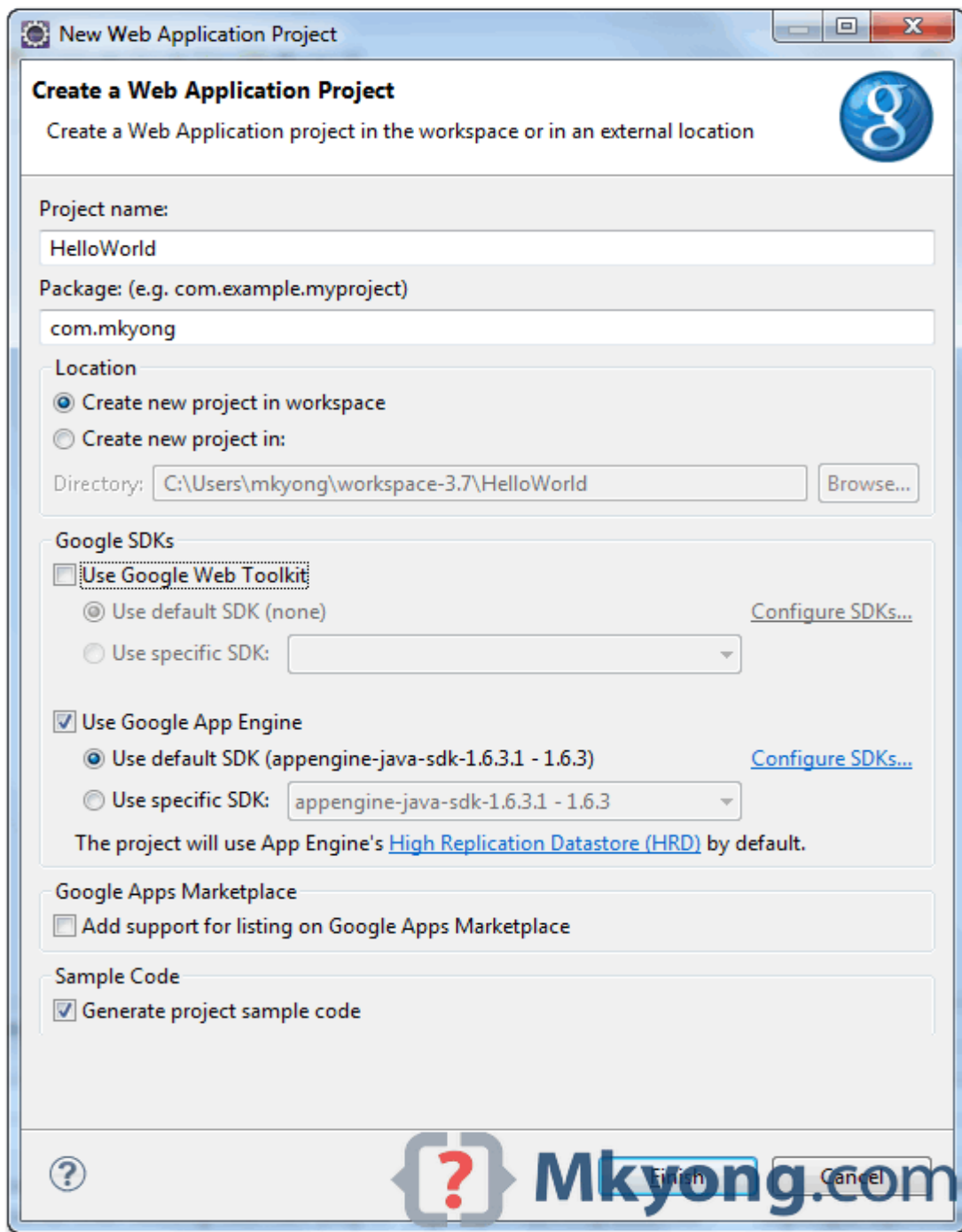


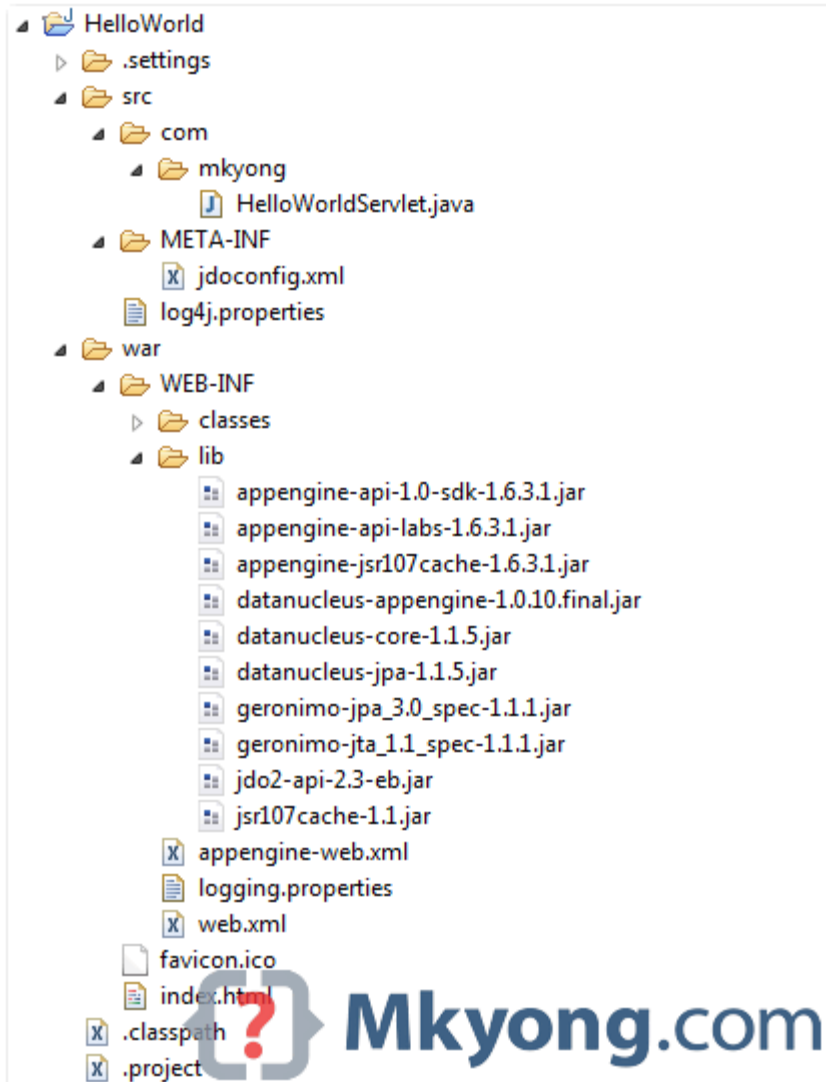
Figure – Deselect the “Google Web ToolKit“, and link your GAE Java SDK via the “configureSDK” link.



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 :*

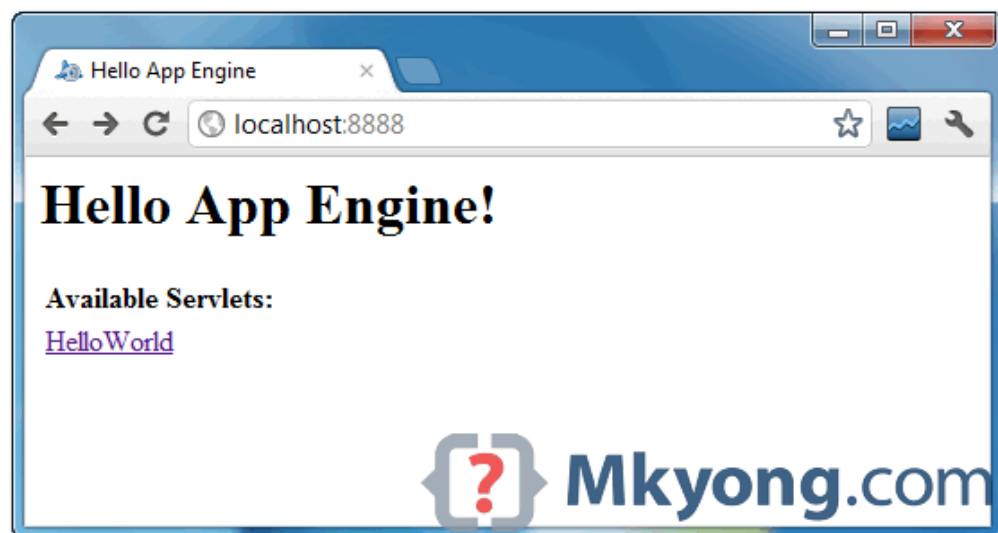
//...

INFO: The server is running at http://localhost:8888/

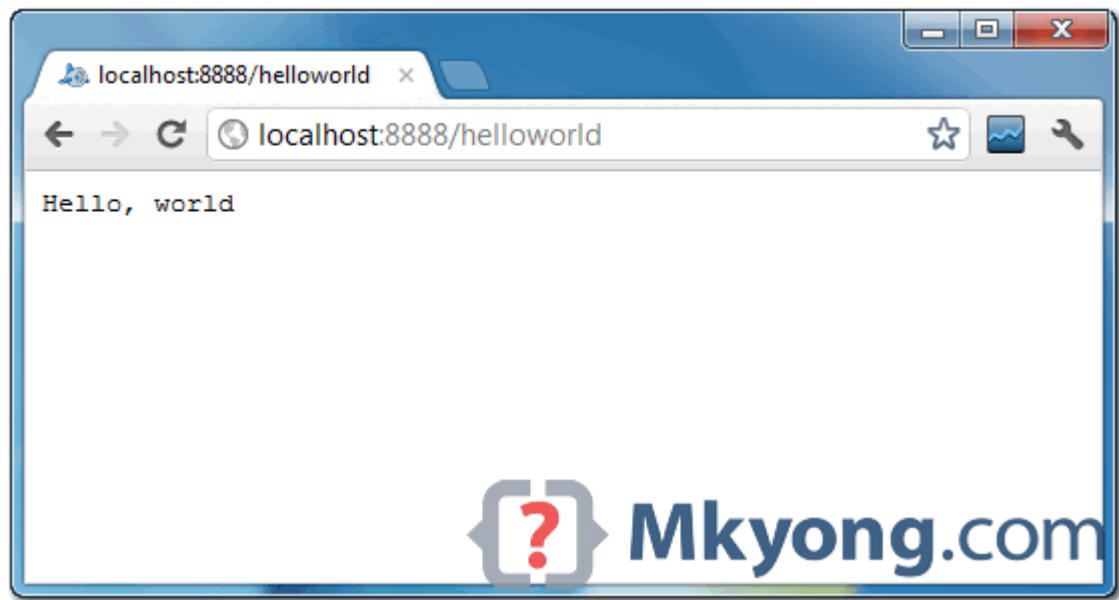
30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl startINFO: The admin console is running at http://localhost:8888/\_ah/admin

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 webapplication.

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.

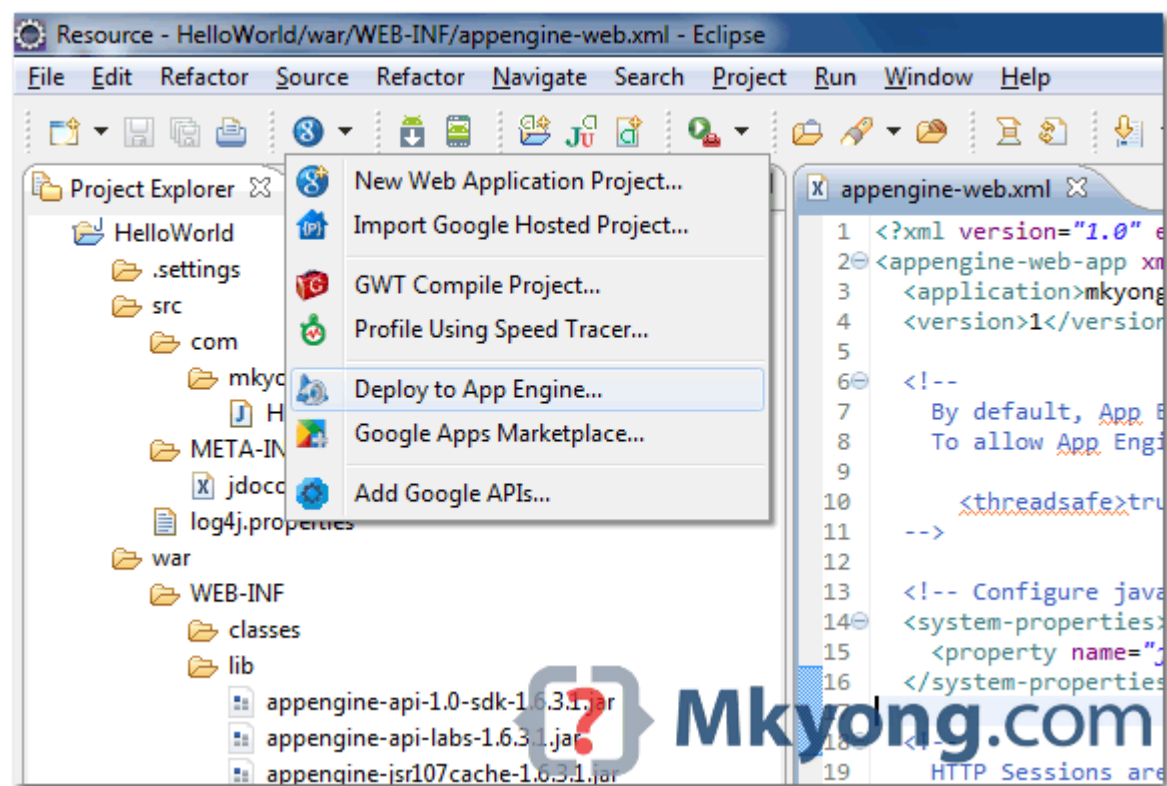
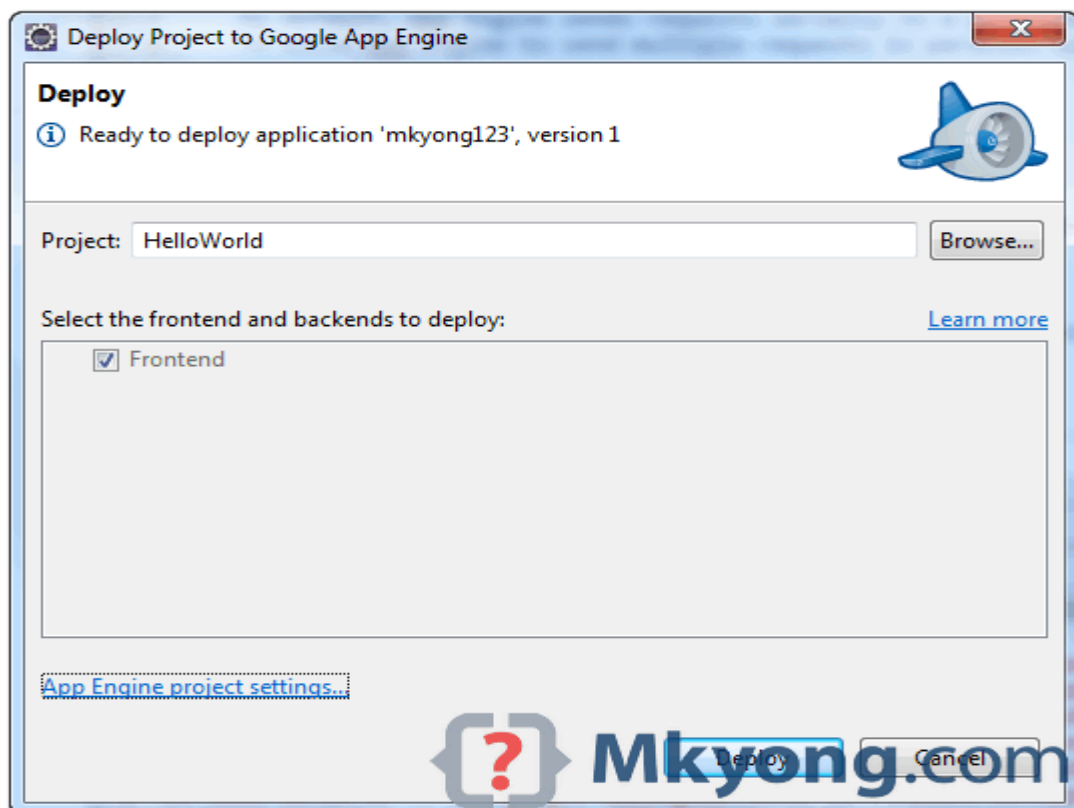


Figure 1.2 – Sign in with your Google account and click on the Deploy button.



*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

## Use GAE launcher to launch the web applications.

### 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](http://www.jedit.org)), create a file called app.yaml in the ae-01-trivial folder with the following contents:

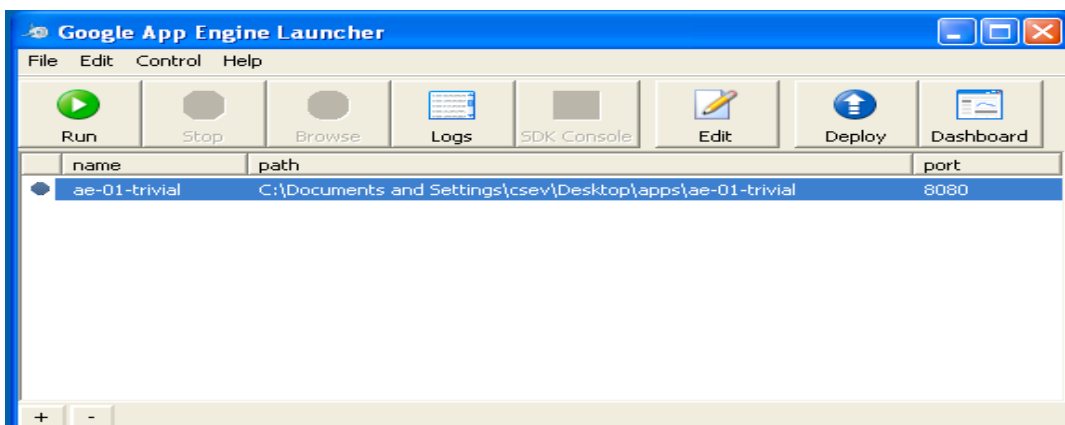
```
application: ae-01-trivial
version: 1
runtime: python api_version: 1
handlers:- url: /.
script: index.py
```

Note: Please do not copy and paste these lines into your text editor – you might end 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 Google App Engine Launcher program that can be found under Applications. Use the File → Add Existing Application command and navigate into the apps directory and select the ae-01-trivial folder. Once you have added the application, select it so that you can control the application using the launcher.



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.py` to change the name “Chuck” to your row

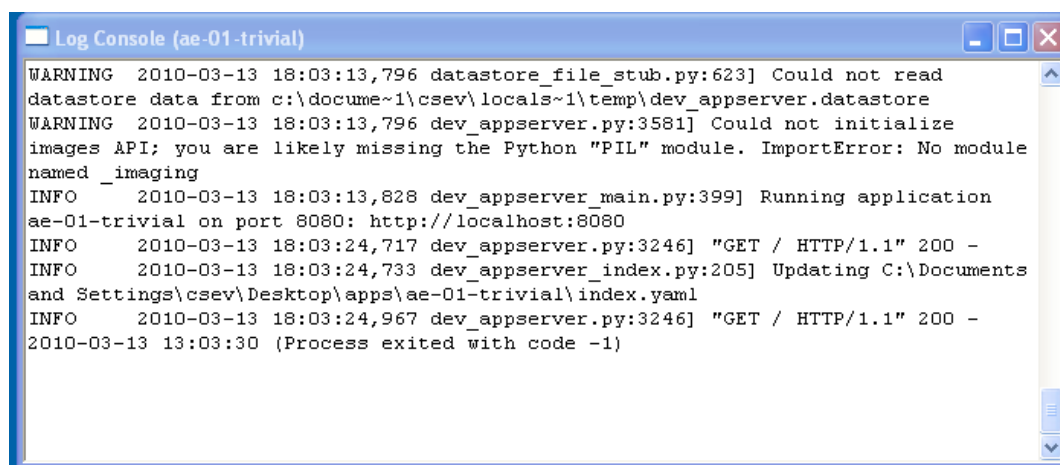
name and press Refresh in the browser to verify your updates.

## Watching the Log

You can watch the internal log of the actions that the 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.

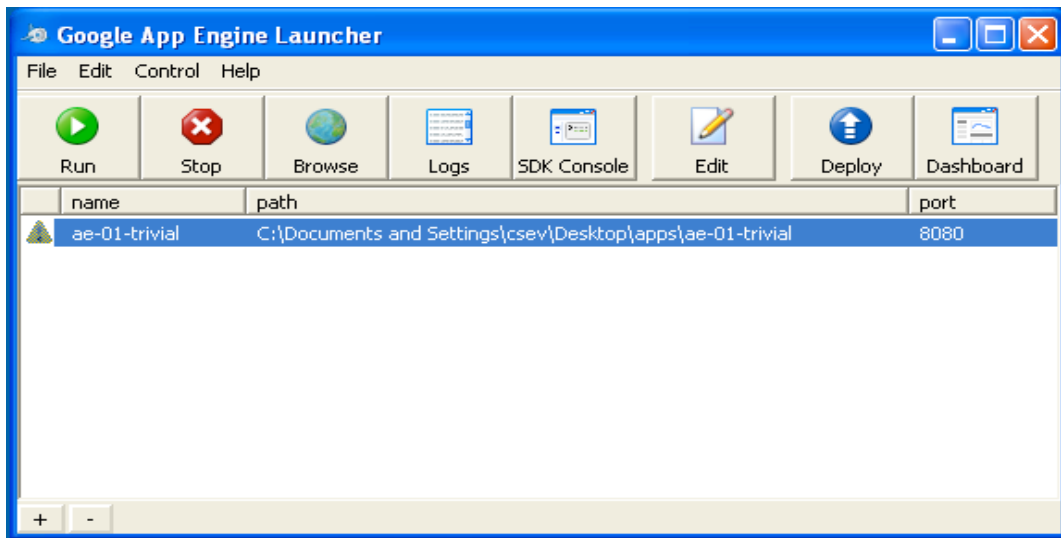
Each time you press Refresh in your browser—you can see it retrieving the output with a GET request.



## Dealing With Errors

With two files to edit, there are two general categories of errors that you may encounter. If

you make a mistake on the app.yaml file, the App Engine will not start 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.

```
http://localhost:8080/ - Windows Internet Explorer
http://localhost:8080/
download python 2.5.4
File Edit View Favorites Tools Help
http://localhost:8080/
Traceback (most recent call last):
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    self.Dispatch(dispatcher, self.rfile, outfile, env_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    base_env_dict=env_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    base_env_dict=base_env_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    self.module_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    reset_modules = exec_script(handler_path, cgi_path, hook)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    handler_path, cgi_path, import_hook)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    module_code = compile(source_file.read(), cgi_path, 'exec')
  File "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\index.py", line 3
    print 'Hello, World!'
SyntaxError: EOL while scanning single-quoted string
```

The error 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](http://en.wikipedia.org/wiki/Stack_trace)

When you make a mistake in the app.yaml file – you must 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:5****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.

**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 space shared, 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` of the VM's owner, `mips`, number of processors, amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudlet scheduler 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)`

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

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

9. Submit the cloudlet list to the broker:

`broker.submitCloudletList(cloudletList)` Sample

Output from the Existing Example:

Starting

CloudSimExample1...

[illegible]

```

Broker is starting... Entities started.
: Broker: Cloud Resource List received with 1
resource(s) 0.0: Broker: Trying to Create VM #0in
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 IDFinish  VM ID  Time  Start Time
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.



**Ex.No:6 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 dealing 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 us 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 runs a virtual machine, for the hosting OS those virtual machines 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 overcome it: it's enough to give authorization to read/write/execute to a directory or to choose 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!

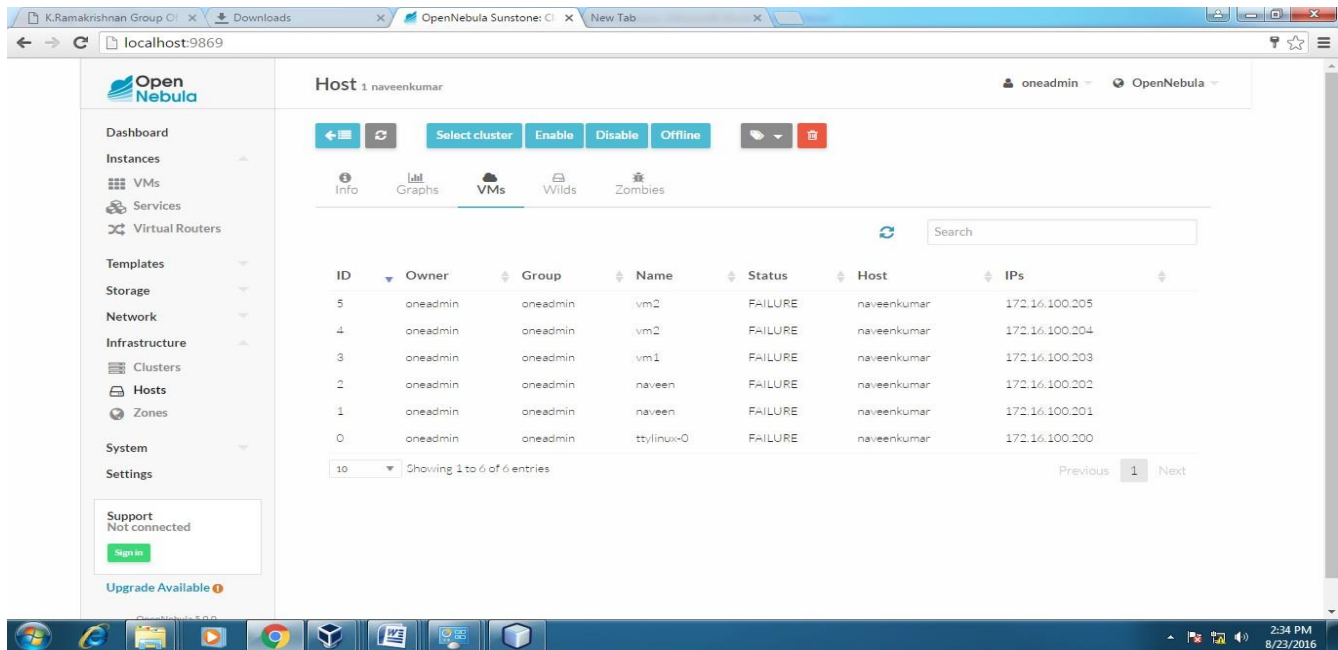
## PROCEDURE:

### 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<sup>th</sup> 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

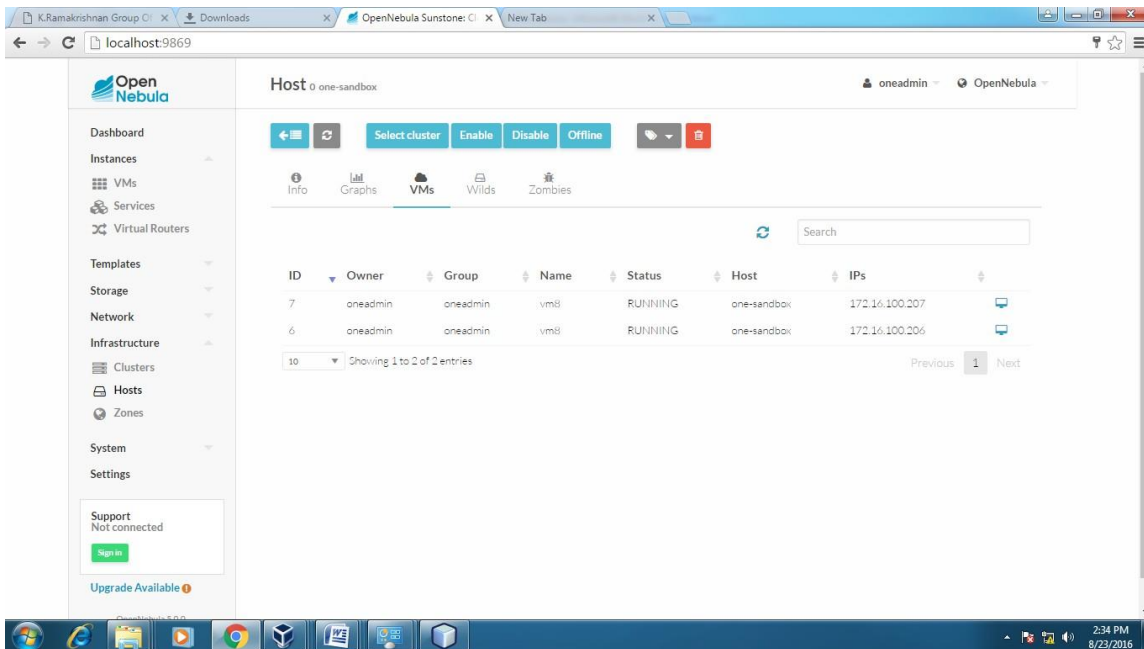
### Host:SACET



The screenshot shows the OpenNebula Sunstone web interface for Host 1 naveenkumar. The left sidebar contains navigation links for Dashboard, Instances, VMs, Services, Virtual Routers, Templates, Storage, Network, Infrastructure, Clusters, Hosts, Zones, System, and Settings. The main content area displays the VMs tab for Host 1 naveenkumar. At the top, there are buttons for Select cluster, Enable, Disable, and Offline. Below these are tabs for Info, Graphs, VMs, Wilds, and Zombies. A search bar is present. The VMs table lists 6 entries, all with a status of FAILURE. The table columns are ID, Owner, Group, Name, Status, Host, and IPs. The bottom of the interface shows a taskbar with various application icons and a system clock indicating 2:34 PM on 8/23/2016.

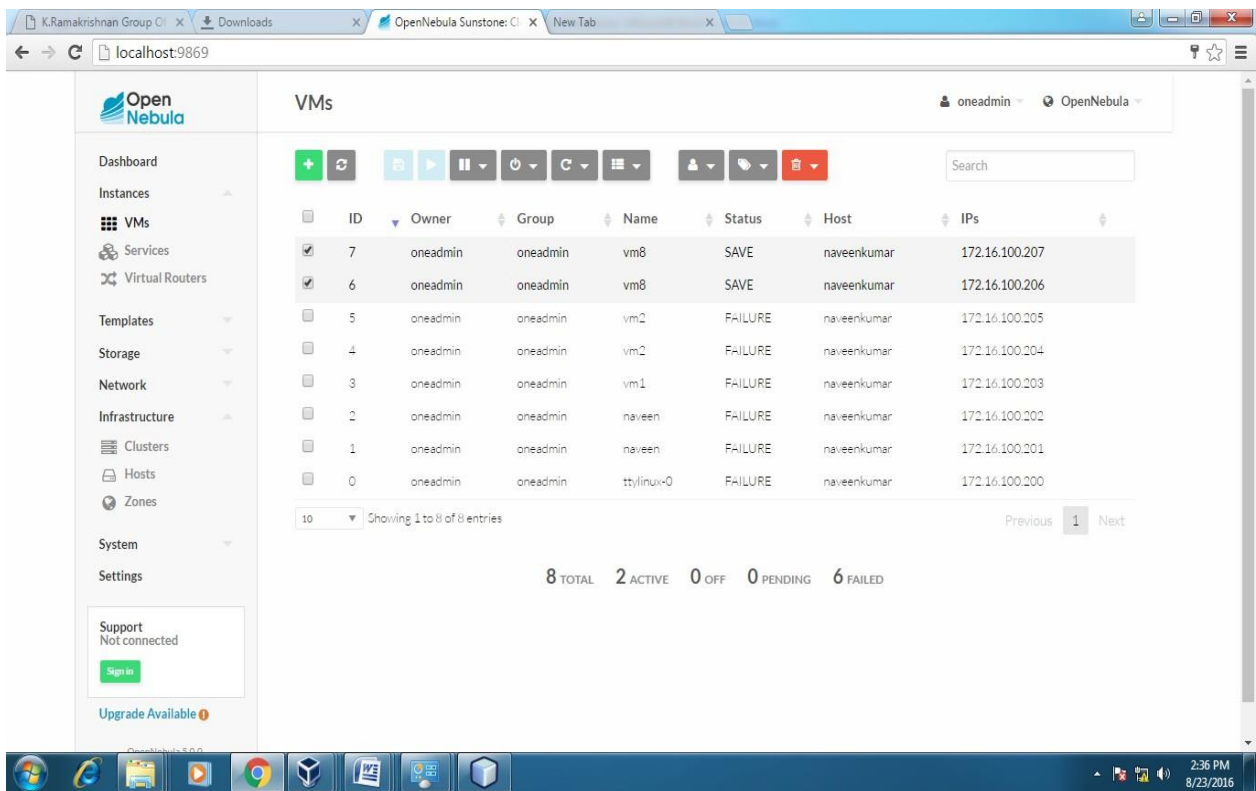
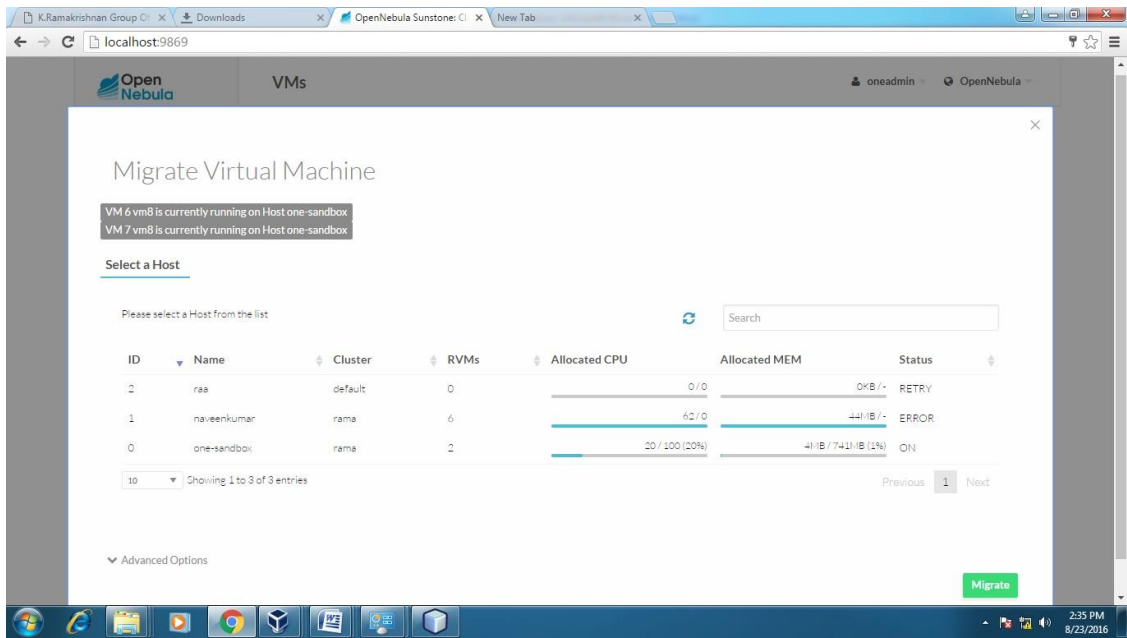
| ID | Owner    | Group    | Name      | Status  | Host        | IPs            |
|----|----------|----------|-----------|---------|-------------|----------------|
| 5  | oneadmin | oneadmin | vm2       | FAILURE | naveenkumar | 172.16.100.205 |
| 4  | oneadmin | oneadmin | vm2       | FAILURE | naveenkumar | 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 | ttlinux-0 | FAILURE | naveenkumar | 172.16.100.200 |

## Host:one-sandbox



The screenshot shows the OpenNebula Sunstone web interface for Host 0 one-sandbox. The left sidebar contains navigation links for Dashboard, Instances, VMs, Services, Virtual Routers, Templates, Storage, Network, Infrastructure, Clusters, Hosts, Zones, System, and Settings. The main content area displays the VMs tab for Host 0 one-sandbox. At the top, there are buttons for Select cluster, Enable, Disable, and Offline. Below these are tabs for Info, Graphs, VMs, Wilds, and Zombies. A search bar is present. The VMs table lists 2 entries, both with a status of RUNNING. The table columns are ID, Owner, Group, Name, Status, Host, and IPs. The bottom of the interface shows a taskbar with various application icons and a system clock indicating 2:34 PM on 8/23/2016.

| ID | Owner    | Group    | Name | Status  | Host        | IPs            |
|----|----------|----------|------|---------|-------------|----------------|
| 7  | oneadmin | oneadmin | vm8  | RUNNING | one-sandbox | 172.16.100.207 |
| 6  | oneadmin | oneadmin | vm8  | RUNNING | one-sandbox | 172.16.100.206 |



## After Migration:

**Hosts**

oneadmin OpenNebula

Select cluster Enable Disable Offline Search

| ID | Name        | Cluster | RVMs | Allocated CPU | Allocated MEM    | Status |
|----|-------------|---------|------|---------------|------------------|--------|
| 2  | raa         | default | 0    | 0 / 0         | 0KB / -          | ERROR  |
| 1  | naveenkumar | rama    | 8    | 82 / 0        | 48MB / -         | ERROR  |
| 0  | one-sandbox | rama    | 0    | 0 / 100 (0%)  | 0KB / 741MB (0%) | ON     |

Showing 1 to 3 of 3 entries

Previous 1 Next

3 TOTAL 1 ON 0 OFF 2 ERROR

Support Not connected Sign in Upgrade Available

## Host:one-sandbox

**Host: one-sandbox**

oneadmin OpenNebula

Select cluster Enable Disable Offline Search

Info Graphs VMs Wilds Zombies

| ID | Owner | Group | Name | Status | Host | IPs |
|----|-------|-------|------|--------|------|-----|
|----|-------|-------|------|--------|------|-----|

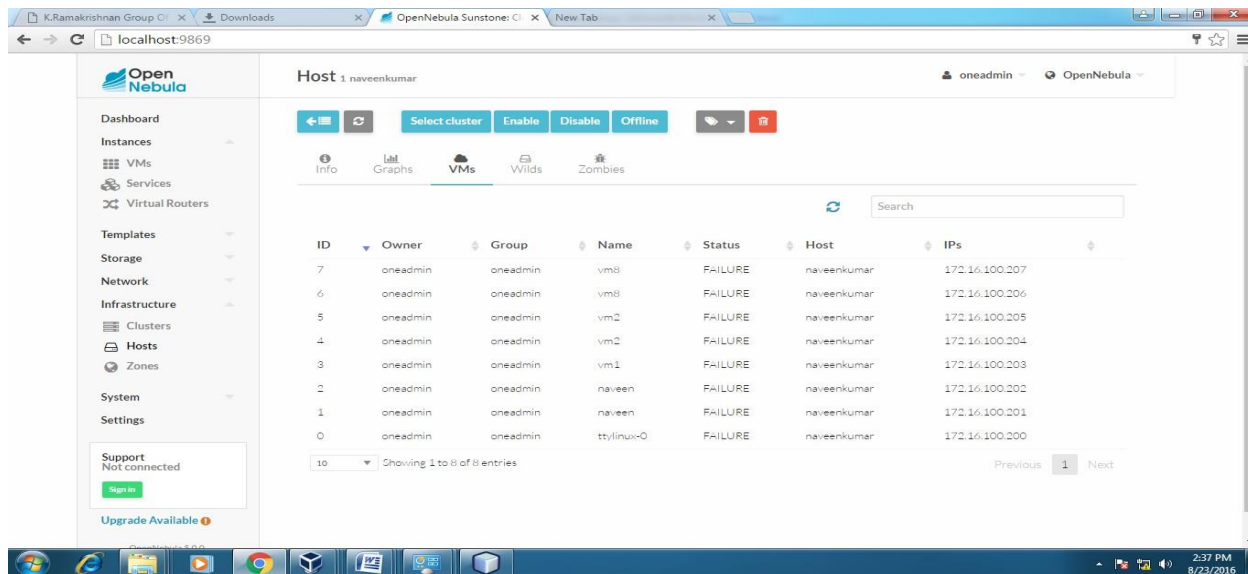
Showing 0 to 0 of 0 entries

Previous Next

There is no data available

Support Not connected Sign in Upgrade Available

## 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:7      **Install Hadoop single node cluster and run simple applications like word count.**

### **Aim:**

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

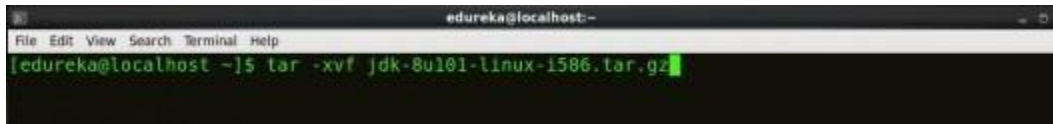
### **Steps:**

#### **Install Hadoop**

**Step 1:** [Click here](#) to download the Java 8 Package. Save this file in your homedirectory.

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

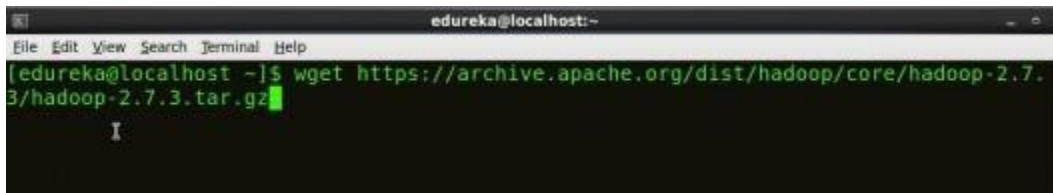
**Command:** `tar -xvf jdk-8u101-linux-i586.tar.gz`

A terminal window titled 'edureka@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[edureka@localhost ~]\$ tar -xvf jdk-8u101-linux-i586.tar.gz' is entered and executed, with a green cursor at the end of the line.

*Fig: Hadoop Installation – Extracting Java Files*

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

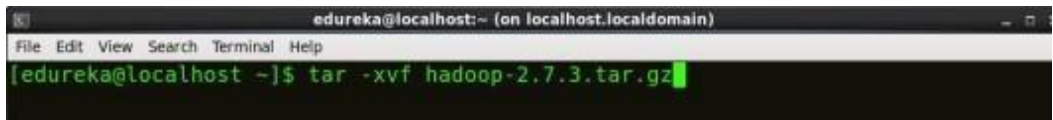
**Command:** `wget- https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz`

A terminal window titled 'edureka@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[edureka@localhost ~]\$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz' is entered and executed, with a green cursor at the end of the line.

*Fig: Hadoop Installation – Downloading Hadoop*

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

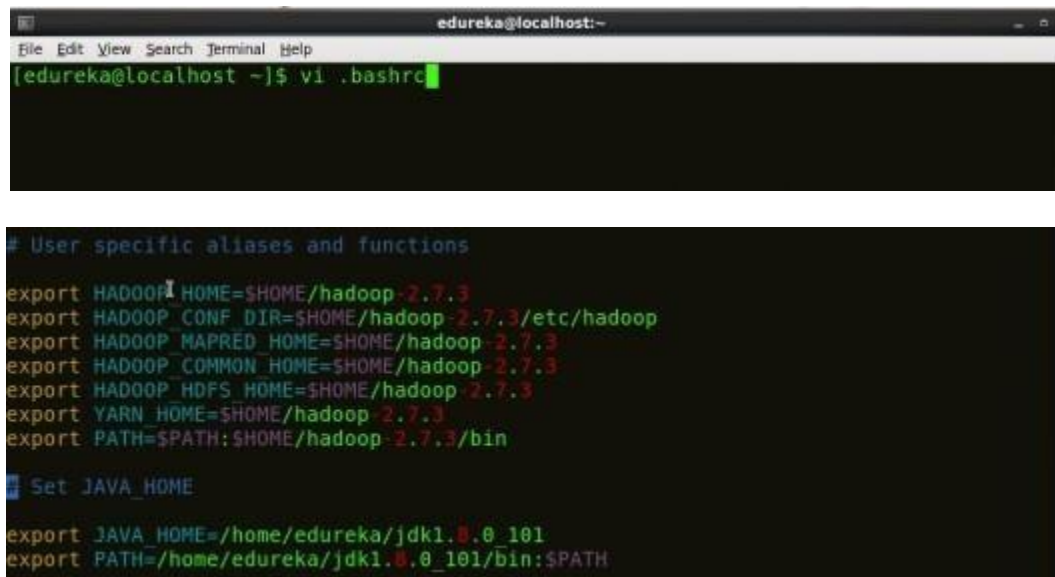
**Command:** `tar -xvf hadoop-2.7.3.tar.gz`

A terminal window titled 'edureka@localhost:~ (on localhost.localdomain)' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[edureka@localhost ~]\$ tar -xvf hadoop-2.7.3.tar.gz' is entered and executed, with a green cursor at the end of the line.

*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



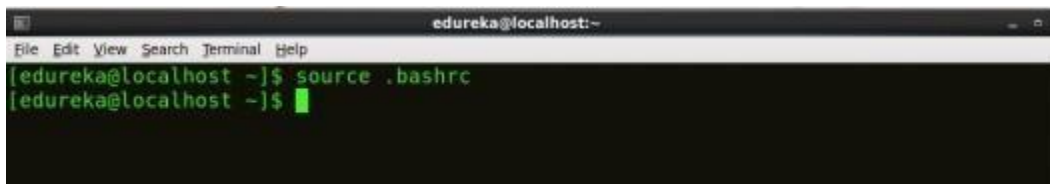
```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ vi .bashrc  
  
# User specific aliases and functions  
  
export HADOOP_HOME=$HOME/hadoop-2.7.3  
export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop  
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.7.3  
export HADOOP_COMMON_HOME=$HOME/hadoop-2.7.3  
export HADOOP_HDFS_HOME=$HOME/hadoop-2.7.3  
export YARN_HOME=$HOME/hadoop-2.7.3  
export PATH=$PATH:$HOME/hadoop-2.7.3/bin  
  
# Set JAVA_HOME  
  
export JAVA_HOME=/home/edureka/jdk1.8.0_101  
export PATH=/home/edureka/jdk1.8.0_101/bin:$PATH
```

*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



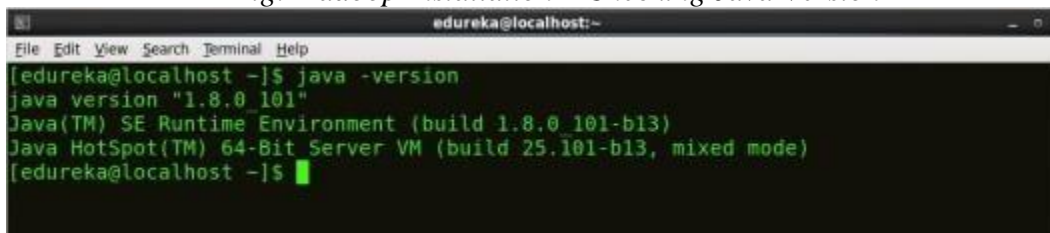
```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ source .bashrc  
[edureka@localhost ~]$
```

*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*



```
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:~$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
[edureka@localhost ~]$
```

*Fig: Hadoop Installation – Checking Hadoop Version*

**Step 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:

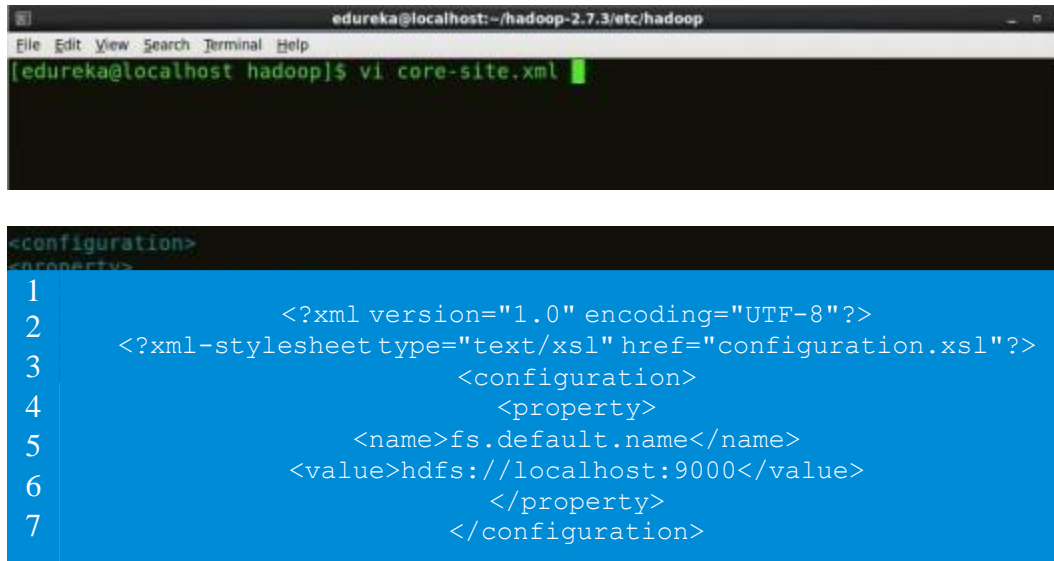
```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh              mapred-env.sh
configuration.xml           httpfs-log4j.properties   mapred-queues.xml.template
container-executor.cfg      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



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ vi core-site.xml

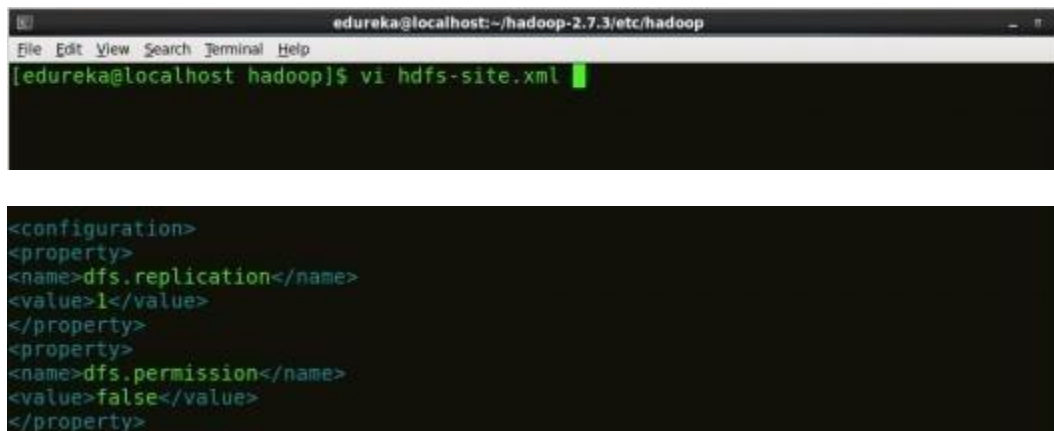
1      <?xml version="1.0" encoding="UTF-8"?>
2      <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3          <configuration>
4              <property>
5                  <name>fs.default.name</name>
6                  <value>hdfs://localhost:9000</value>
7              </property>
   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



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ 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*

```

1
2      <?xml version="1.0" encoding="UTF-8"?>
3 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
4      <configuration>
5          <property>
6              <name>dfs.replication</name>
7              <value>1</value>
8          </property>
9          <property>
10             <name>dfs.permission</name>
11             <value>>false</value>
12         </property>
13     </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* template.

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

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

```

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

```

```

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

```

```

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

```

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

```

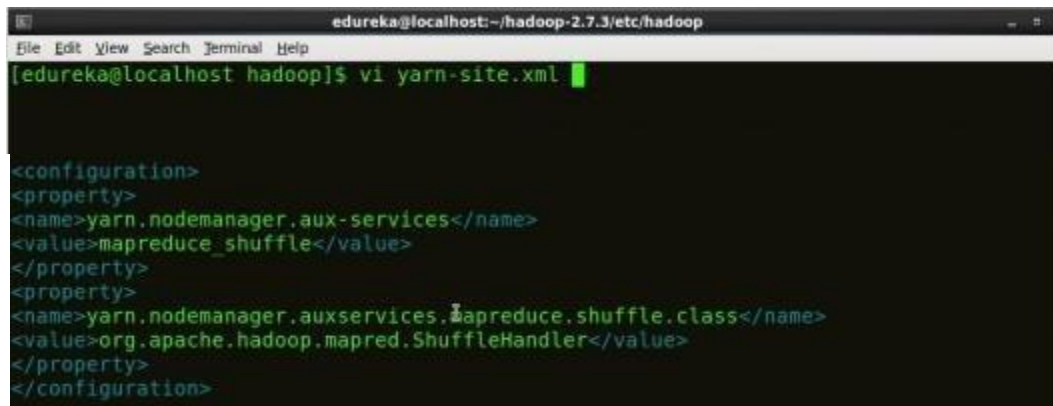
1      <?xml version="1.0" encoding="UTF-8"?>
2      <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3      <configuration>
4          <property>
5              <name>mapreduce.framework.name</name>
6              <value>yarn</value>
7              </property>
          </configuration>

```

**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



```

edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ vi yarn-site.xml
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>

```

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

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

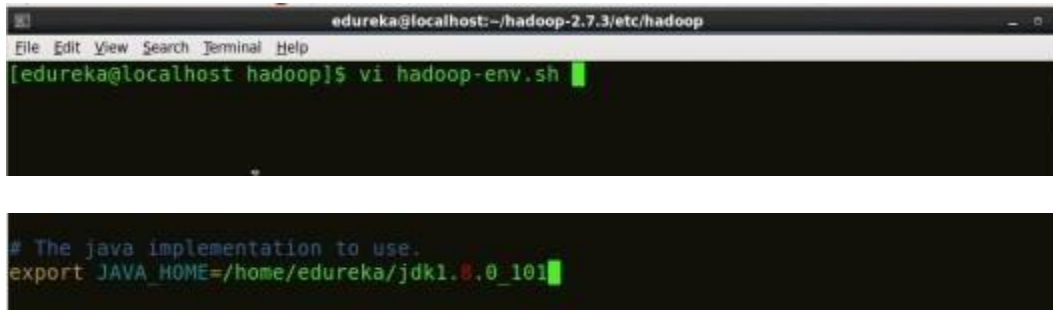
```

1
2      <?xml version="1.0">
3      <configuration>
4          <property>
5              <name>yarn.nodemanager.aux-services</name>
6              <value>mapreduce_shuffle</value>
7              </property>
8          <property>
9              <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</
10             name>
11             <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



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ vi hadoop-env.sh

# The java implementation to use.
export JAVA_HOME=/home/edureka/jdk1.8.0_101
```

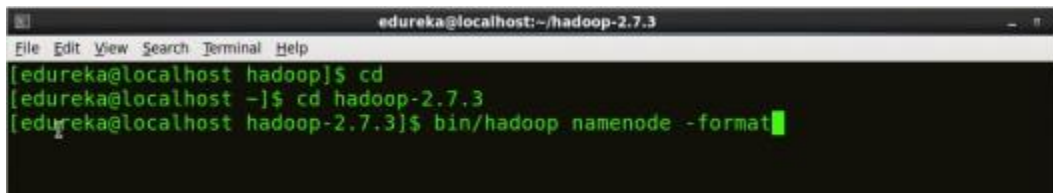
*Fig: Hadoop Installation – Configuring hadoop-env.sh Step*

**12:** Go to Hadoop home directory and format the NameNode.

**Command:** cd

**Command:** cd hadoop-2.7.3

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



```
edureka@localhost:~/hadoop-2.7.3
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ cd
[edureka@localhost ~]$ cd hadoop-2.7.3
[edureka@localhost hadoop-2.7.3]$ 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

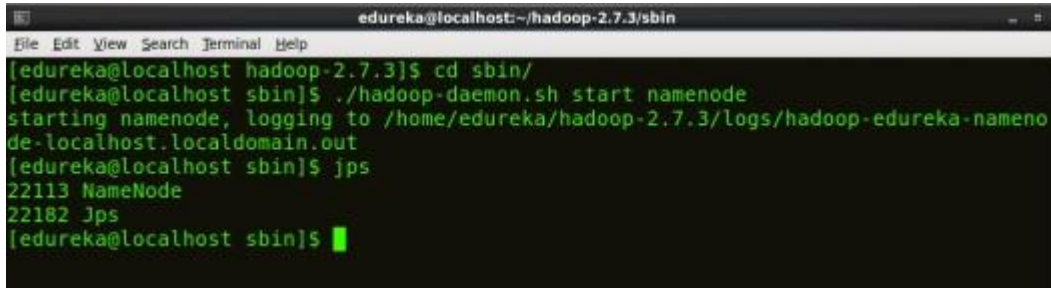
The above command is a combination of *start-dfs.sh*, *start-yarn.sh* & *mr-jobhistory-daemon.sh*

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`

A terminal window titled 'edureka@localhost:~/hadoop-2.7.3/sbin' showing the process of starting the NameNode. The user navigates to the 'sbin' directory and runs './hadoop-daemon.sh start namenode', which outputs 'starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-namenode-localhost.localdomain.out'. Then, the user runs 'jps', which outputs '22113 NameNode' and '22182 Jps'.

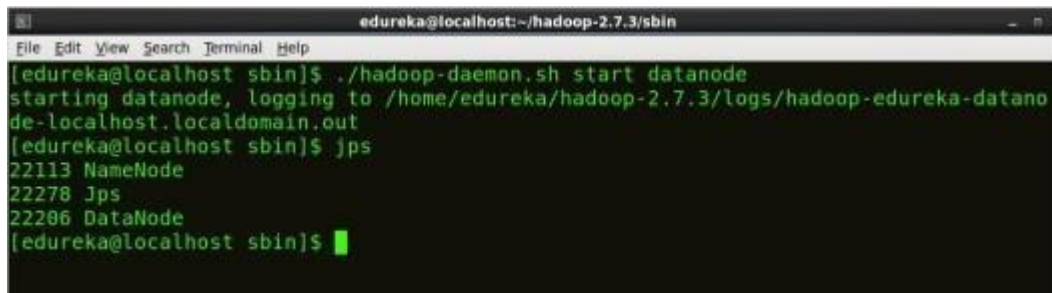
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost hadoop-2.7.3]$ cd sbin/
[edureka@localhost sbin]$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-namenode-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22182 Jps
[edureka@localhost sbin]$
```

*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`



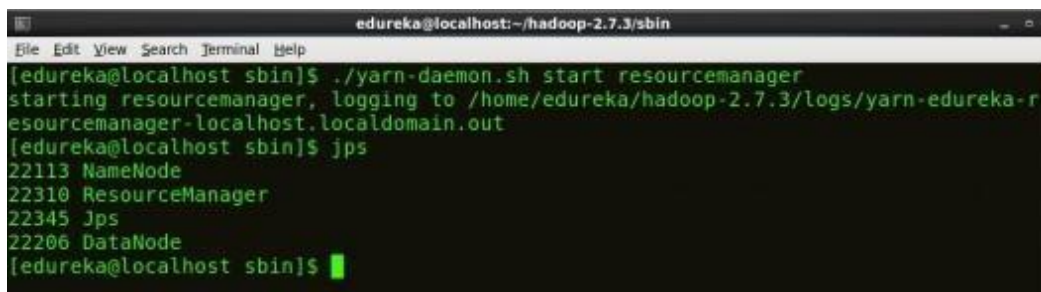
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datano
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278 Jps
22206 DataNode
[edureka@localhost sbin]$
```

*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`



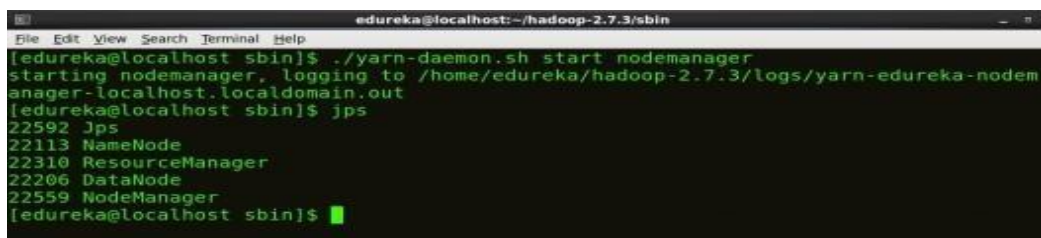
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-r
esourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22345 Jps
22206 DataNode
[edureka@localhost sbin]$
```

*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`



```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-nodem
anager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22592 Jps
22113 NameNode
22310 ResourceManager
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```





[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.**

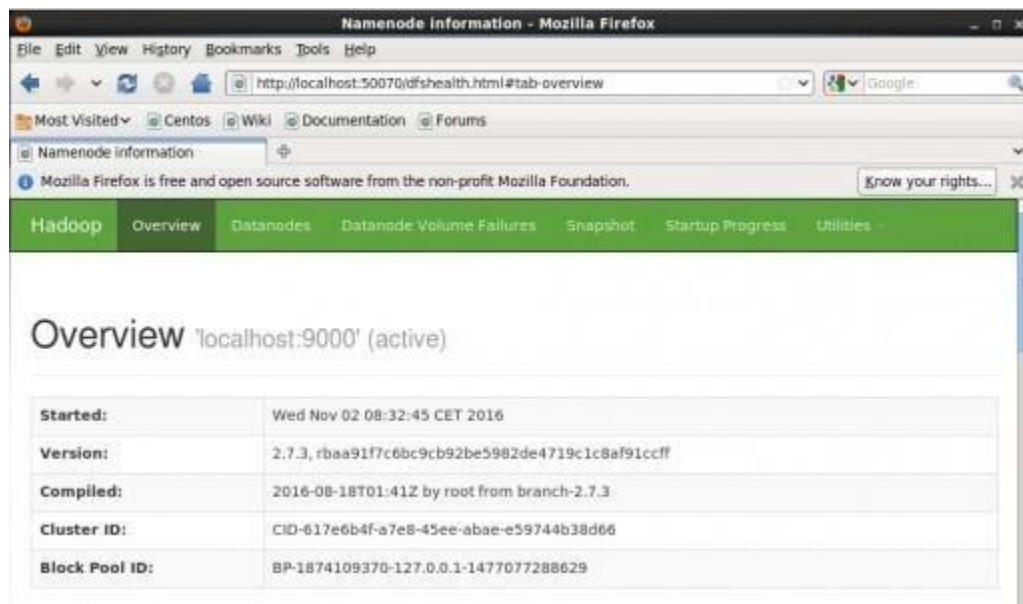
**Command:** `jps`

```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/edureka/hadoop-2.7.3/logs/mapred-edureka-h
istoryserver-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22694 JobHistoryServer
22727 Jps
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```

*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.