

EX NO. : 1

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

Aim:

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

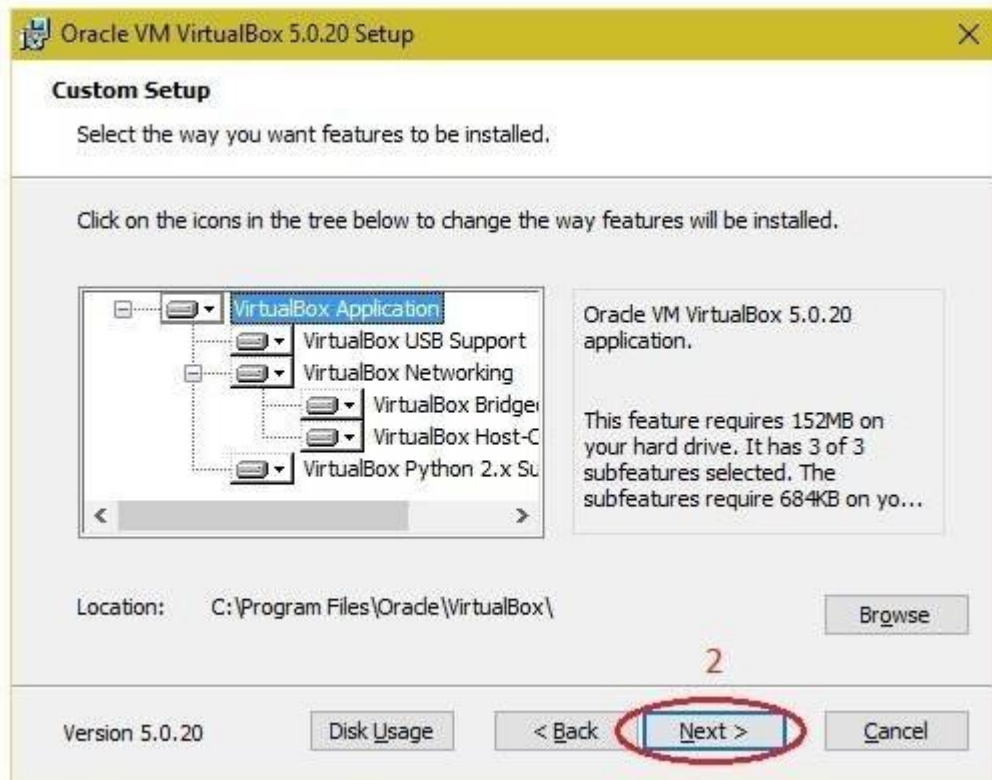
PROCEDURE:

Steps to install Virtual Box:

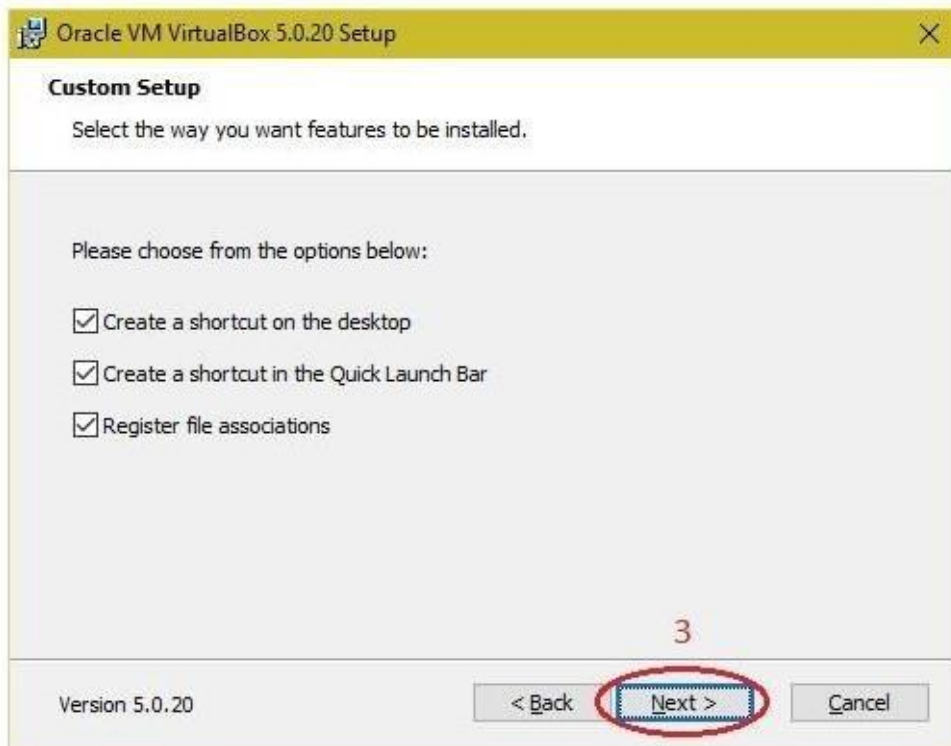
1. Download the Virtual box exe and click the exe file...and select next button..



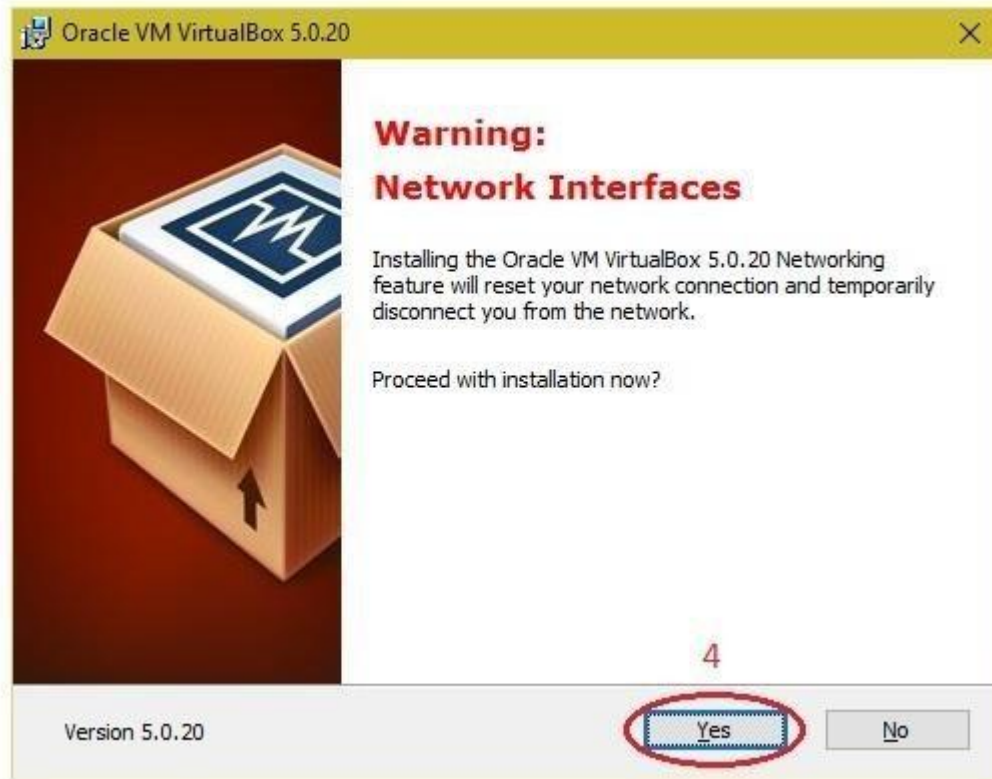
2. Click the next button



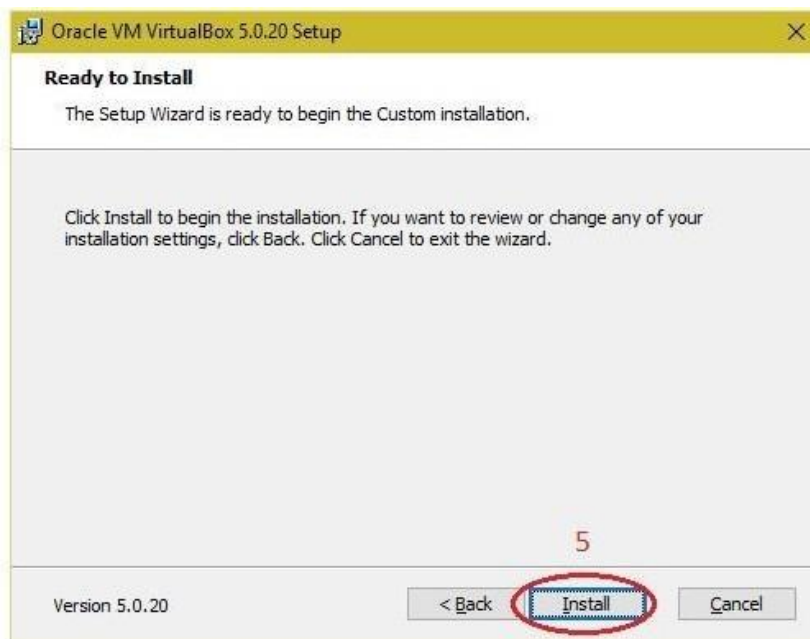
3. Click the next button



4. Click the YES button..



5. Click the install button...



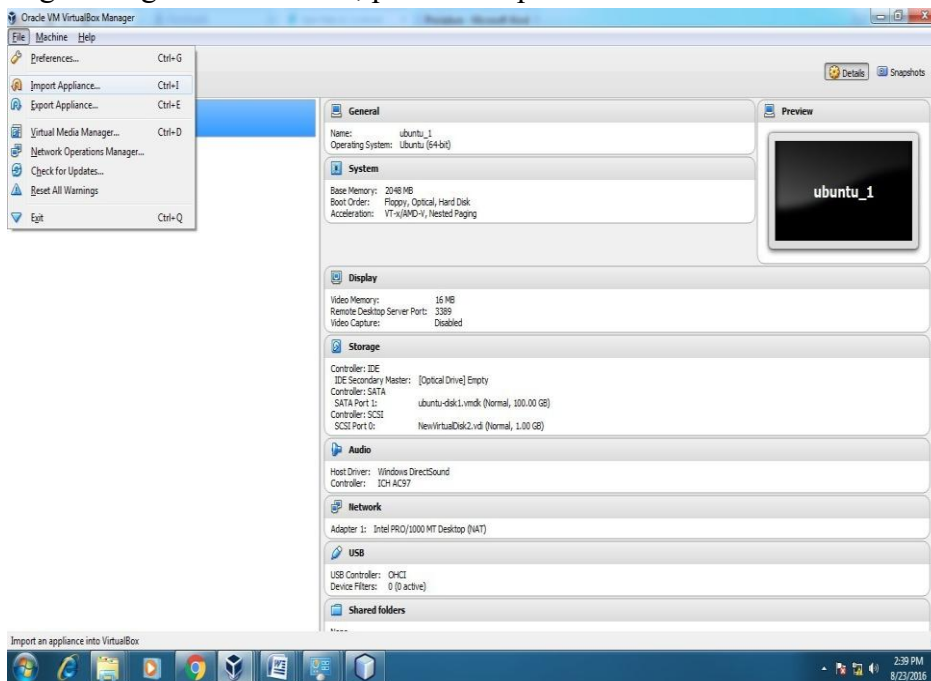
6. Then installation was completed..the show virtual box icon on desktop screen....

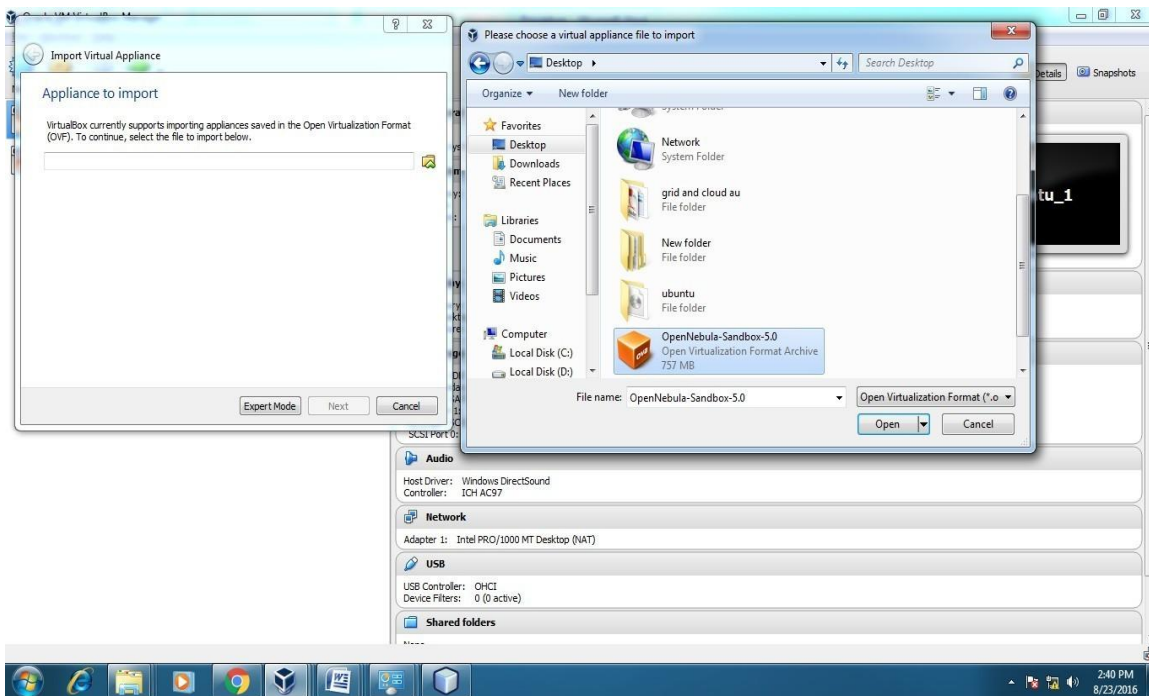


VirtualBox

Steps to import Open nebula sandbox:

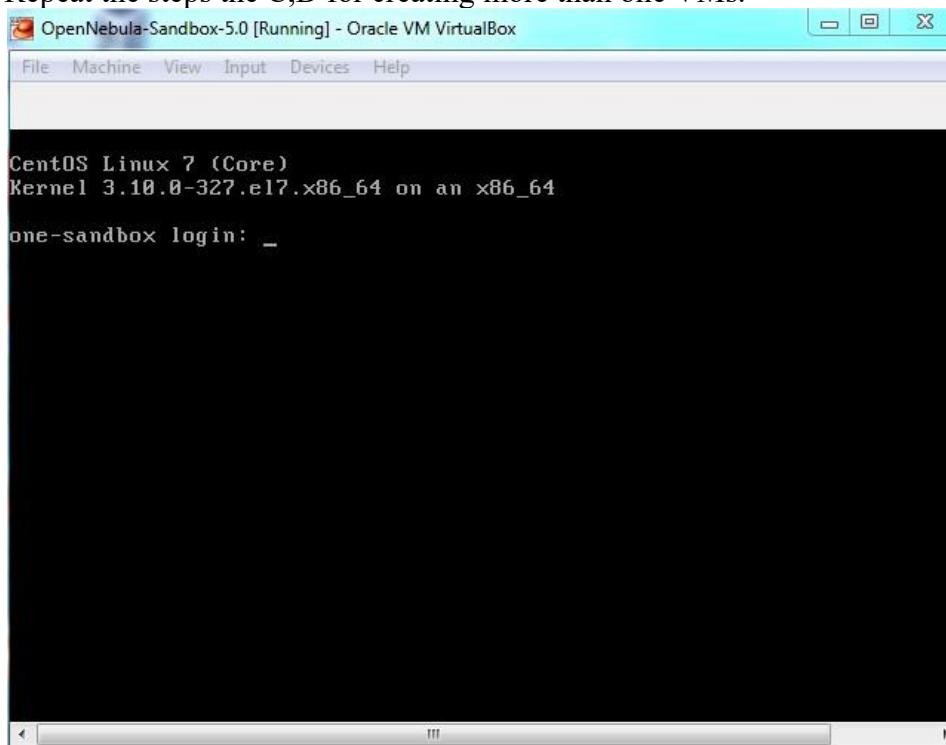
1. Open Virtual box
2. File ☰ import Appliance
3. Browse OpenNebula-Sandbox-5.0.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the Open Nebula
6. Login using username: root, password:opennebula

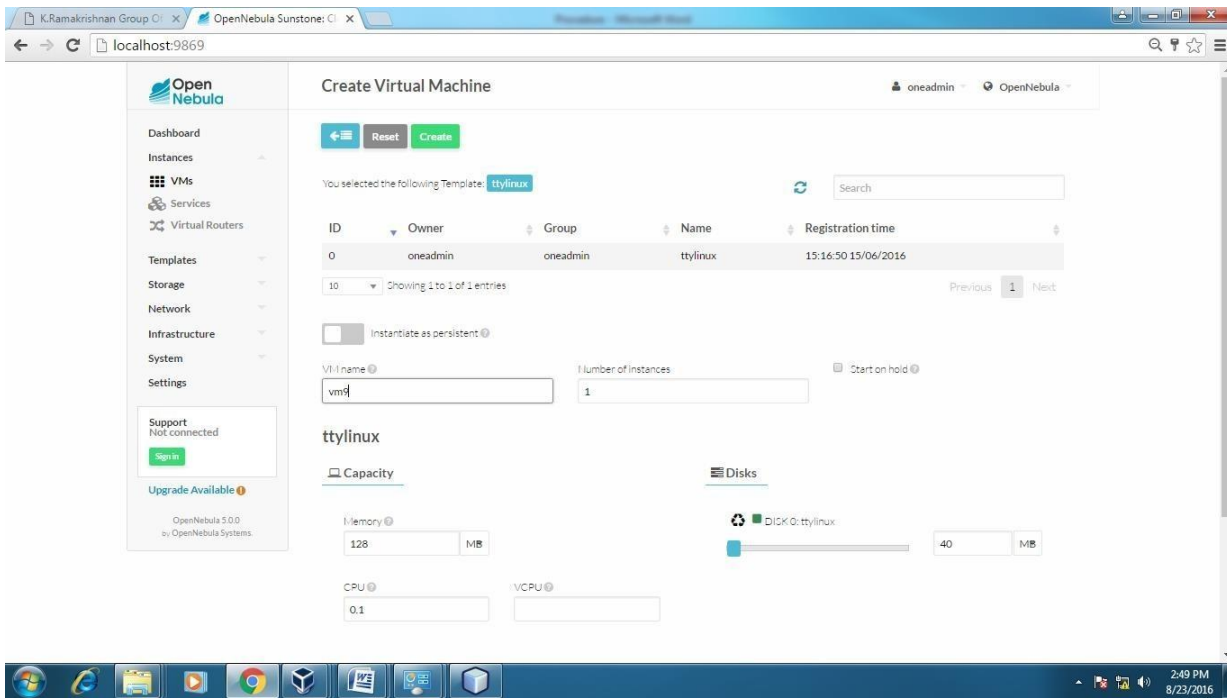
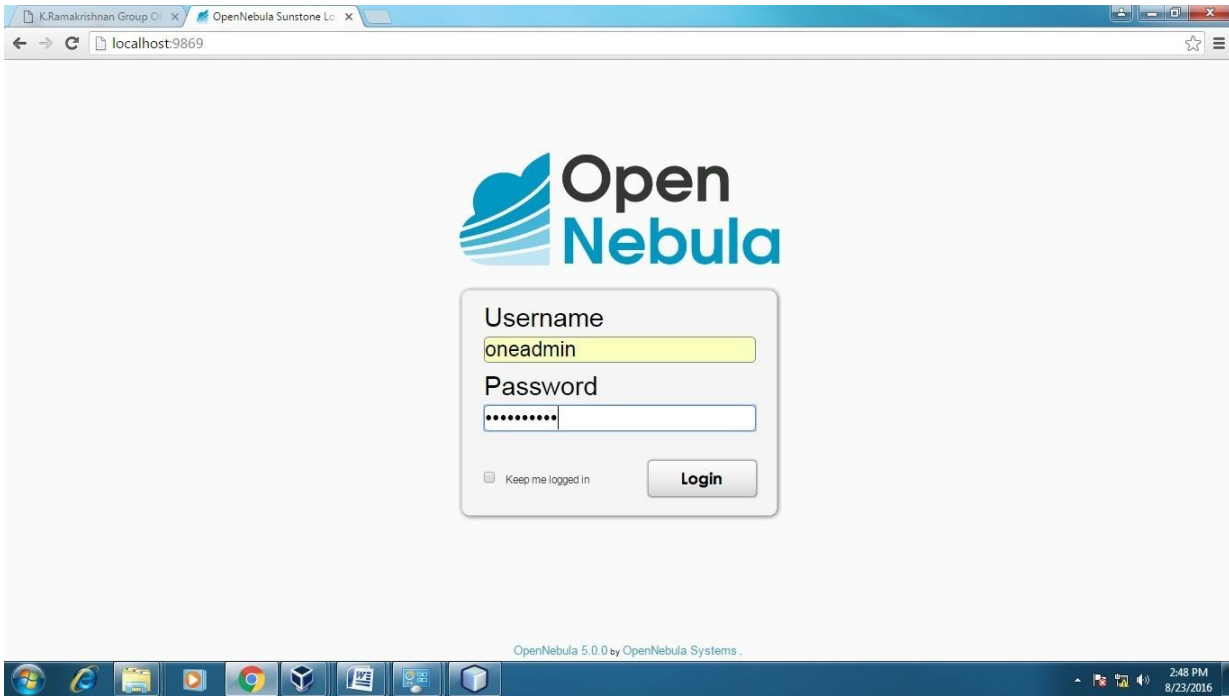




Steps to create Virtual Machine through opennebula

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.





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, linkedin.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 virtual box and execute Simple Programs

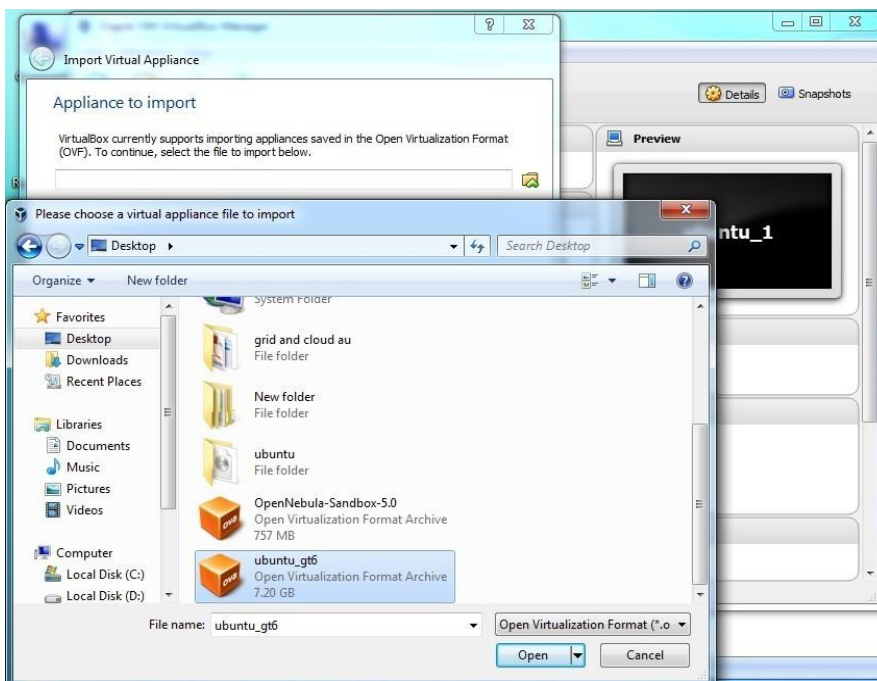
Aim:

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

PROCEDURE:

Steps to import .ova file:

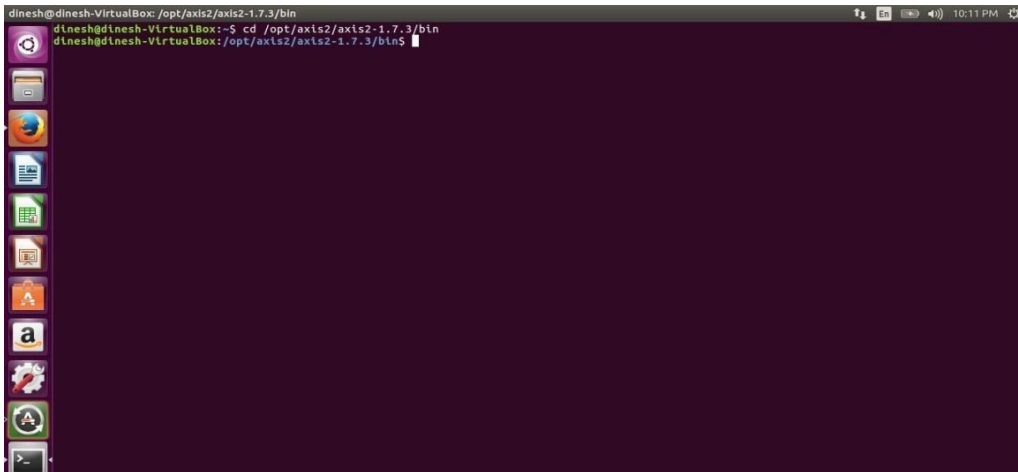
1. Open Virtual box
2. File ☰ import Appliance
3. Browse ubuntu_gt6.ova file
4. Then go to setting, select Usb and choose USB 1.1
5. Then Start the ubuntu_gt6
6. Login using username: dinesh, password:99425.



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`

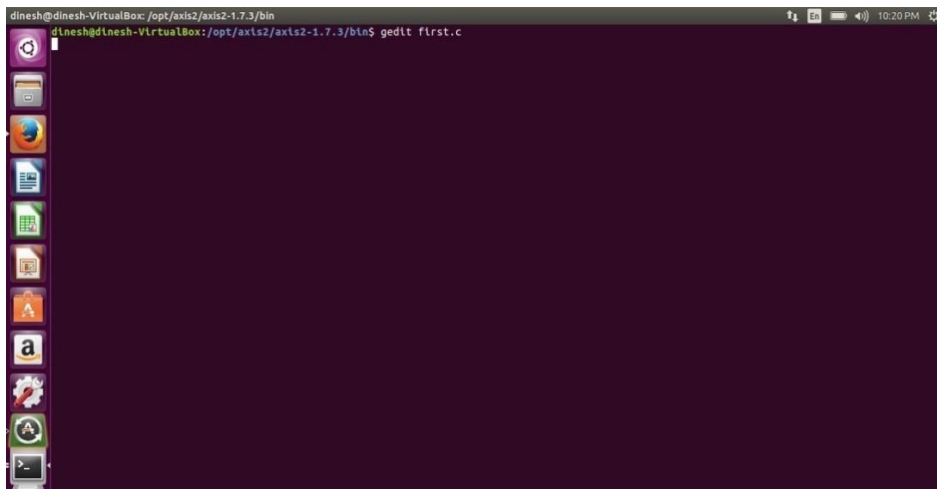
1. Type `cd /opt/axis2/axis2-1.7.3/bin` then press enter



```
dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:~$ cd /opt/axis2/axis2-1.7.3/bin
dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$
```

A terminal window screenshot showing the execution of the `cd` command. The prompt is `dinesh@dinesh-VirtualBox:~$`. The command `cd /opt/axis2/axis2-1.7.3/bin` is entered and executed. The prompt changes to `dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$`. The terminal title bar shows `dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin` and the time is 10:11 PM.


2. Type `gedit first.c`



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

A terminal window screenshot showing the execution of the `gedit` command. The prompt is `dinesh@dinesh-VirtualBox:/opt/axis2/axis2-1.7.3/bin$`. The command `gedit first.c` is entered and executed. The terminal title bar shows `dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin` and the time is 10:20 PM.

3. Type the c program

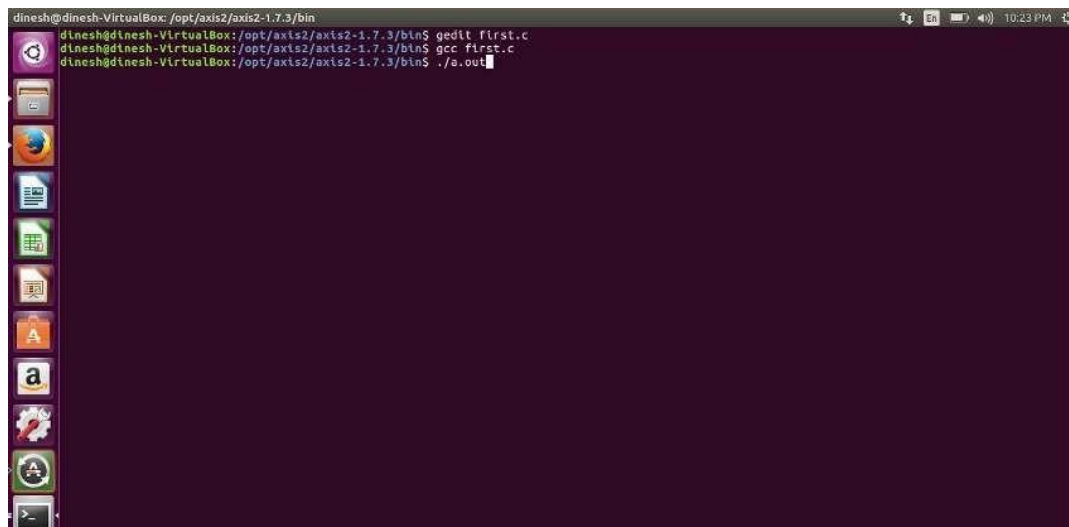


The screenshot shows a gedit editor window titled "first.c (~/) - gedit". The window contains the following C code:

```
#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'...", "C", "Tab Width: 8", "Ln 13, Col 2", and "INS".

4. Running the C program

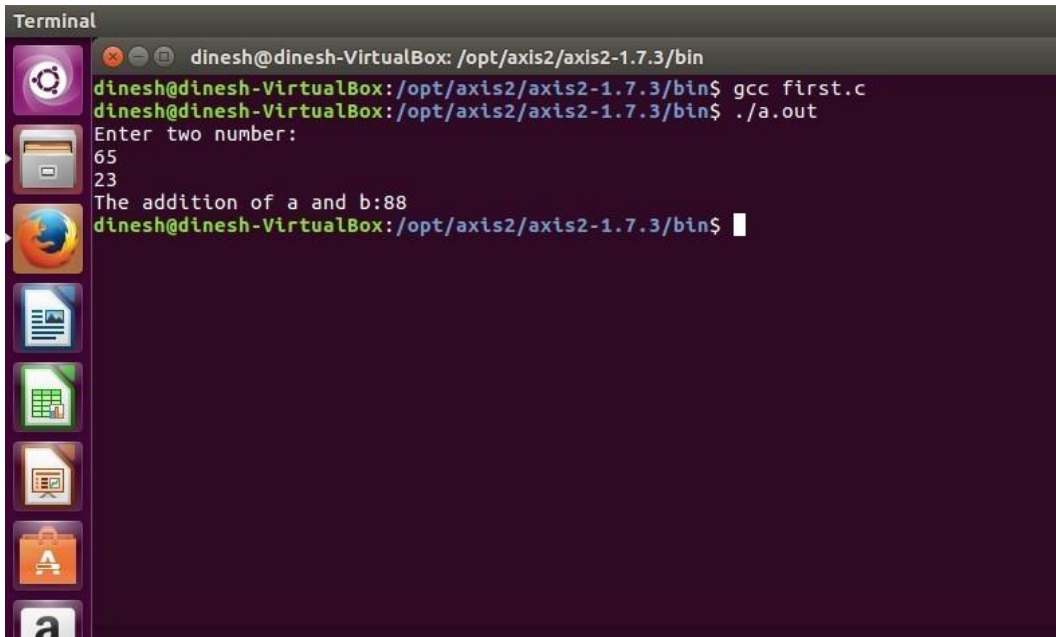


The screenshot shows a terminal window with the following commands and output:

```
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 title is "dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin". The status bar at the bottom shows "10:23 PM".

5. Display the output:

A terminal window titled "Terminal" with a dark purple background. The window title bar shows "dinesh@dinesh-VirtualBox: /opt/axis2/axis2-1.7.3/bin". The terminal content shows the following commands and output:

```
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$
```

The terminal has a vertical sidebar on the left with various application icons like a gear, a folder, a globe, a document, a spreadsheet, a presentation, a calendar, and a terminal icon.

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 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

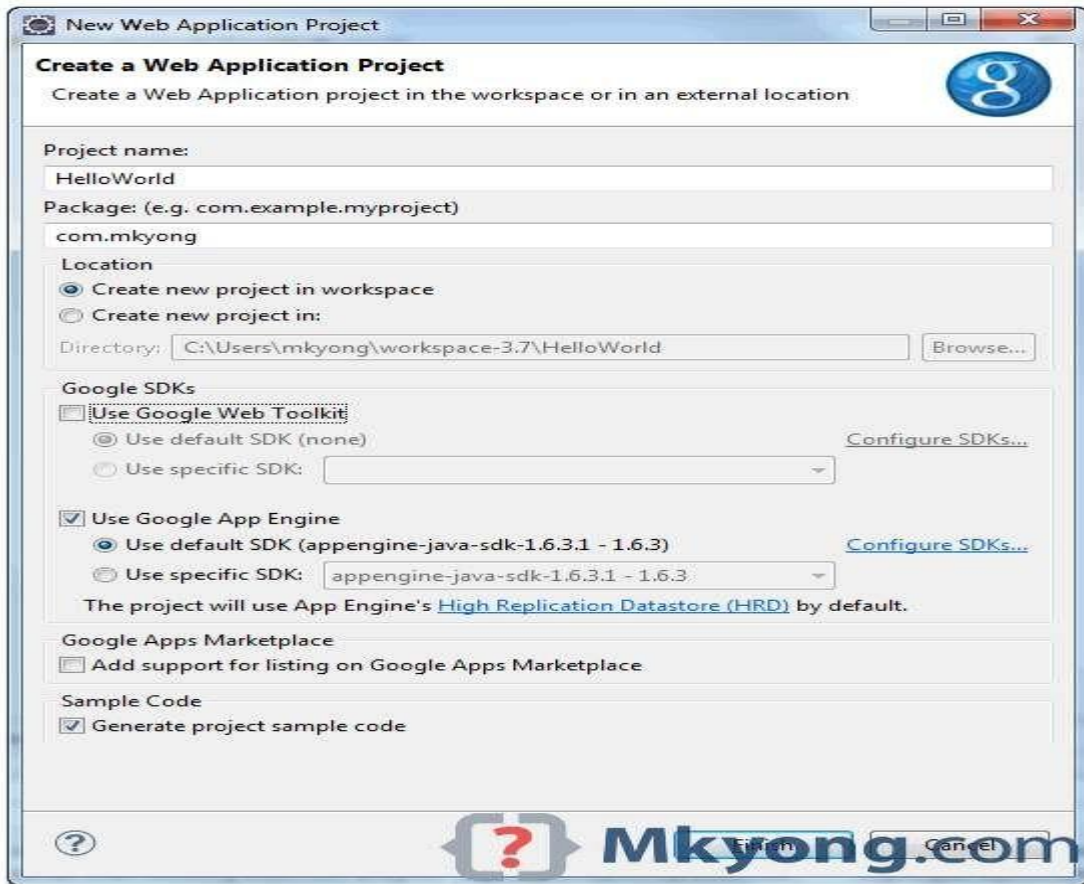
Read this guide – [how to install Google Plugin for Eclipse](#). If you install the Google App Engine Java SDK together with “**Google Plugin for Eclipse**“, then go to step 2, Otherwise, get the [Google App Engine Java SDK](#) and extract it.

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

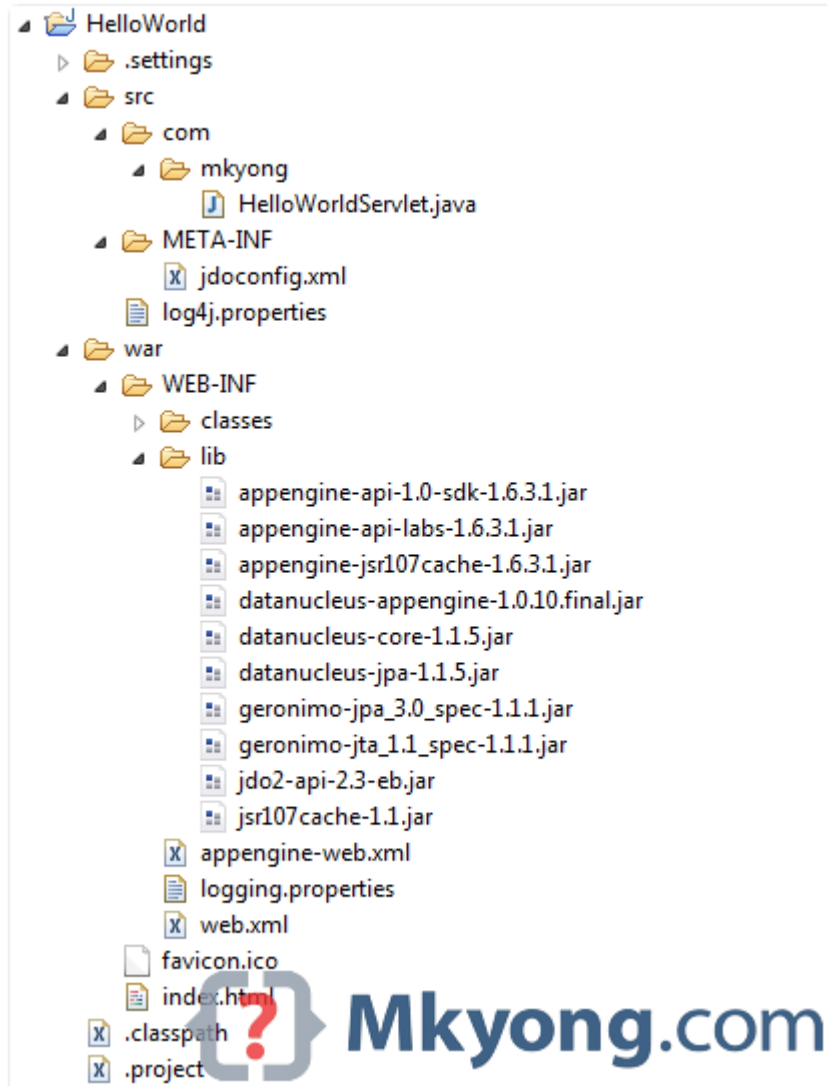
*Figure – Deselect the “**Google Web Toolkit**“, and link your GAE Java SDK via the “**configure SDK**” link.*



Click

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/

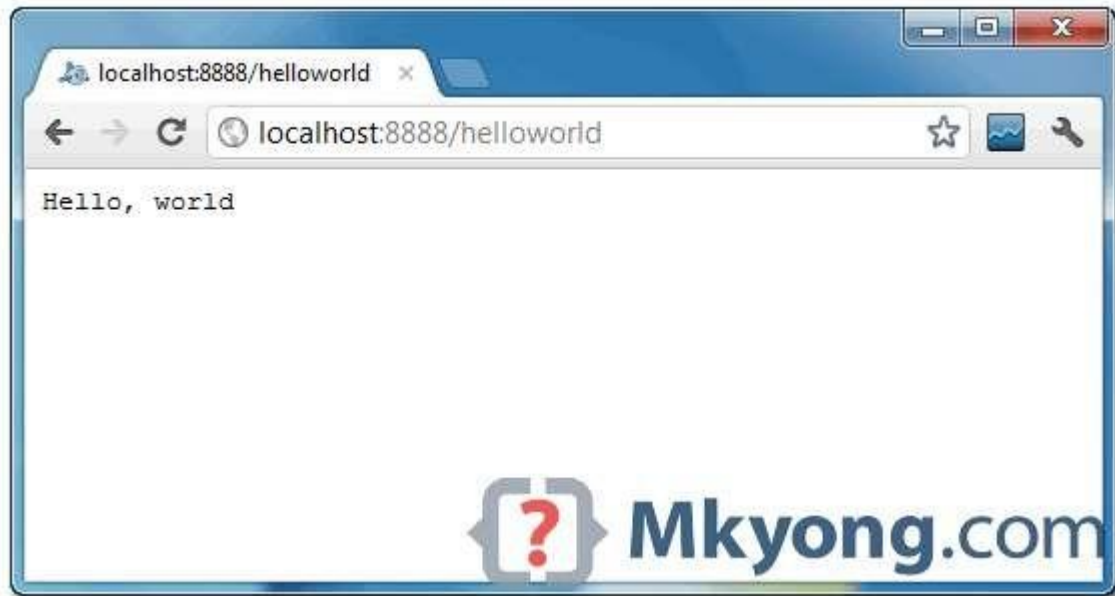
30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl start INFO: The admin console is running at http://localhost:8888/_ah/admin

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.

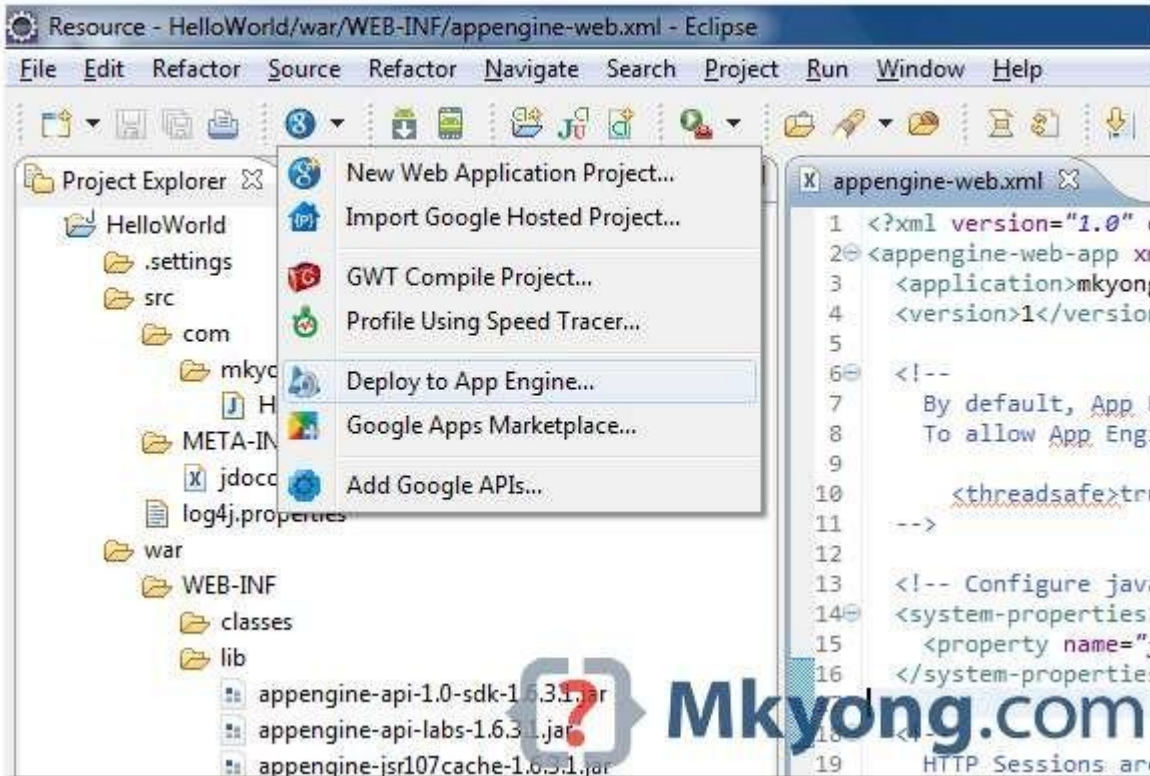


Figure 1.2 – Sign in with your Google account and click on the Deploy button.

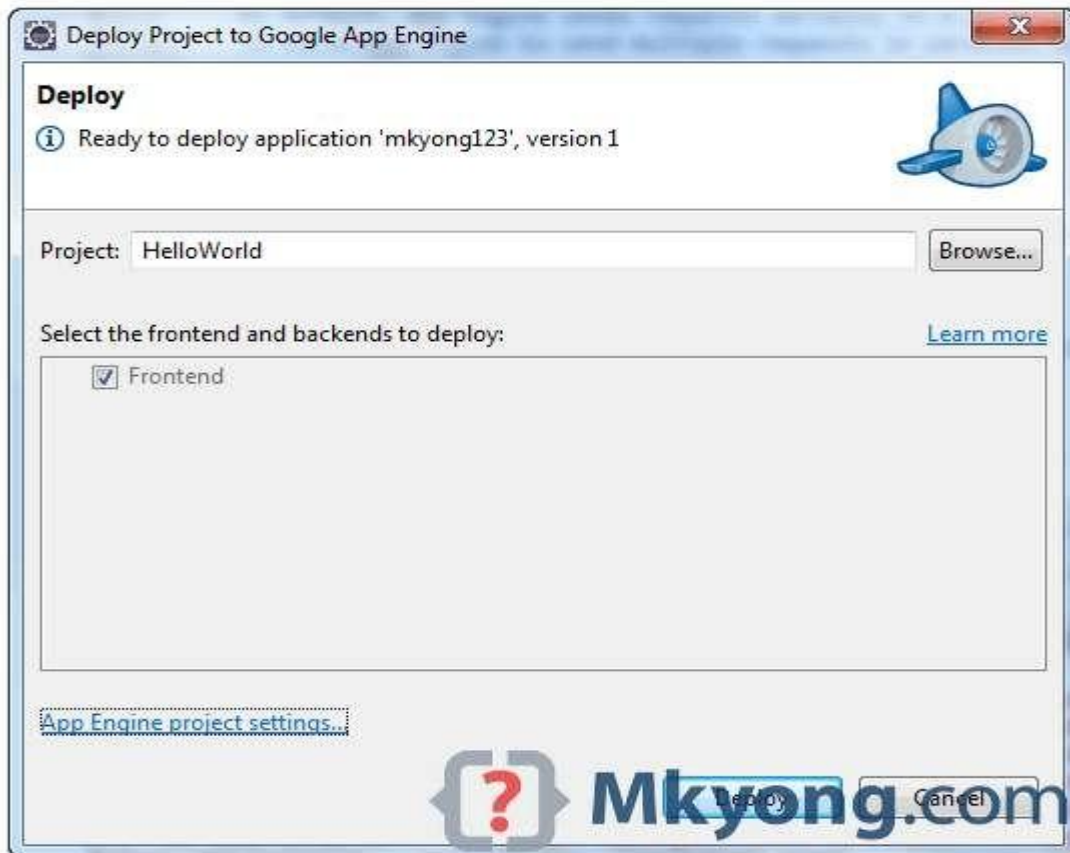


Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – <http://mkyong123.appspot.com/>



Result:

Thus the simple application was created successfully.

EX. NO.:4 Simulate a cloud scenario using CloudSim and run a scheduling 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())
```


===== OUTPUT =====

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

*****Datacenter:

Datacenter_0*****	User id
Debt	
3	35.6

CloudSimExample1 finished!

RESULT:

The simulation was successfully executed.

EX.NO.:5

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), create a file called app.yaml in the ae-01-trivial folder with the following contents:

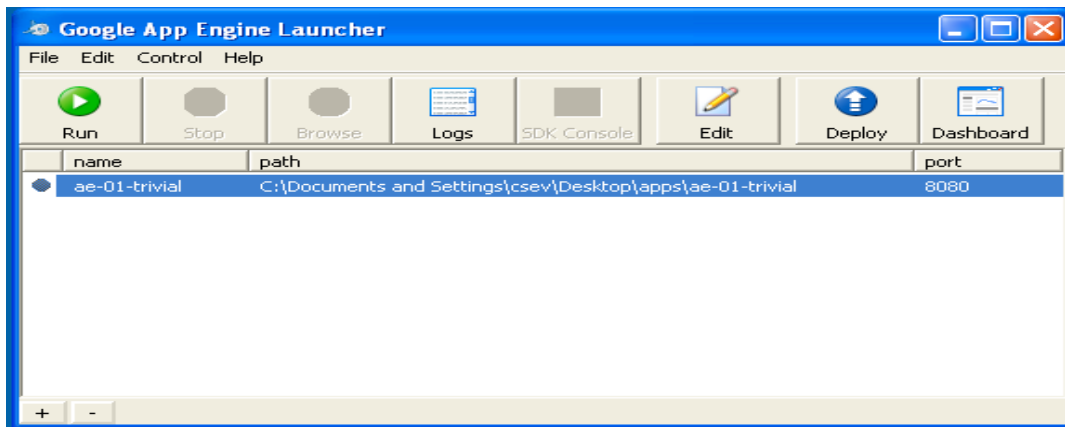
```
application: ae-01-trivial
version: 1
runtime: python api_version: 1
handlers:- url: /*
script: index.py
```

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

Watching the Log

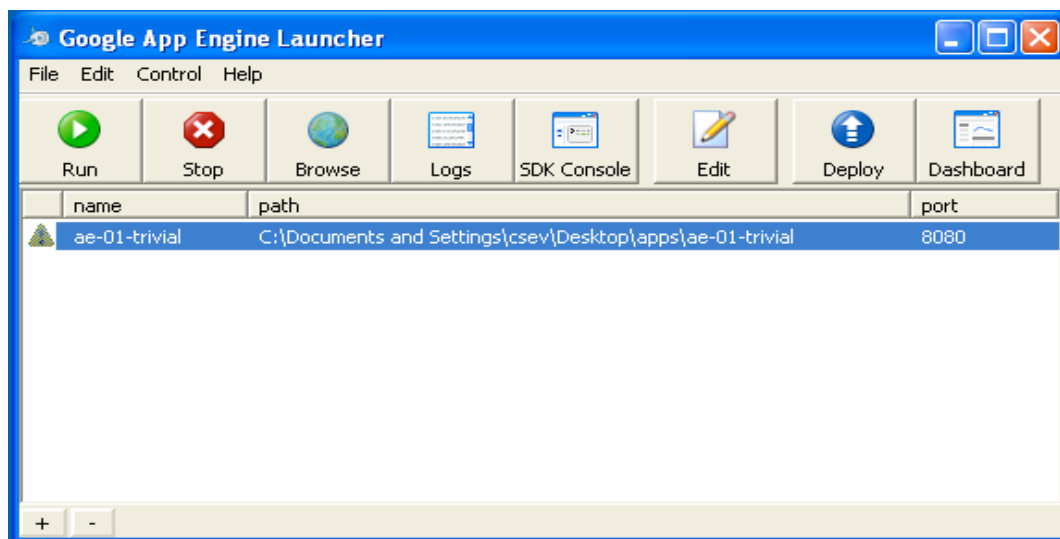
You can watch the internal log of the actions that the web server is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the Logs button to bring up a log window:

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

```
Log Console (ae-01-trivial)
WARNING 2010-03-13 18:03:13,796 datastore_file_stub.py:623] Could not read
datastore data from c:\docume~1\csev\locals~1\temp\dev_appserver.datastore
WARNING 2010-03-13 18:03:13,796 dev_appserver.py:3581] Could not initialize
images API; you are likely missing the Python "PIL" module. ImportError: No module
named _imaging
INFO 2010-03-13 18:03:13,828 dev_appserver_main.py:399] Running application
ae-01-trivial on port 8080: http://localhost:8080
INFO 2010-03-13 18:03:24,717 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO 2010-03-13 18:03:24,733 dev_appserver_index.py:205] Updating C:\Documents
and Settings\csev\Desktop\apps\ae-01-trivial\index.yaml
INFO 2010-03-13 18:03:24,967 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
2010-03-13 13:03:30 (Process exited with code -1)
```

Dealing With Errors

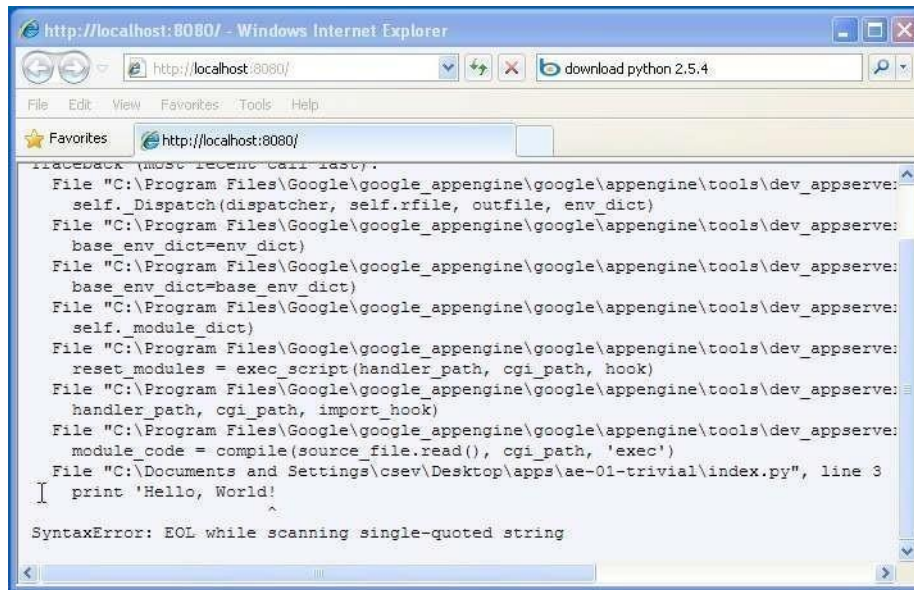
With two files to edit, there are two general categories of errors that you may encounter. If you make a mistake on the app.yaml file, the App Engine will not start and your launcher will show a yellow icon near your application:



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

```
Log Console (ae-01-trivial)
Invalid object:
Unknown url handler type.
<URLMap
  static_dir=None
  secure=default
  script=None
  url=.*
  static_files=None
  upload=None
  mime_type=None
  login=optional
  require_matching_file=None
  auth_fall_action=redirect
  expiration=None
>
in "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\app.yaml", line 8,
column 1
```

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.



```
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

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: 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 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 overcome 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

The screenshot shows the OpenNebula web interface for Host 1 naveenkumar. The interface includes a sidebar with navigation options like Dashboard, Instances, VMs, Services, and Virtual Routers. The main content area displays a table of VMs with the following data:

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

The table shows 6 entries, with the current view displaying 1 to 6 of 6 entries. The status for all VMs is FAILURE. The interface also includes a search bar and navigation controls like 'Previous' and 'Next'.

Host:one-sandbox

The screenshot shows the OpenNebula web interface for Host 0 one-sandbox. The interface includes a sidebar with navigation options like Dashboard, Instances, VMs, Services, and Virtual Routers. The main content area displays a table of VMs with the following data:

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm0	RUNNING	one-sandbox	172.16.100.207
6	oneadmin	oneadmin	vm0	RUNNING	one-sandbox	172.16.100.206

The table shows 2 entries, with the current view displaying 1 to 2 of 2 entries. The status for both VMs is RUNNING. The interface also includes a search bar and navigation controls like 'Previous' and 'Next'.

VMs

Migrate Virtual Machine

VM 6 vm8 is currently running on Host one-sandbox
 VM 7 vm8 is currently running on Host one-sandbox

Select a Host

Please select a Host from the list

ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
0	naive	default	0	0 / 0	0 / 0	RETRY
1	naveenkumar	name	0	82 / 0	44 / 0	ERROR
0	one-sandbox	name	2	20 / 200 (10%)	4 / 10 / 7-4 / 10 (14%)	OK

Showing 1 to 3 of 3 entries

Advanced Options

Migrate

2:35 PM 8/23/2016

VMs

ID	Owner	Group	Name	Status	Host	IPs
<input checked="" type="checkbox"/>	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.207
<input checked="" type="checkbox"/>	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.206
<input type="checkbox"/>	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
<input type="checkbox"/>	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
<input type="checkbox"/>	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
<input type="checkbox"/>	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
<input type="checkbox"/>	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
<input type="checkbox"/>	oneadmin	oneadmin	ttlinux-0	FAILURE	naveenkumar	172.16.100.200

Showing 1 to 8 of 8 entries

8 TOTAL 2 ACTIVE 0 OFF 0 PENDING 6 FAILED

Support Not connected

Upgrade Available

2:35 PM 8/23/2016

After Migration:

The screenshot shows the OpenNebula Sunstone interface. The left sidebar contains navigation menus for Dashboard, Instances, VMs, Services, Virtual Routers, Templates, Storage, Network, Infrastructure, Clusters, Hosts, Zones, System, and Settings. The main content area is titled "Hosts" and displays a table of host information. The table has columns for ID, Name, Cluster, RVMs, Allocated CPU, Allocated MEM, and Status. Below the table, there is a summary: 3 TOTAL, 1 ON, 0 OFF, 2 ERROR. The taskbar at the bottom shows the system time as 2:36 PM on 8/23/2016.

ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
2	raa	default	0	0 / 0	0KB / -	ERROR
1	naveenkumar	rama	8	82 / 0	481KB / -	ERROR
0	one-sandbox	rama	0	0 / 100 (0%)	0KB / 7411KB (0%)	ON

Showing 1 to 3 of 3 entries

3 TOTAL 1 ON 0 OFF 2 ERROR

Host:one-sandbox

The screenshot shows the OpenNebula Sunstone interface for a specific host. The left sidebar is the same as in the previous screenshot. The main content area is titled "Host: one-sandbox" and features a summary card with tabs for Info, Graphs, VMs, Wilds, and Zombies. The "VMs" tab is selected, showing a table with columns for ID, Owner, Group, Name, Status, Host, and IPs. The table is empty, and a message states "There is no data available". The taskbar at the bottom shows the system time as 2:37 PM on 8/23/2016.

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

Showing 0 to 0 of 0 entries

Host:SACET

The screenshot shows the OpenNebula web interface for host 'naveenkumar'. The interface includes a sidebar with navigation options like Dashboard, Instances, VMs, Services, and Virtual Routers. The main content area displays a table of VMs with columns for ID, Owner, Group, Name, Status, Host, and IPs. All VMs listed are in a 'FAILURE' state.

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	FAILURE	naveenkumar	172.16.100.207
6	oneadmin	oneadmin	vm8	FAILURE	naveenkumar	172.16.100.206
5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
0	oneadmin	oneadmin	tt/linux-0	FAILURE	naveenkumar	172.16.100.200

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.

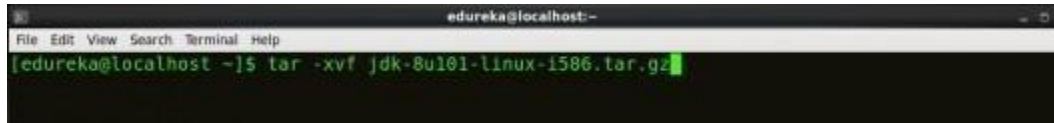
Steps:

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

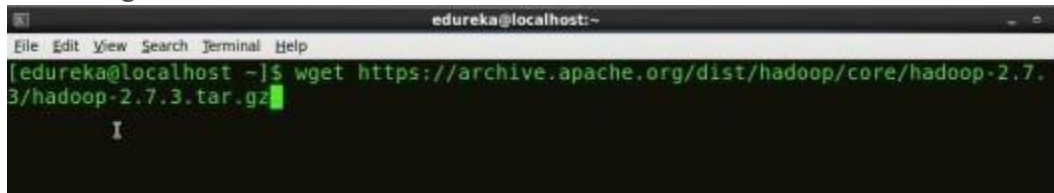


```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ tar -xvf jdk-8u101-linux-i586.tar.gz
```

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

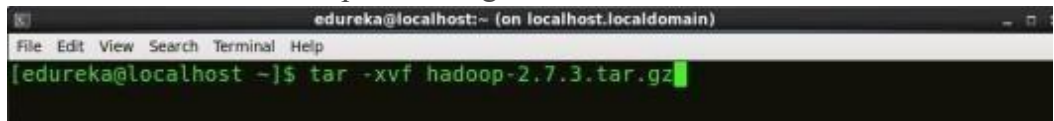


```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Downloading Hadoop

Step 4: Extract the Hadoop tar File.

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

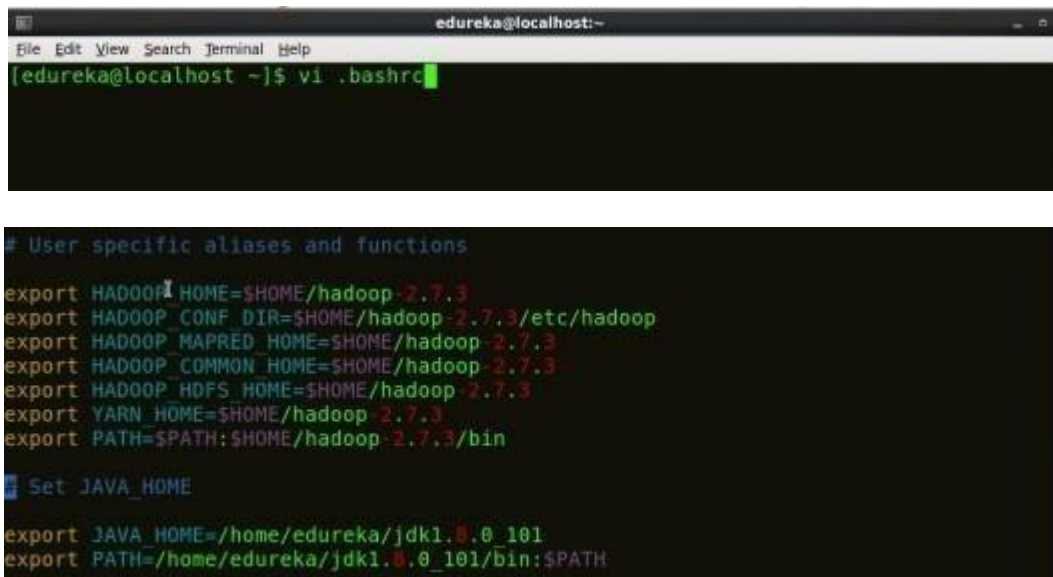


```
edureka@localhost:~ (on localhost.localdomain)  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ tar -xvf hadoop-2.7.3.tar.gz
```

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



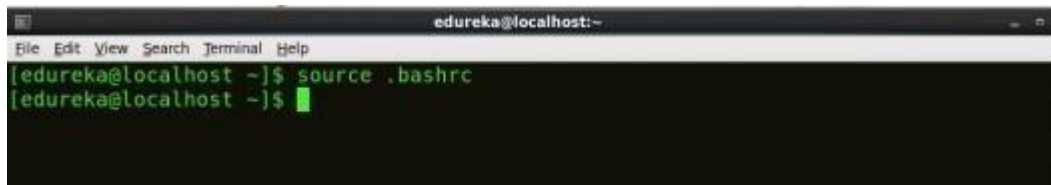
```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ vi .bashrc  
  
# User specific aliases and functions  
  
export HADOOP_HOME=/home/edureka/hadoop-2.7.3  
export HADOOP_CONF_DIR=/home/edureka/hadoop-2.7.3/etc/hadoop  
export HADOOP_MAPRED_HOME=/home/edureka/hadoop-2.7.3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



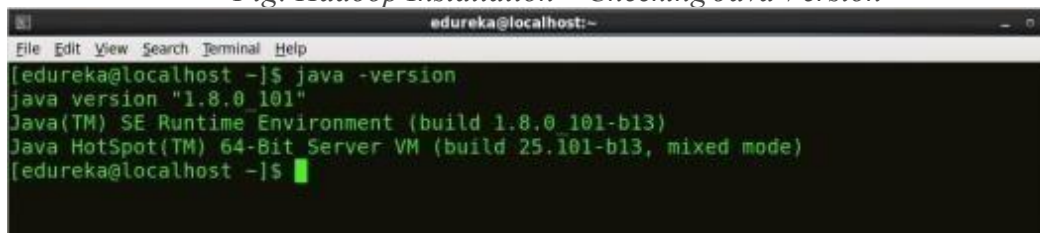
```
edureka@localhost:~  
File Edit View Search Terminal Help  
[edureka@localhost ~]$ source .bashrc  
[edureka@localhost ~]$
```

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 2.5.8
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:

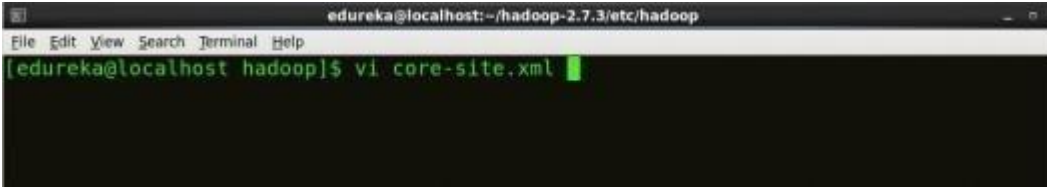
```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh              mapred-env.sh
configuration.xml          httpfs-log4j.properties   mapred-queues.xml.template
container-executor.cfg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

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
[edureka@localhost hadoop]$ vi core-site.xml
```

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

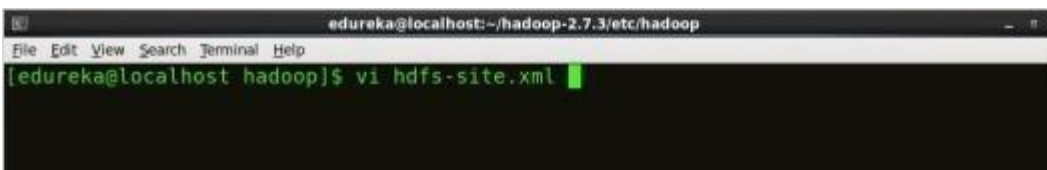
```
4         <configuration>
5             <property>
6                 <name>fs.default.name</name>
7                 <value>hdfs://localhost:9000</value>
```

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
[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>
```

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

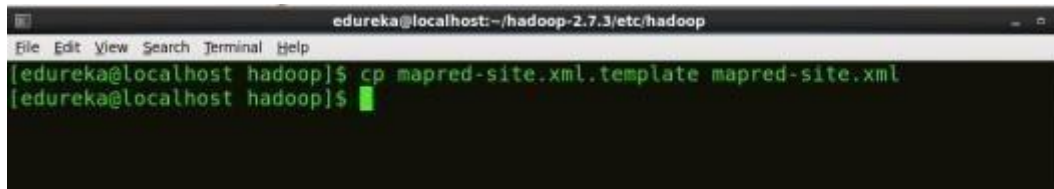
inside configuration tag:

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.

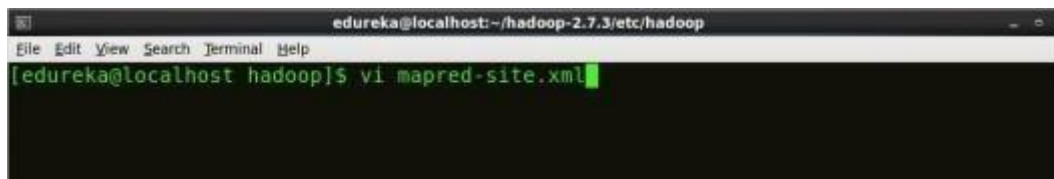
In some cases, *mapred-site.xml* file is not available. So, we have to create the *mapred-site.xml* file using *mapred-site.xml* template.

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

Command: `vi mapred-site.xml`.



```
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:

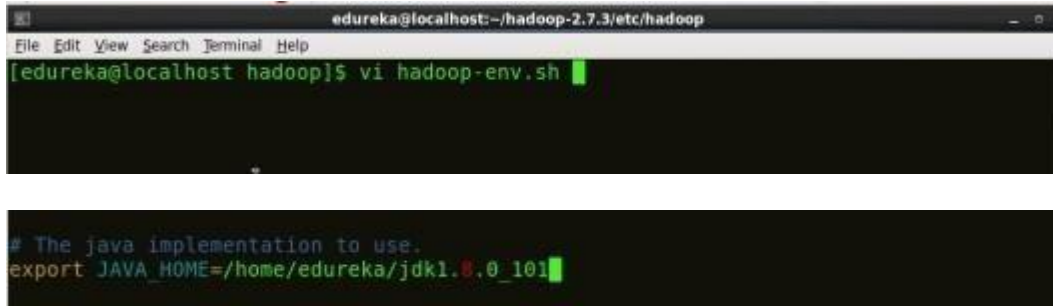
```

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</ name>
10                 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
11             </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
```

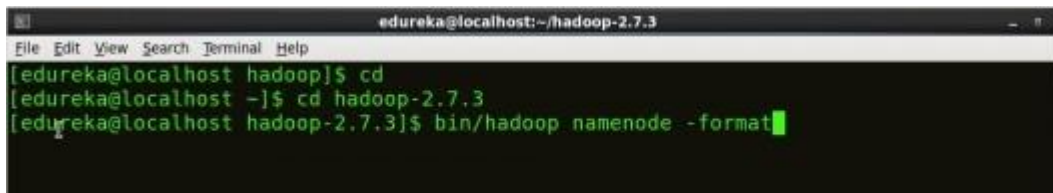
Fig: Hadoop Installation – Configuring hadoop-env.sh Step

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

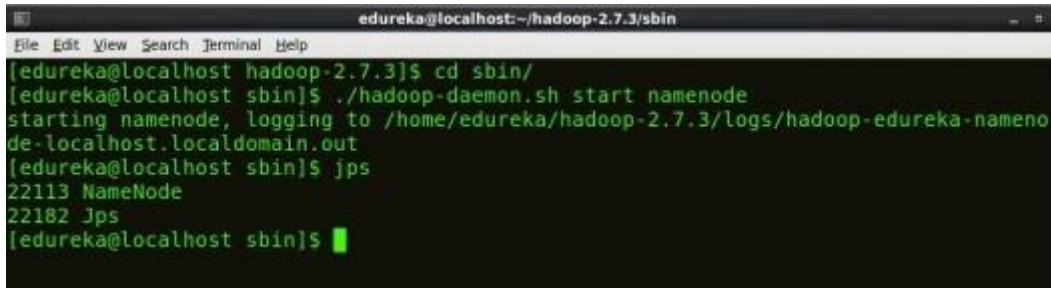
The above command is a combination of *start-dfs.sh*, *start-yarn.sh* & *mr-jobhistory-daemon.sh*

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`



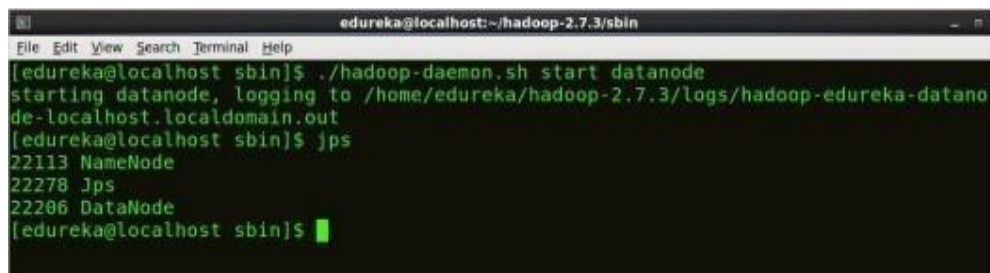
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost hadoop-2.7.3]$ cd sbin/
[edureka@localhost sbin]$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-nameno
de-localhost.localdomain.out
[edureka@localhost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



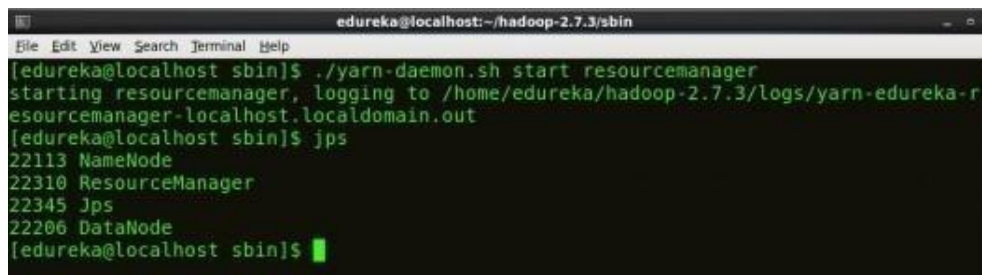
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datano
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278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



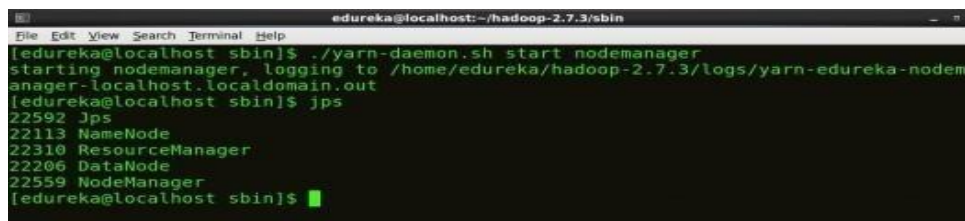
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-r
esourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113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

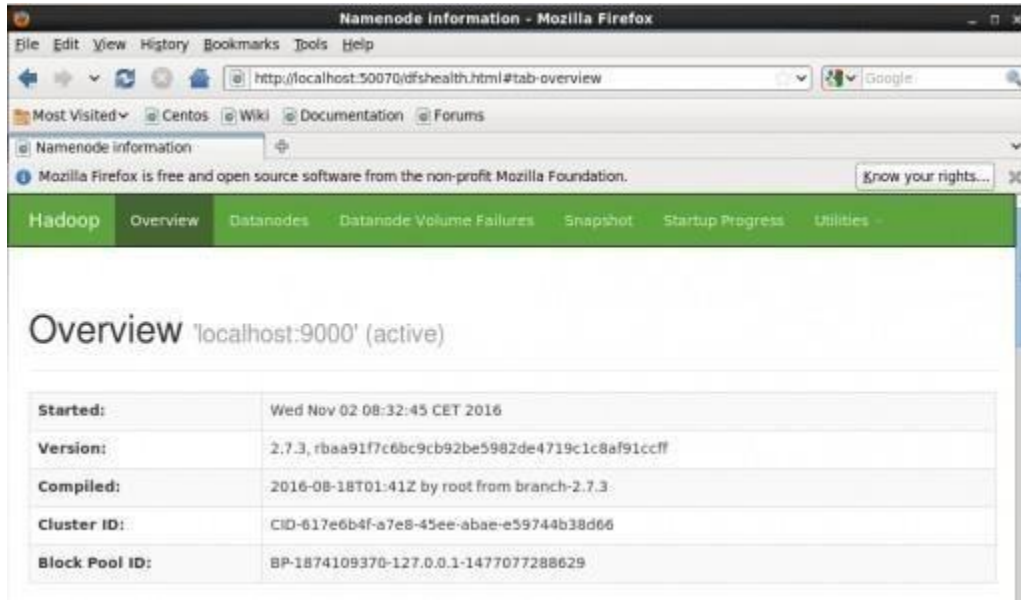


Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster

Result:

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

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.

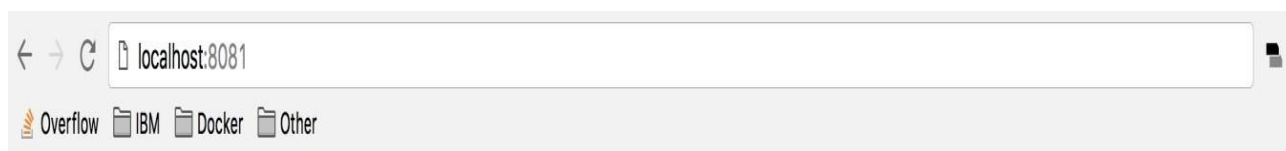
```
$ docker container run --detach --publish 8081:27017 --name mongo mongo:4.4 Unable
to find image mongo:4.4 locally
4.4: Pulling from library/mongo
d13d02fa248d: Already exists
bc8e2652ce92: Pull complete
3cc856886986: Pull complete
c319e9ec4517: Pull complete
b4cbf8808f94: Pull complete
cb98a53e6676: 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.



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

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