V V COLLEGE OF EN ENGINEERING

V V Nagar, Arasoor, Sathankulam (TK)
Tisaiyanvilai (Via), Tuticorin-628 656.
Ph:04637-274300 -329 (30 lines),273412,272467
www.vvcoe.org

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS8711 – CLOUD COMPUTING LABORATORY

Academic Year :2021-22 (ODD)

V V COLLEGE OF EN ENGINEERING

V V Nagar, Arasoor, Sathankulam (TK)
Tisaiyanvilai (Via), Tuticorin-628 656.
Ph:04637-274300 -329(30 lines),273412,272467
www.vvcoe.org

Bonafide Certificate

Certified that this is bonafide record of work done by

Mr./Ms...of

the..........7th..........Semester in............ ***COMPUTER SCIENCE AND***............

*Engineering of this college in the ……......... **CS8711 - CLOUD***

***COMPUTING LABORATORY**............ during............**2021-2022**........in*

Partial fulfilment of the requirements of the B.E Degree Course of the Anna University.

Staff -in-charge **Head of the Department**

University Registration No. ....

University Examination held on ....

Internal Examiner *External Examiner*

INDEX

Exp.No.	Date	Experiment Program	Marks	Staff Sign

EX.NO.1:	**INSTALL VIRTUALBOX WITH LINUX OS ON TOP OF WINDOWS**

AIM:

To install Virtualbox with Ubuntu OS on top of windows host operating systems.

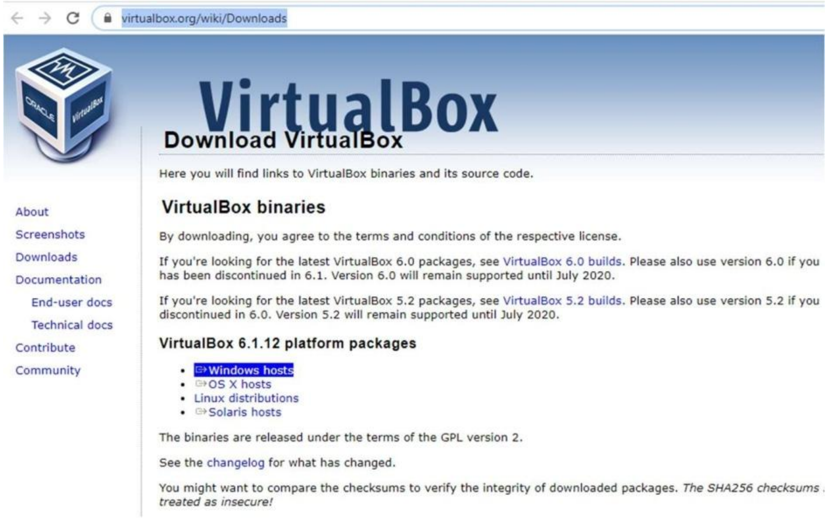
.

PROCEDURE:

Steps to install VirtualBox:

1. Download VirtualBox installer for windows.
2. The installer can be downloaded from the link

   https://www.virtualbox.org/wiki/Downloads



Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see VirtualBox 6.0 builds. Please also use version 6.0 if you has been discontinued in 6.1. Version 6.0 will remain supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see VirtualBox 5.2 builds. Please also use version 5.2 if you discontinued in 6.0. Version 5.2 will remain supported until July 2020.

VirtualBox 6.1.12 platform packages

- Windows hosts
- OS X hosts
- Linux distributions
- Solaris hosts

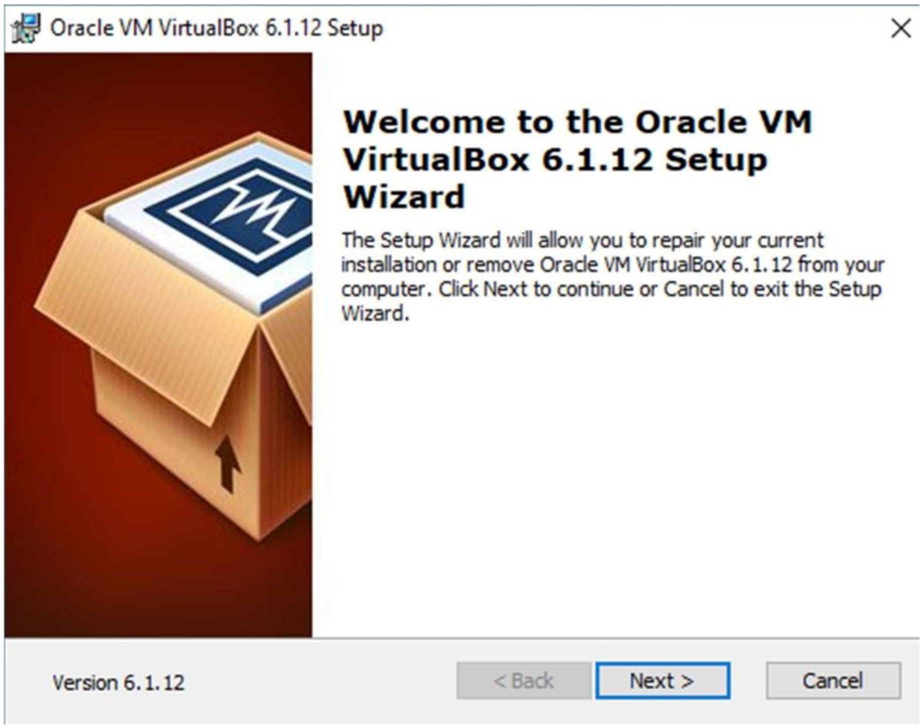
The binaries are released under the terms of the GPL version 2.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums treated as insecure!*

3. Click “Windows host” to download the binary version for windows host.
4. The installer file downloaded will have the file name format like VirtualBox-VersionNumber-BuildNumber-Win.exe.

   Example: VirtualBox-6.1.12-139181-Win.exe.
5. Double click on the installer to launch the setup Wizard. Click on Next to continue.



Oracle VM VirtualBox 6.1.12 Setup

Welcome to the Oracle VM VirtualBox 6.1.12 Setup Wizard

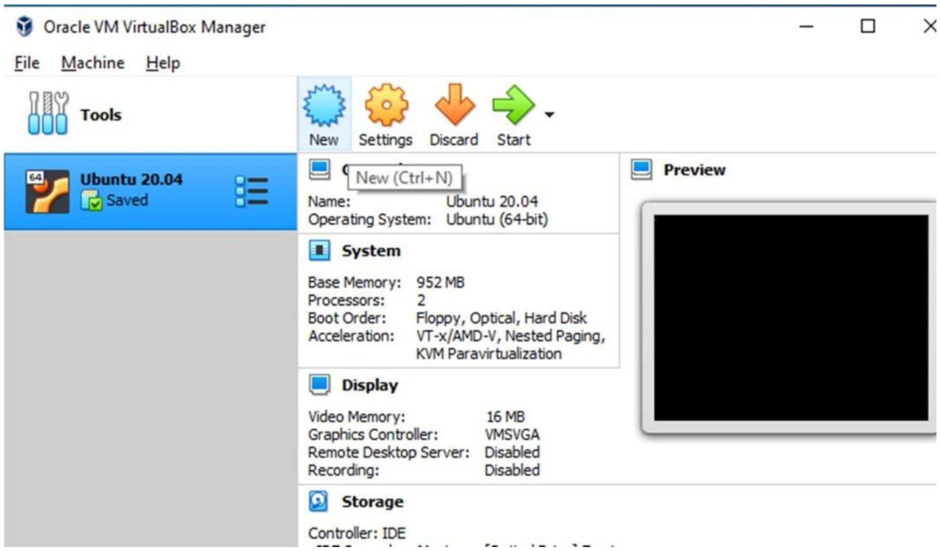
The Setup Wizard will allow you to repair your current installation or remove Oracle VM VirtualBox 6.1.12 from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Version 6.1.12

6. Custom setup dialog box will be opened. Accept the default settings and click next.
7. Select the way you want the features to be installed. You can accept the default and click next.
8. A dialog box opens with Network Interfaces warning. Click Yes to proceed.
9. Click install to begin the installation process.
10. When prompted with a message to install (Trust) Oracle Universal Serial Bus, click Install to continue.
11. After the installation completes, click finish to exit the setup wizard.
12. Launch the Oracle VM VirtualBox.

Steps to create a virtual machine [Ubuntu] in VirtualBox:

1. Open the Oracle VM VirtualBox.
2. Click New icon or ‘Ctrl + N’ to create a new virtual machine.



Name: Ubuntu 20.04
Operating System: Ubuntu (64-bit)

System

Base Memory: 952 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

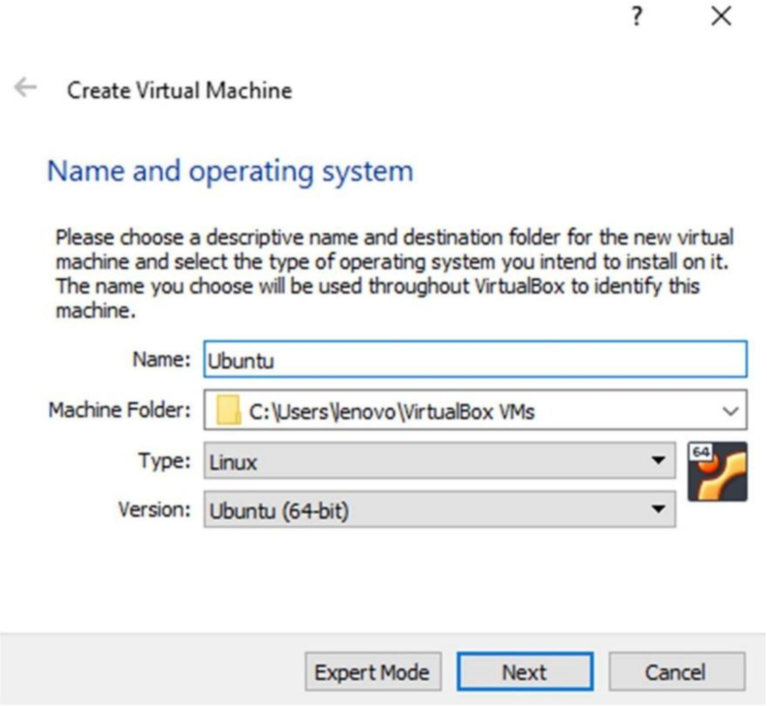
Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE

3. Enter a name for the new virtual machine. Choose the Type and Version. Note that VirtualBox automatically changes 'Type' to Linux and 'Version' to 'Ubuntu (64 bit)' if the name is given as ‘Ubuntu’. Click Next.



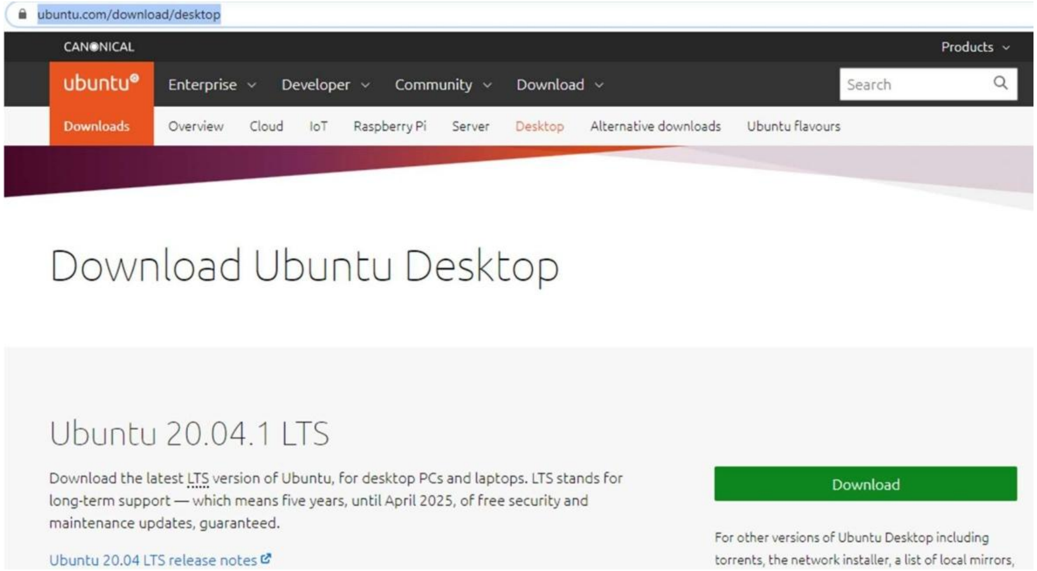
Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Ubuntu
Machine Folder: C:\Users\lenovo\VirtualBox VMs
Type: Linux
Version: Ubuntu (64-bit)

4. Select the amount of RAM to use. The ideal amount of RAM will automatically be selected. Do not increase the RAM into the red section of the slider; keep the slider in the green section.
5. Accept the default 'Create a virtual hard drive now' and click 'Create' button.
6. Choose the hard disk file type as VDI (VirtualBox Disk Image). Click Next.

7. Click Next to accept the default option ‘Dynamically allocated’ for storage on physical hard drive.
8. Select the size of the virtual hard disk and click create.
9. The newly created virtual machine will be displayed in the dashboard.
10. Download the ISO file [Ubuntu disk image file]. Latest version of Ubuntu iso file can be downloaded from the link h ttps://ubuntu.com/download/desktop. Click Download button.



Download Ubuntu Desktop

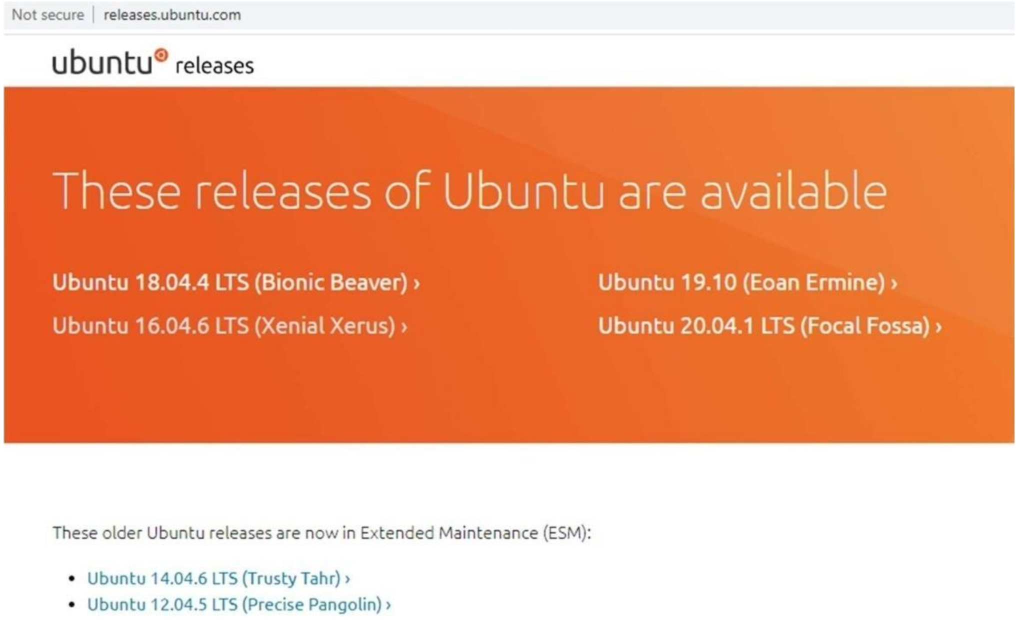
Ubuntu 20.04.1 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

Ubuntu 20.04 LTS release notes

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors,

11. For previous versions, goto http://releases.ubuntu.com. Choose the preferred version of Ubuntu and download the iso file.



These releases of Ubuntu are available

Ubuntu 18.04.4 LTS (Bionic Beaver) ›

Ubuntu 16.04.6 LTS (Xenial Xerus) ›

Ubuntu 19.10 (Eoan Ermine) ›

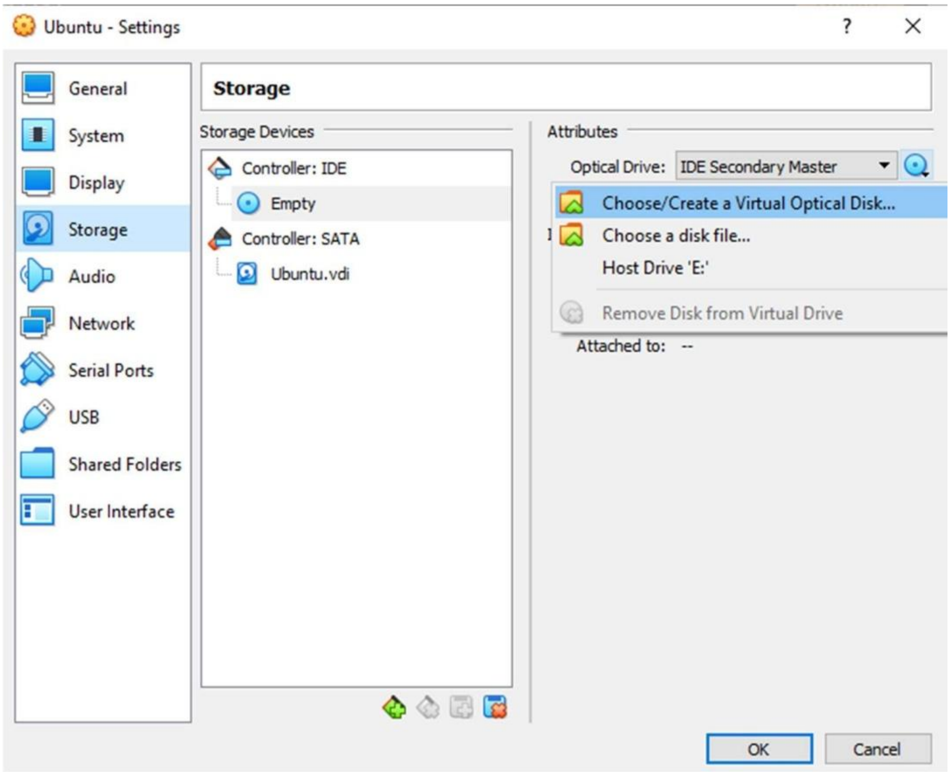
Ubuntu 20.04.1 LTS (Focal Fossa) ›

These older Ubuntu releases are now in Extended Maintenance (ESM):

- Ubuntu 14.04.6 LTS (Trusty Tahr) ›
- Ubuntu 12.04.5 LTS (Precise Pangolin) ›

12. To setup the Ubuntu disk image file (iso file) goto settings.
13. Click Storage. Under ‘Storage Devices’ section click ‘Empty’.

14. In Attributes section, click the disk image and then "Choose Virtual Optical Disk File".



Storage

Storage Devices

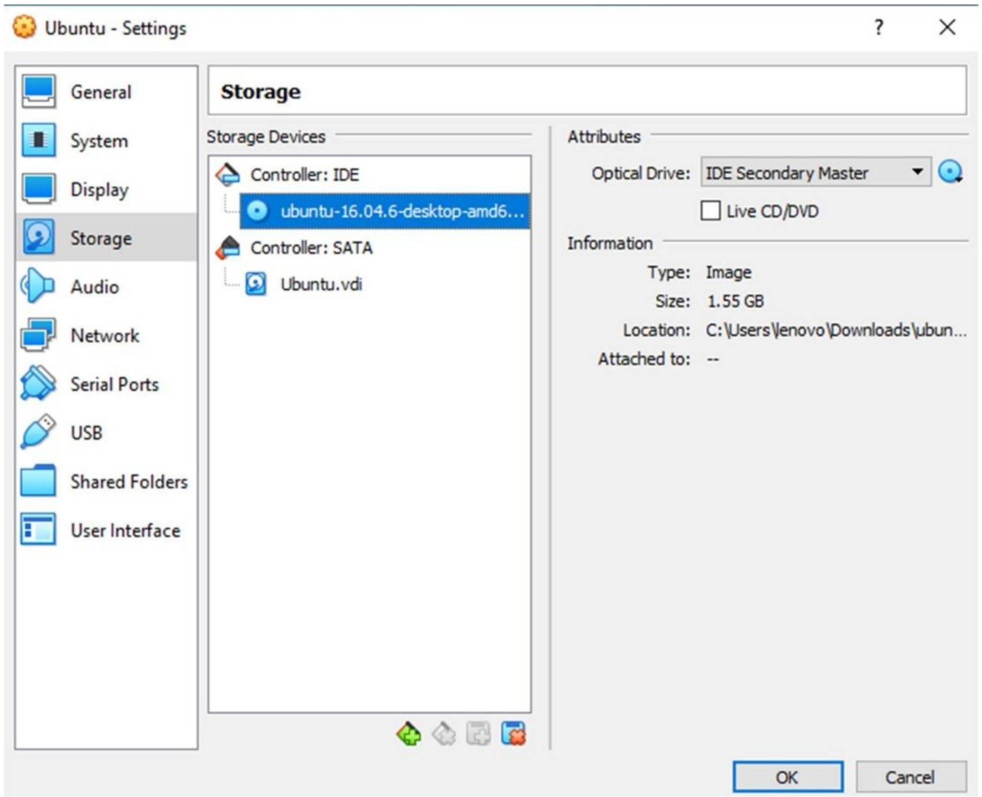
- Controller: IDE
  - Empty
- Controller: SATA
  - Ubuntu.vdi

Attributes

Optical Drive: IDE Secondary Master

Attached to: --

15. Browse and select the downloaded iso file. Click ok.



Storage

Storage Devices

- Controller: IDE
  - ubuntu-16.04.6-desktop-amd6...
- Controller: SATA
  - Ubuntu.vdi

Attributes

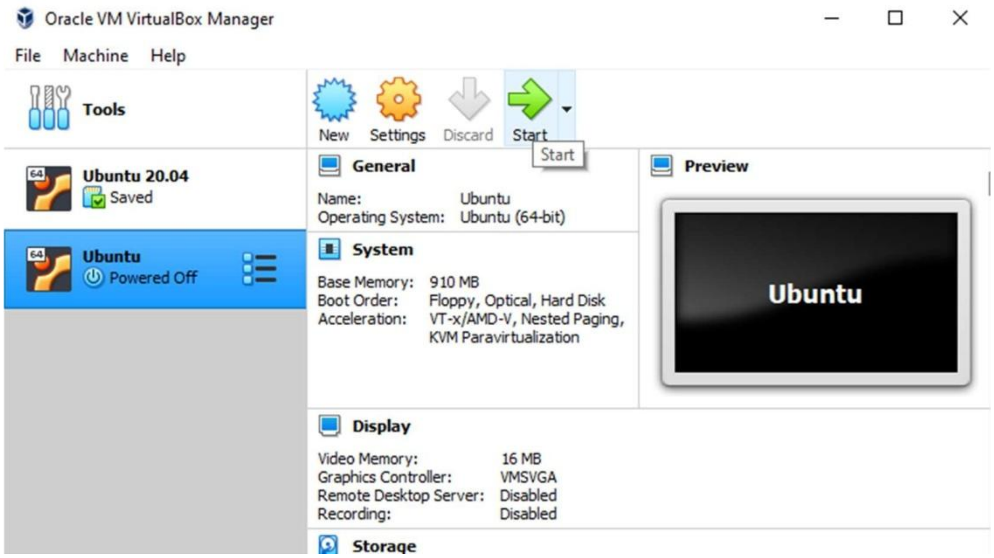
Optical Drive: IDE Secondary Master

□ Live CD/DVD

Information

Type: Image
Size: 1.55 GB
Location: C:\Users\lenovo\Downloads\ubun...
Attached to: --

16. Select the newly created virtual machine in the dashboard and click start button.



General

Name: Ubuntu
Operating System: Ubuntu (64-bit)

System

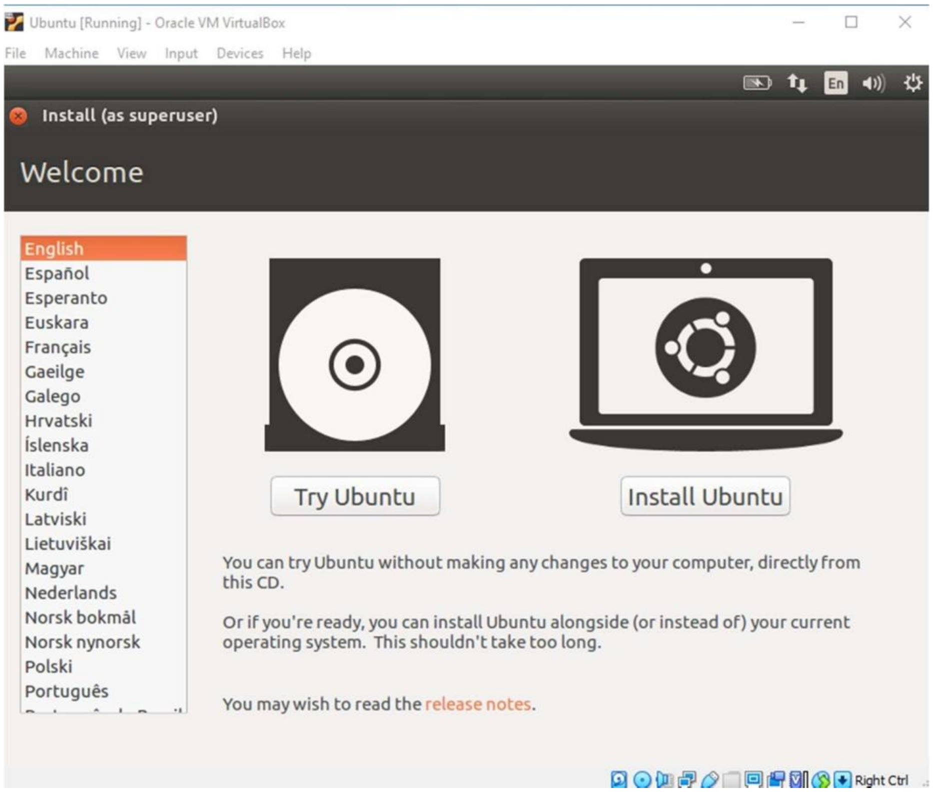
Base Memory: 910 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

17. In the welcome screen, click ‘Install Ubuntu’ button.



You can try Ubuntu without making any changes to your computer, directly from this CD.

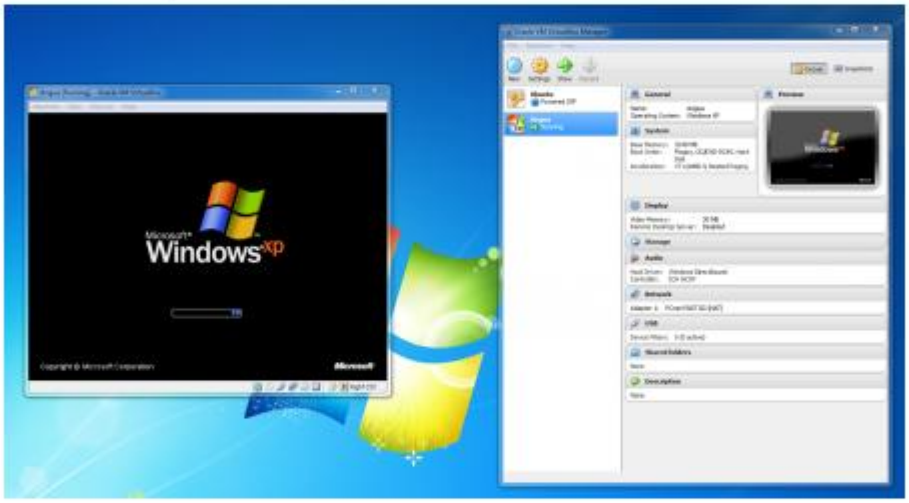
Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

18. Click 'Continue' button.
19. Make sure 'Erase disk and install Ubuntu' option is selected and click 'Install Now' button.
20. Choose the default and click continue.
21. Setup up your profile by creating username and password.
22. After installation is complete, click 'Restart Now' button and follow the instructions.
23. The Ubuntu OS is ready to use. Login with the username and password.

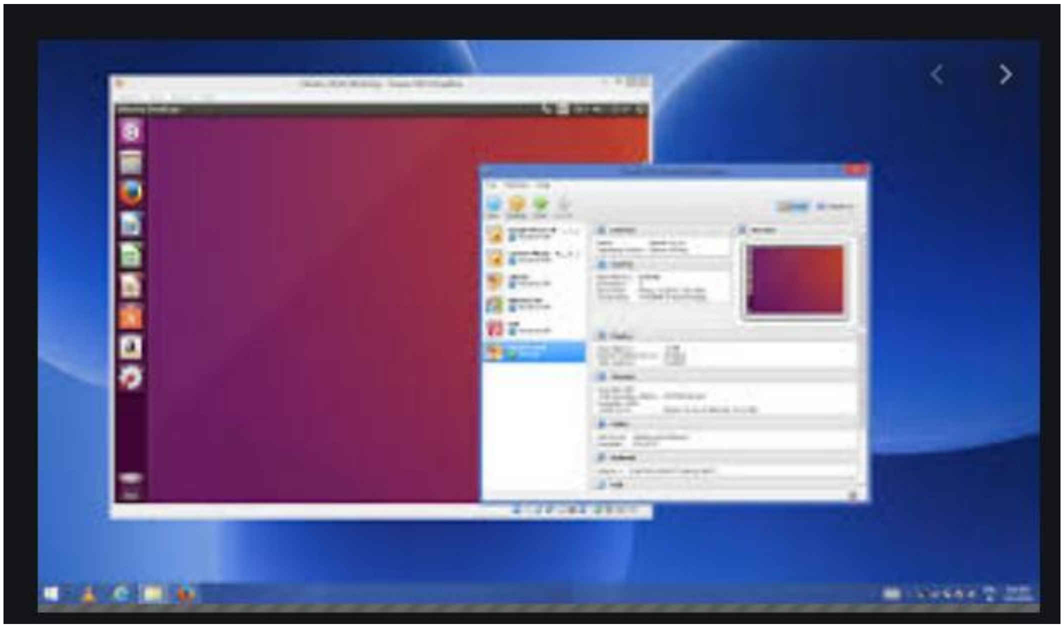
OUTPUT 1:

Virtualbox on top of windows.



OUTPUT 2:

Installation of Virtual box with **Linux OS (Guest OS/VM)**on top of windows Host.



RESULT:

The Virtualbox installation is completed and the Virtual machine is created on top of windows host operating system.

EX.NO:2	**INSTALL A C COMPILER IN THE VIRTUAL MACHINE**

AIM:

To install a C compiler in the virtual machine and execute a sample program.

PROCEDURE:

1. Launch the virtual box and open the virtual machine (Ubuntu).
2. Run the following command in the virtual machine terminal.

```
$ sudo apt update
$ sudo apt install gcc
```

It will install all the necessary packages for gcc complier.

3. Type the C program in the text editor and save the file with with .c extension.

//demo.c

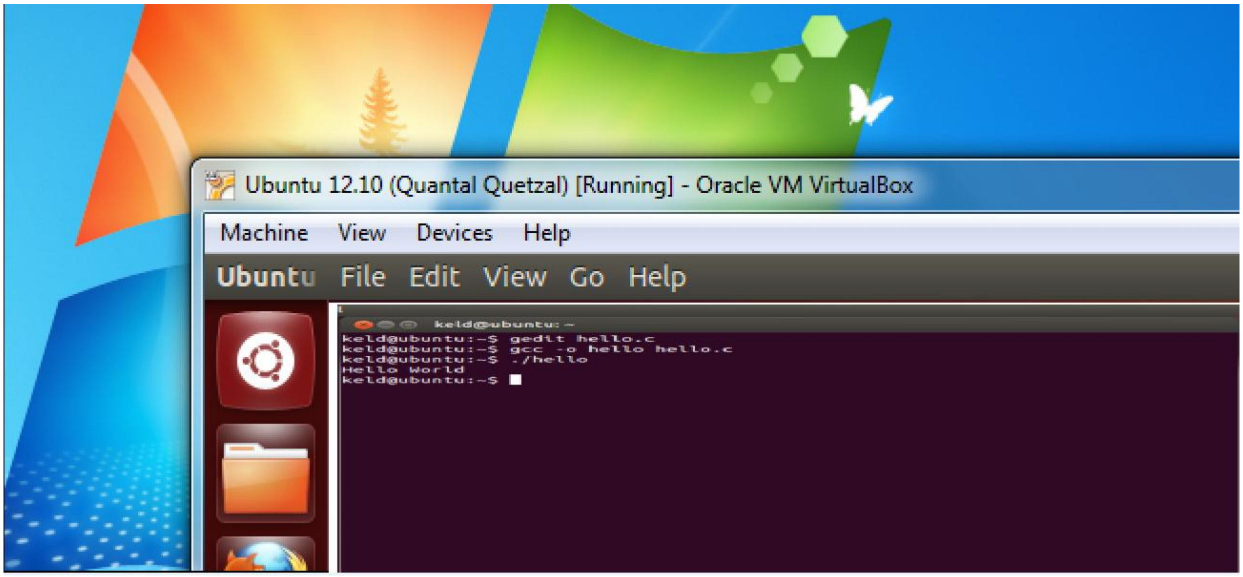
```
#include<stdio.h> int
main()
{
printf("Hello World"); return
0;
}
```

4. Compile and Run the C Program

```
cc demo.c
./a.out
```

OUTPUT:

Installation of a C compiler in the virtual machine and executing a sample program



Ubuntu 12.10 (Quantal Quetzal) [Running] - Oracle VM VirtualBox
Machine
View
Devices
Help
Ubuntu
File
Edit
View
Go
Help
keld@ubuntu: ~

```
keld@ubuntu:~$ gedit hello.c
keld@ubuntu:~$ gcc -o hello hello.c
keld@ubuntu:~$ ./hello
Hello World
keld@ubuntu:~$ 
```

RESULT:

Thus a C compiler is installed on a Ubuntu Virtual Machine on top of Windows Host and executed a C program on a virtual machine.

EX.NO :3	INSTALL GOOGLE APP ENGINE AND CREATE A WEB APPLICATIONS USING JAVA

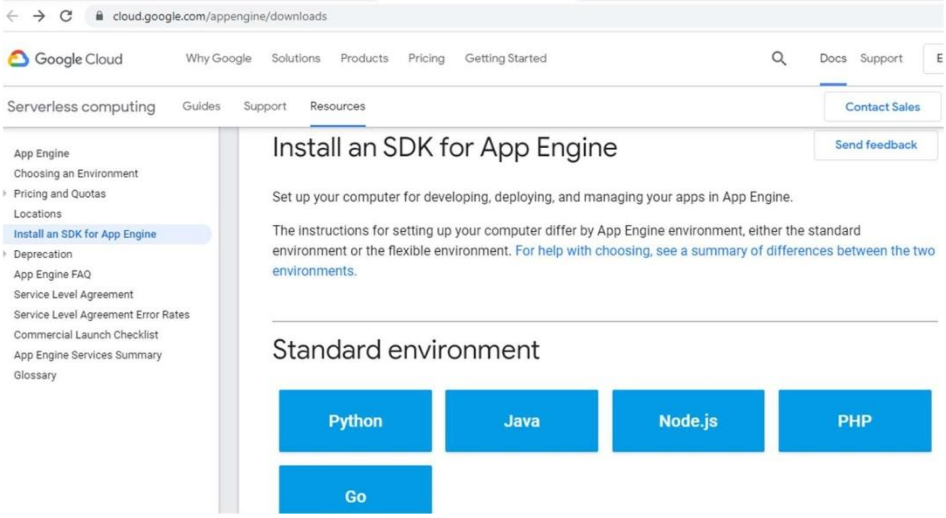
AIM:

To install Google App Engine and to create *hello world* app and other simple web applications using java.

PROCEDURE:

Google App Engine SDK Installation:

1. Download the Google Cloud SDK installer using thelink https://cloud.google.com/appengine/downloads.
2. Select the standard environment asJava.

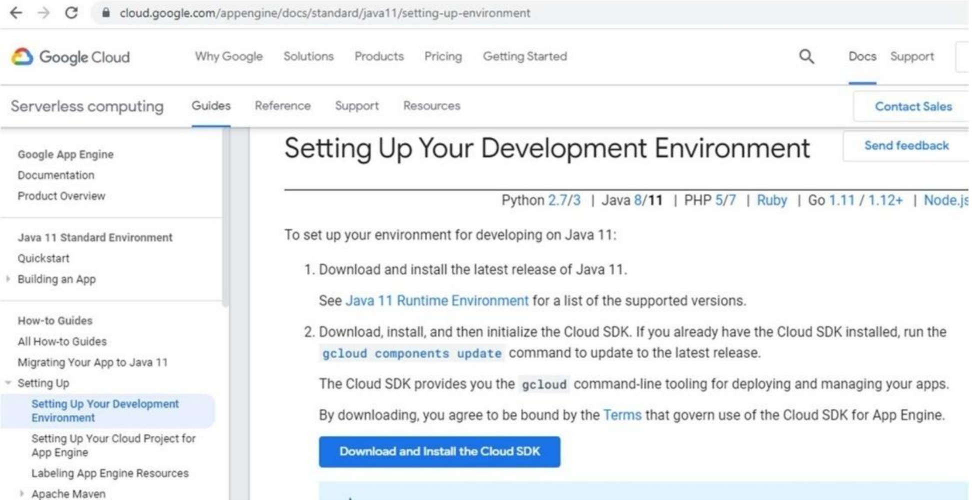


Install an SDK for App Engine

Set up your computer for developing, deploying, and managing your apps in App Engine.

The instructions for setting up your computer differ by App Engine environment, either the standard environment or the flexible environment. For help with choosing, see a summary of differences between the two environments.

Standard environment


cloud.google.com/appengine/docs/standard/java11/setting-up-environment
Google Cloud
Why Google
Solutions
Products
Pricing
Getting Started
Docs
Support
Serverless computing
Guides
Reference
Support
Resources
Contact Sales
Send feedback
Google App Engine
Documentation
Product Overview
Java 11 Standard Environment
Quickstart
Building an App
How-to Guides
All How-to Guides
Migrating Your App to Java 11
Setting Up
Setting Up Your Development Environment
Setting Up Your Cloud Project for App Engine
Labeling App Engine Resources
Apache Maven
Python 2.7/3 | Java 8/11 | PHP 5/7 | Ruby | Go 1.11 / 1.12+ | Node.js
Download and Install the Cloud SDK

Setting Up Your Development Environment

To set up your environment for developing on Java 11:

1. Download and install the latest release of Java 11.

   See Java 11 Runtime Environment for a list of the supported versions.

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

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

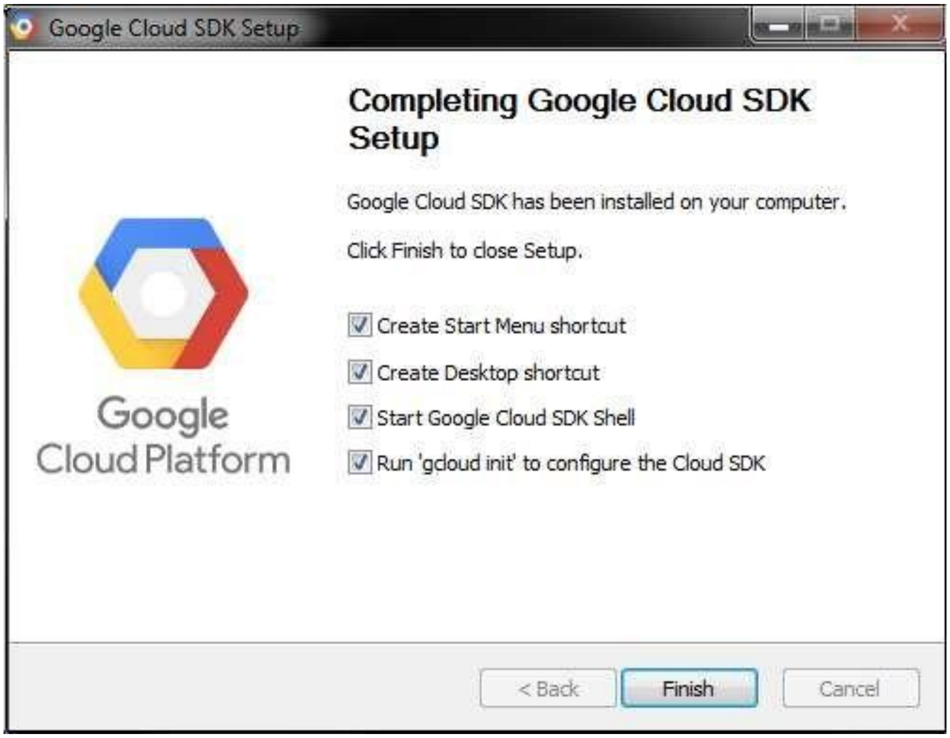
   By downloading, you agree to be bound by the Terms that govern use of the Cloud SDK for App Engine.

3. Click ‘Download and Install the Cloud SDK’. Launch the installer and follow the prompts.
4. After installation has completed, the installer presents severaloptions:

Make sure that the following are selected:

- Start Google Cloud SDKShell
- Run`"gcloudinit"`

Theinstallerthenstartsaterminalwindowandrunsthe`gcloudinit`command.



Google Cloud SDK Setup

Completing Google Cloud SDK Setup

Google Cloud SDK has been installed on your computer.

Click Finish to close Setup.

☑ Create Start Menu shortcut

☑ Create Desktop shortcut

☑ Start Google Cloud SDK Shell

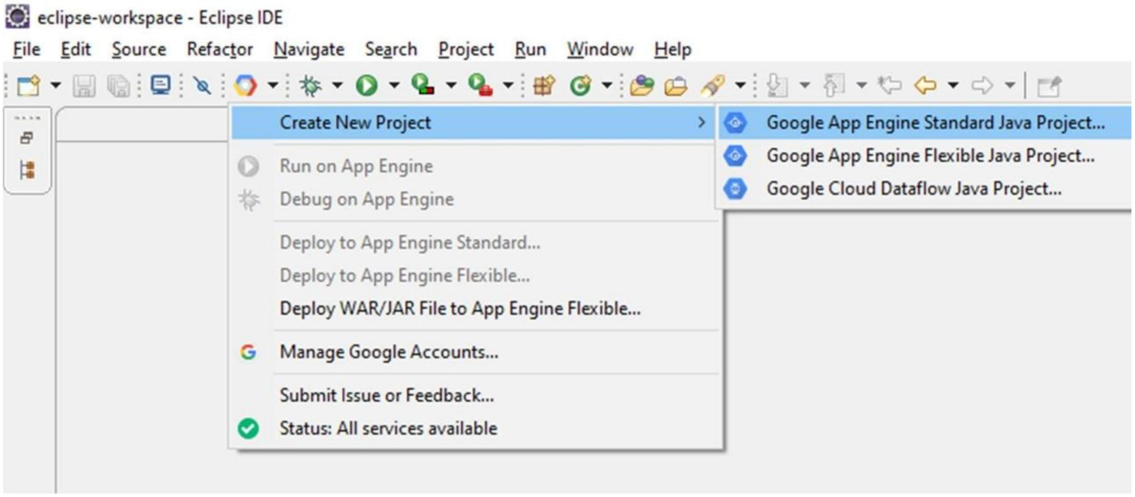
☑ Run 'gcloud init' to configure the Cloud SDK

5. Run the following command in your terminal to install the gcloud component that includes the App Engine extension for Java11:

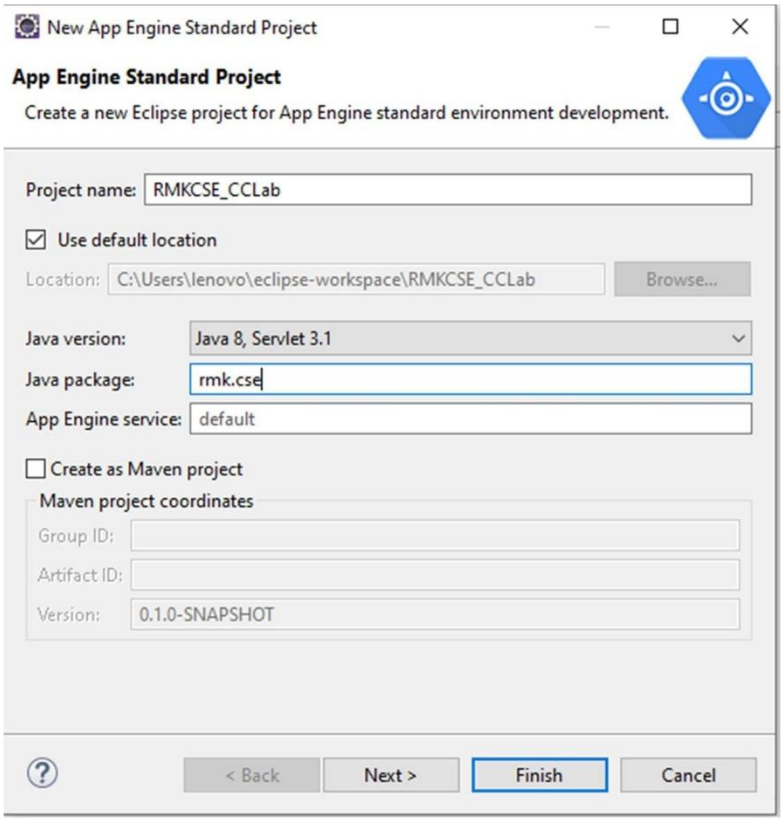
```
gcloudcomponentsinstallapp-engine-java
```

Creating a new App Engine standard project in Eclipse:

6. Eclipse with the cloud tools is used to create App Engine standardproject.
7. To install the Cloud Tools in Eclipse, select Help > Eclipse Marketplace... and search for ‘Google Cloud Tools for Eclipse’ and clickinstall.
8. After installation restart eclipse whenprompted.
9. Click the **Google Cloud Platform** toolbarbutton.
10. Select **Create New Project** >**Google App Engine Standard JavaProject**.


eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Create New Project
Run on App Engine
Debug on App Engine
Deploy to App Engine Standard...
Deploy to App Engine Flexible...
Deploy WAR/JAR File to App Engine Flexible...
Manage Google Accounts...
Submit Issue or Feedback...
Status: All services available
Google App Engine Standard Java Project...
Google App Engine Flexible Java Project...
Google Cloud Dataflow Java Project...

11. Enter the project name and packagename.



New App Engine Standard Project

App Engine Standard Project

Create a new Eclipse project for App Engine standard environment development.

Project name: RMKCSE_CCLab

☑ Use default location

Location: C:\Users\lenovo\eclipse-workspace\RMKCSE_CCLab

Java version: Java 8, Servlet 3.1

Java package: rmk.cse

App Engine service: default

□ Create as Maven project

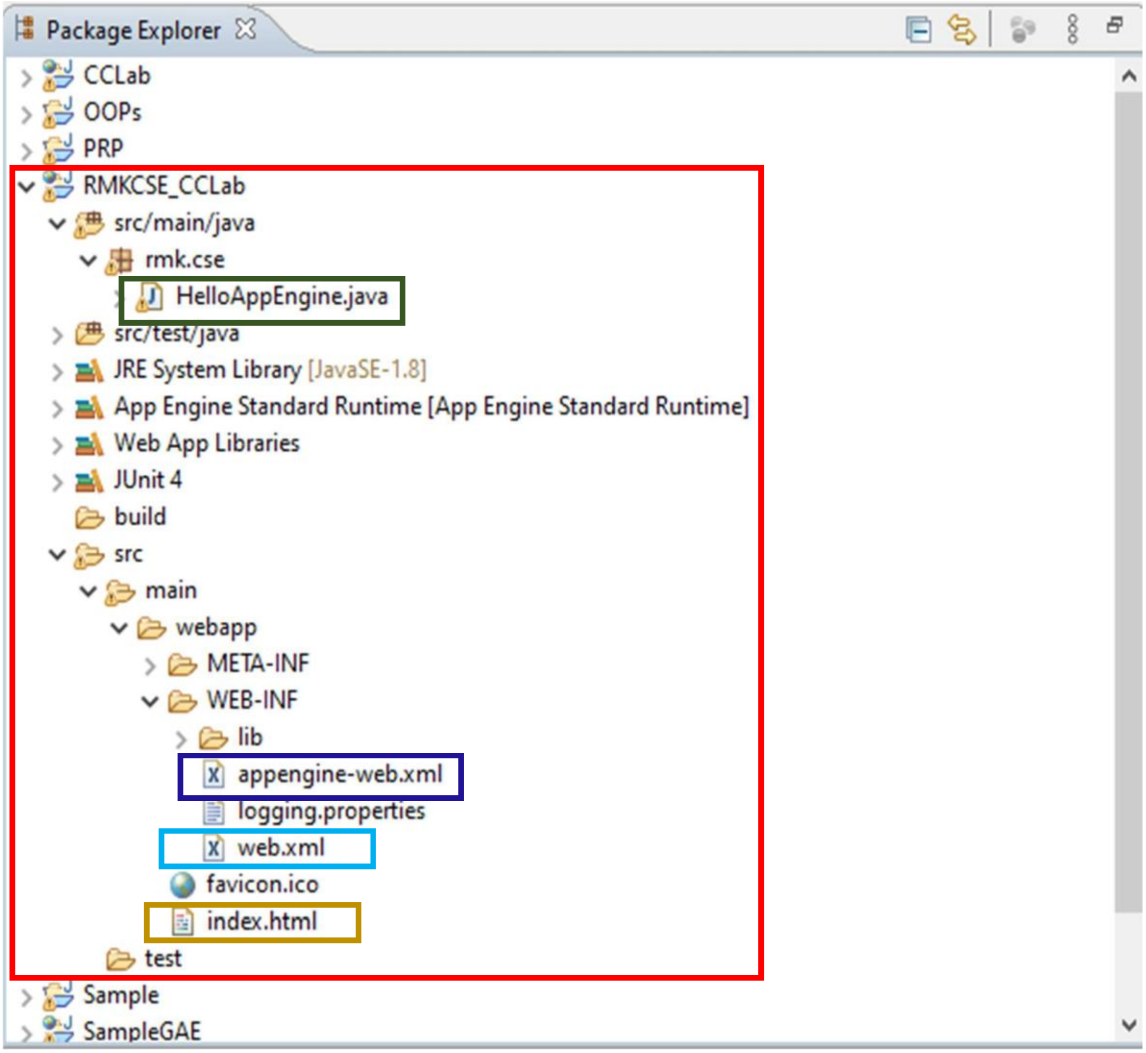
Maven project coordinates

Group ID:

Artifact ID:

Version: 0.1.0-SNAPSHOT

12. Click Next. Select the libraries required for theproject.
13. ClickFinish.
14. The wizard generates a native Eclipse project, with a simple servlet, that you can run and deploy from theIDE.



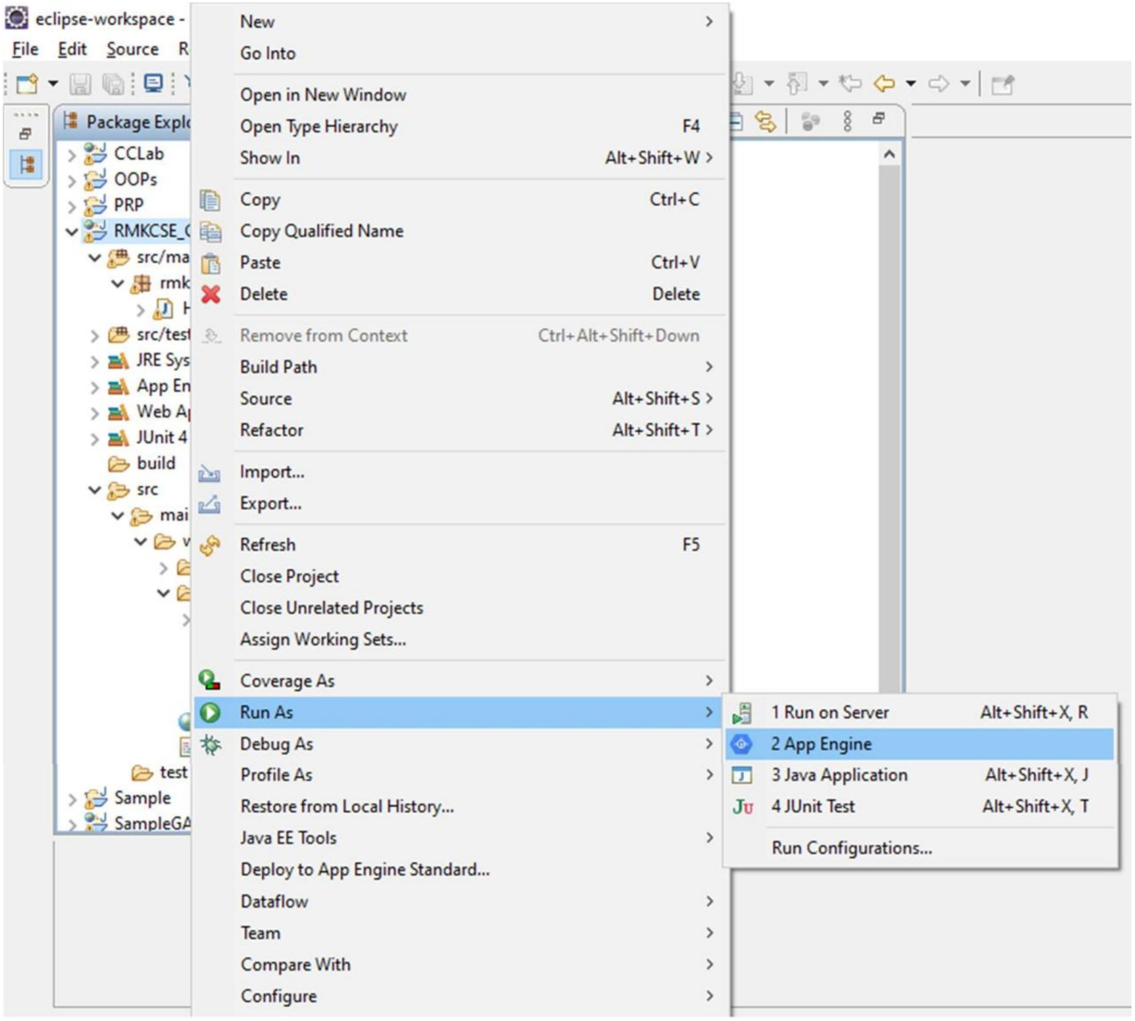
```
Package Explorer
> CCLab
> OOPs
> PRP
v RMKCSE_CCLab
  v src/main/java
    v rmk.cse
      > HelloAppEngine.java
  > src/test/java
  > JRE System Library [JavaSE-1.8]
  > App Engine Standard Runtime [App Engine Standard Runtime]
  > Web App Libraries
  > JUnit 4
    build
  v src
    v main
      v webapp
        > META-INF
        v WEB-INF
          > lib
            appengine-web.xml
            logging.properties
            web.xml
          favicon.ico
          index.html
      test
> Sample
> SampleGAE
```

15. App Engine Java applications use the Java Servlet API to interact with the web server. Modify the defaultHelloAppEngine.java file with your application code.

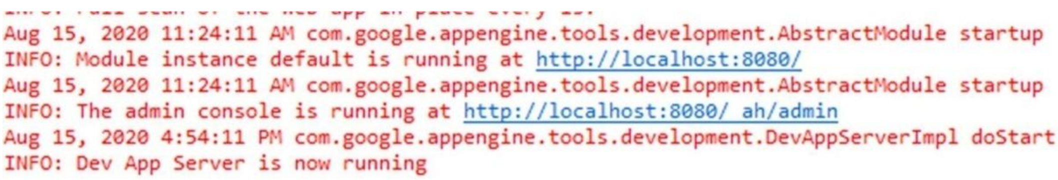
16. **appengine-web.xml**is a Google App Engine specific configurationfile.

17. **web.xml**is a standard web application configurationfile.

18. Right click the project in the Package Explorer, select Run As > AppEngine.



19. Eclipse opens its internal web browser to your application. You can also open an external browser and navigate to http://localhost:8080. Either way, you'll see a static HTML page with a link to theservlet.



```
Aug 15, 2020 11:24:11 AM com.google.appengine.tools.development.AbstractModule startup
INFO: Module instance default is running at http://localhost:8080/
Aug 15, 2020 11:24:11 AM com.google.appengine.tools.development.AbstractModule startup
INFO: The admin console is running at http://localhost:8080/_ah/admin
Aug 15, 2020 4:54:11 PM com.google.appengine.tools.development.DevAppServerImpl doStart
INFO: Dev App Server is now running
```

```
//HelloAppEngine.java

package rmk.cse;

import java.io.IOException;

import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import
javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(
name = "HelloAppEngine", urlPatterns = {"/hello"}
)
public classHelloAppEngine extends HttpServlet {

@Override
public voiddoGet(HttpServletRequest request, HttpServletResponse response)
throws IOException {

response.setContentType("text/plain"); response.setCharacterEncoding("UTF-8");

response.getWriter().print("Hello App Engine!\r\n");

}

}
```

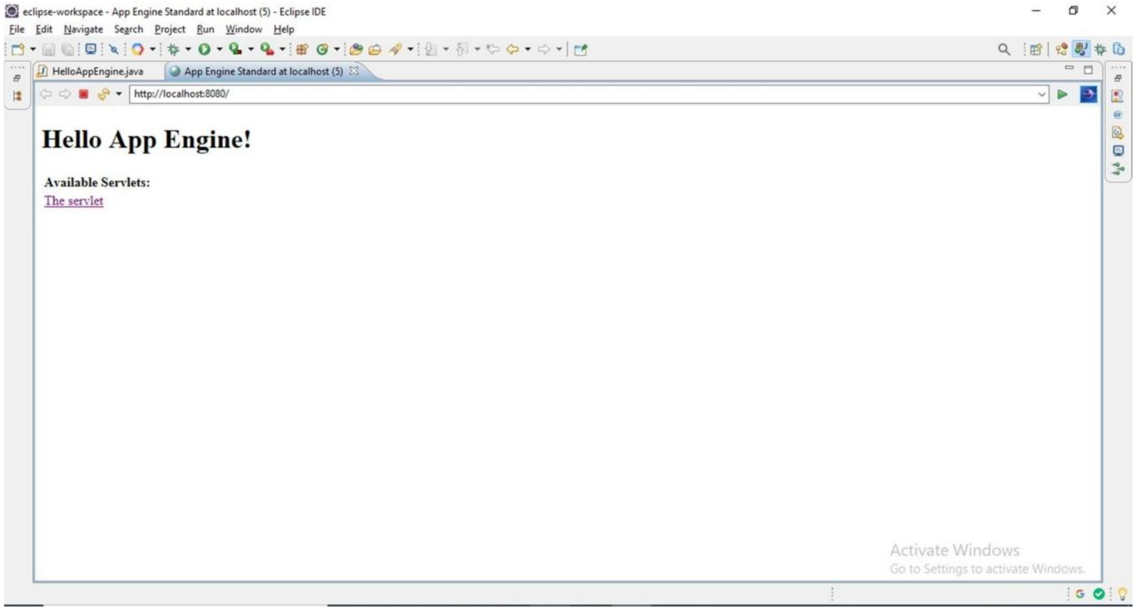
```
//appengine-web.xml

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

<threadsafe>true</threadsafe>
<sessions-enabled>false</sessions-enabled>
<runtime>java8</runtime>

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


</appengine-web-app>
```

```
//web.xml

<?xml version="1.0" encoding="utf-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
version="3.1">
<welcome-file-list>
<welcome-file>index.html</welcome-file>

<welcome-file>index.jsp</welcome-file>
```

```
</welcome-file-list>

</web-app>

//index.html

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
<meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8"
/>
<title>Hello App Engine</title>
</head>

<body>
<h1>Hello App Engine!</h1>

<table>
<tr>
<td colspan="2" style="font-weight:bold;">Available Servlets:</td>
</tr>
<tr>
<td><a href='/hello'>The servlet</a></td>
</tr>
</table>
</body>

</html>
```

OUTPUT 1:

INSTALLATION OF A GOOGLE APP ENGINE


eclipse-workspace - App Engine Standard at localhost (5) - Eclipse IDE
File Edit Navigate Search Project Run Window Help
HelloAppEngine.java
App Engine Standard at localhost (5)
http://localhost:8080/
Activate Windows
Go to Settings to activate Windows.

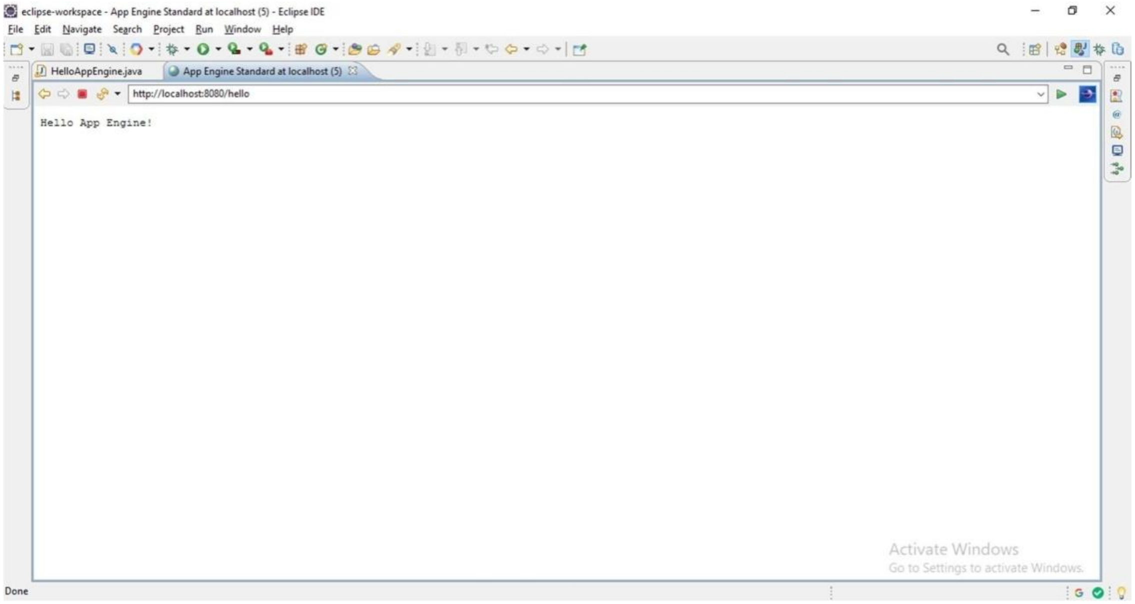
Hello App Engine!

Available Servlets:

The servlet

OUTPUT 2:

CREATION OF A WEB APPLICATIONS ON GAE


eclipse-workspace - App Engine Standard at localhost (5) - Eclipse IDE
File Edit Navigate Search Project Run Window Help
HelloAppEngine.java
App Engine Standard at localhost (5)
http://localhost:8080/hello
Activate Windows
Go to Settings to activate Windows.
Done

```
Hello App Engine!
```

Result:

The Google App Engine is installed and *hello world* app is created in java environment.

Ex.No.4	GOOGLE APP ENGINE PYTHON HELLO WORLD EXAMPLE USING ECLIPSE

AIM:

To launch a web application using **Google App Engine** (GAE).

TOOLS USED :

In this exercise, we are going to create a GAE based Python **web project (hello world)using** Eclipse.

1. Python 2.7
2. Eclipse 3.7 + PyDev plugin
3. Google App Engine SDK for Python 1.6.4

PROCEDURE:

P.S Assume Python 2.7 and Eclipse 3.7 are installed.

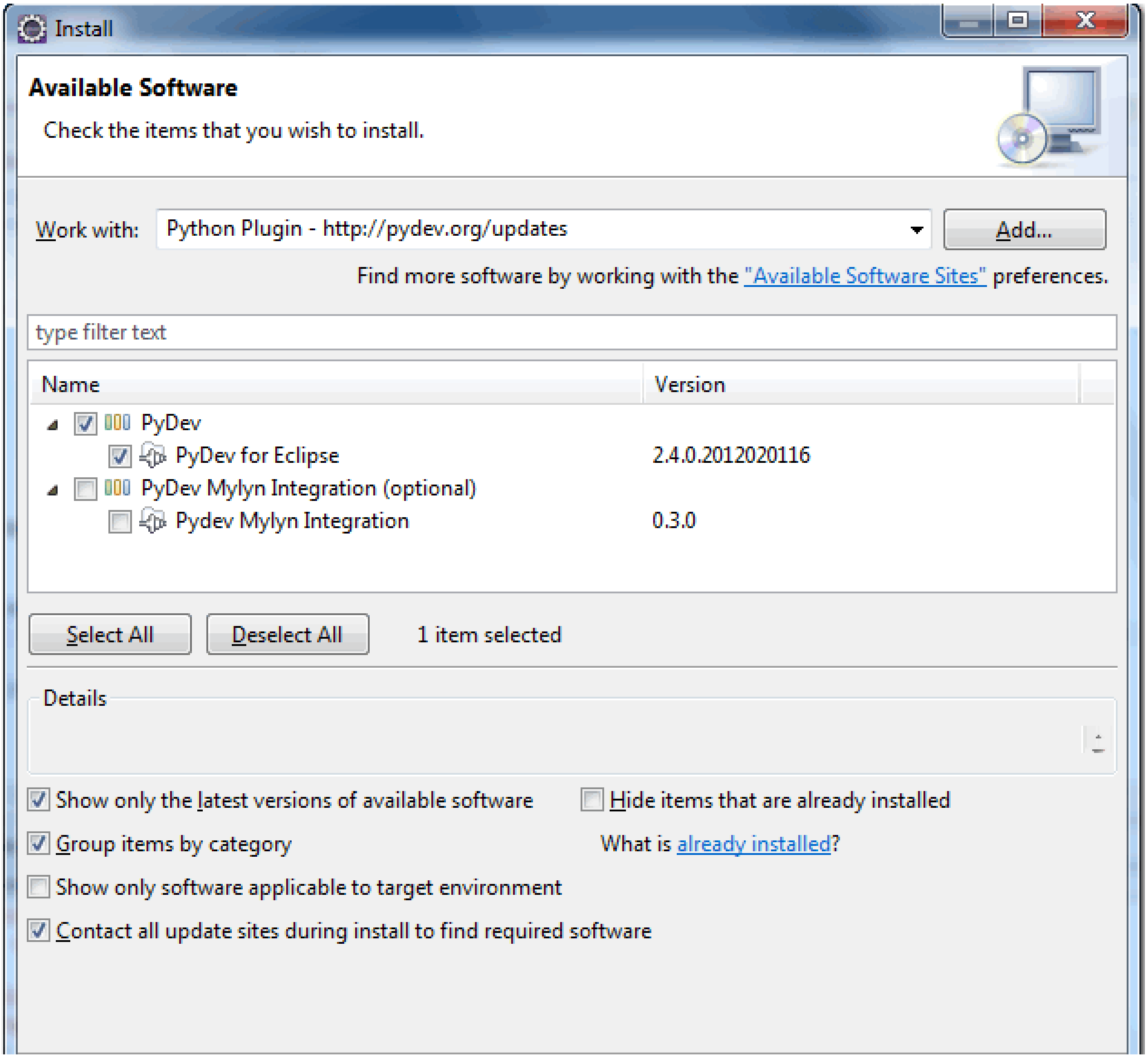
Step:1. Install PyDev plugin for Eclipse

Use following URL to install PyDev as Eclipse plugin.

http://pydev.org/updates

Figure 1 – In Eclipse , menu, “Help –> Install New Software..” and put above URL. Select “**PyDev for**

Eclipse” option, follow steps, and restart Eclipse once completed.



Available Software

Check the items that you wish to install.

Work with: Python Plugin - http://pydev.org/updates

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ PyDev	
☑ PyDev for Eclipse	2.4.0.2012020116
□ PyDev Mylyn Integration (optional)	
□ Pydev Mylyn Integration	0.3.0

1 item selected

Details

☑ Show only the latest versions of available software

□ Hide items that are already installed

☑ Group items by category

What is already installed?

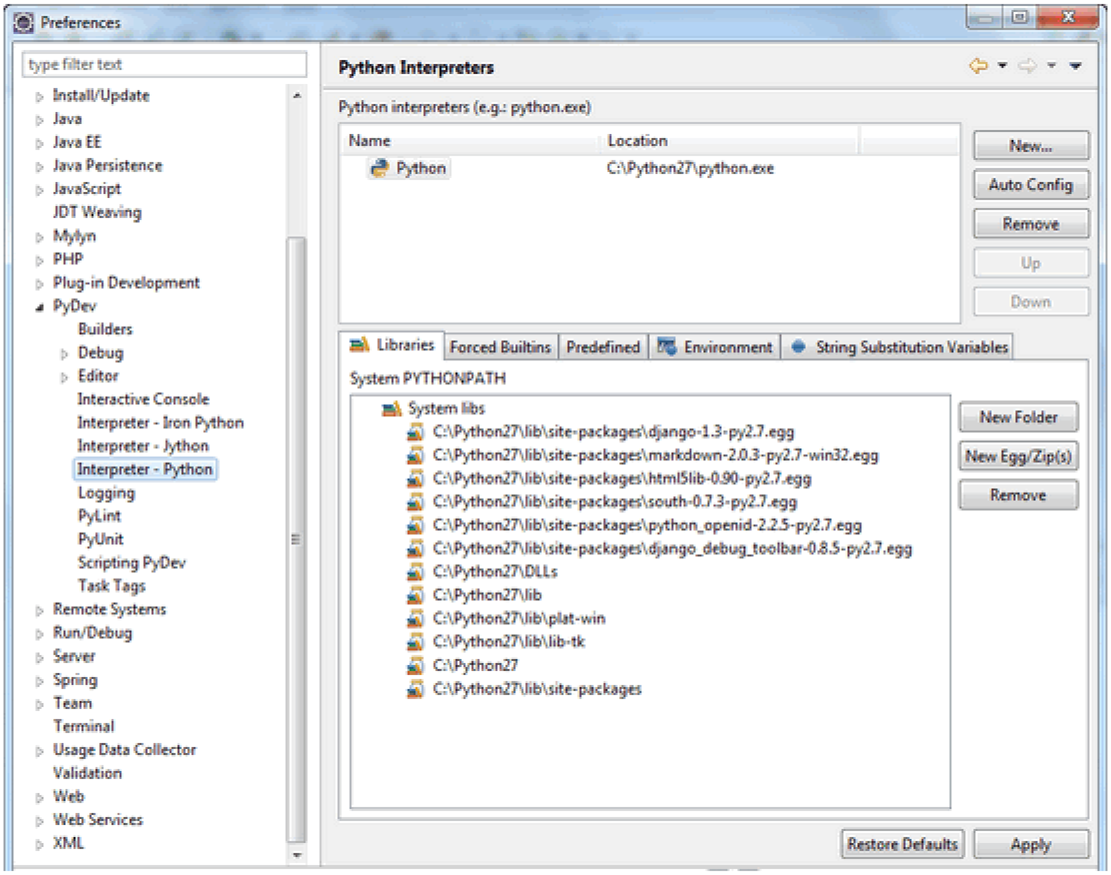
□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

Step 2. Verify PyDev

After Eclipse is restarted, make sure **PyDev's interpreter** is pointed to your "`python.exe`".

Figure 2 – Eclipse -> Windows –> Preferences, make sure "**Interpreter – Python**" is configured properly.



Python interpreters (e.g.: python.exe)

Name	Location
Python	C:\Python27\python.exe

System PYTHONPATH

```
System libs
    C:\Python27\lib\site-packages\django-1.3-py2.7.egg
    C:\Python27\lib\site-packages\markdown-2.0.3-py2.7-win32.egg
    C:\Python27\lib\site-packages\html5lib-0.90-py2.7.egg
    C:\Python27\lib\site-packages\south-0.7.3-py2.7.egg
    C:\Python27\lib\site-packages\python_openid-2.2.5-py2.7.egg
    C:\Python27\lib\site-packages\django_debug_toolbar-0.8.5-py2.7.egg
    C:\Python27\DLLs
    C:\Python27\lib
    C:\Python27\lib\plat-win
    C:\Python27\lib\lib-tk
    C:\Python27
    C:\Python27\lib\site-packages
```

Step:3. Google App Engine SDK Python

Download and install Google App Engine SDK for Python.

Step:4. Python Hello World in Eclipse

Following steps to show you how to create a GAE project via Pydev plugin.

Figure 4.1 – Eclipse menu, File -> New -> Other… , PyDev folder, choose "**PyDev Google App Engine**

Project".

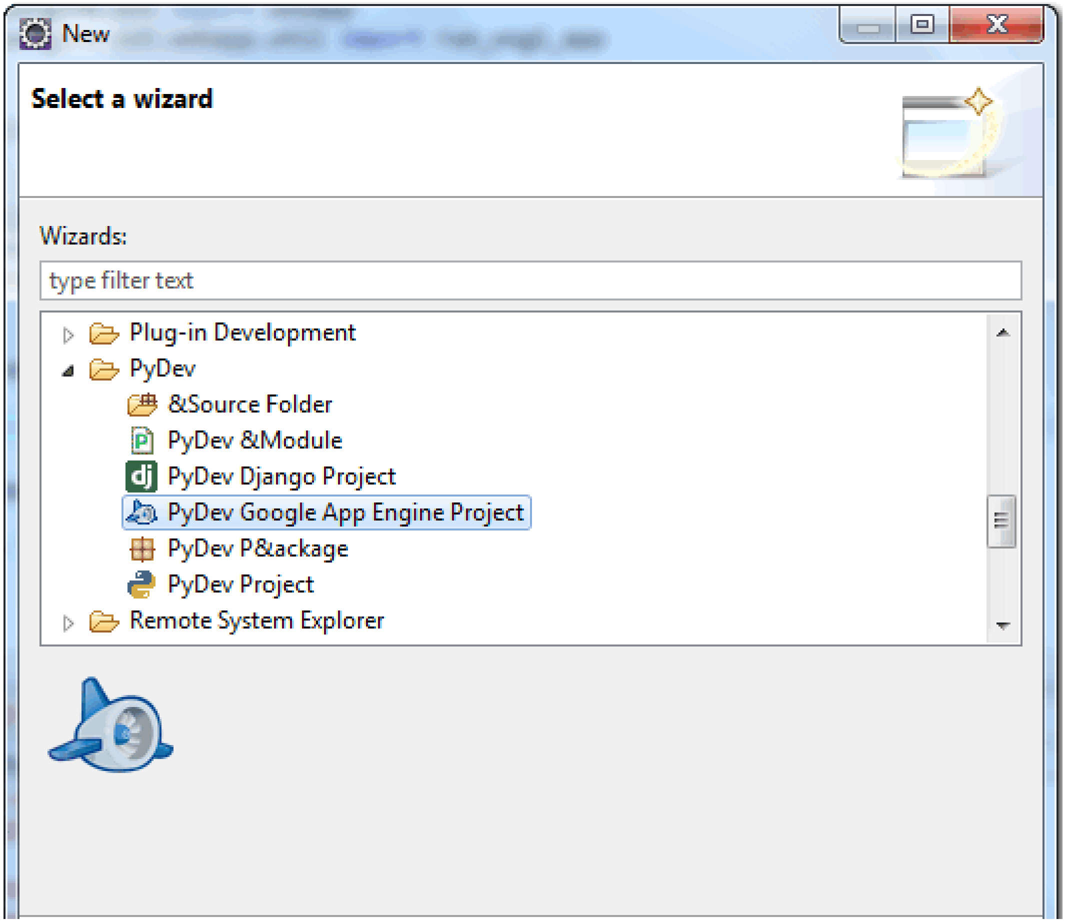
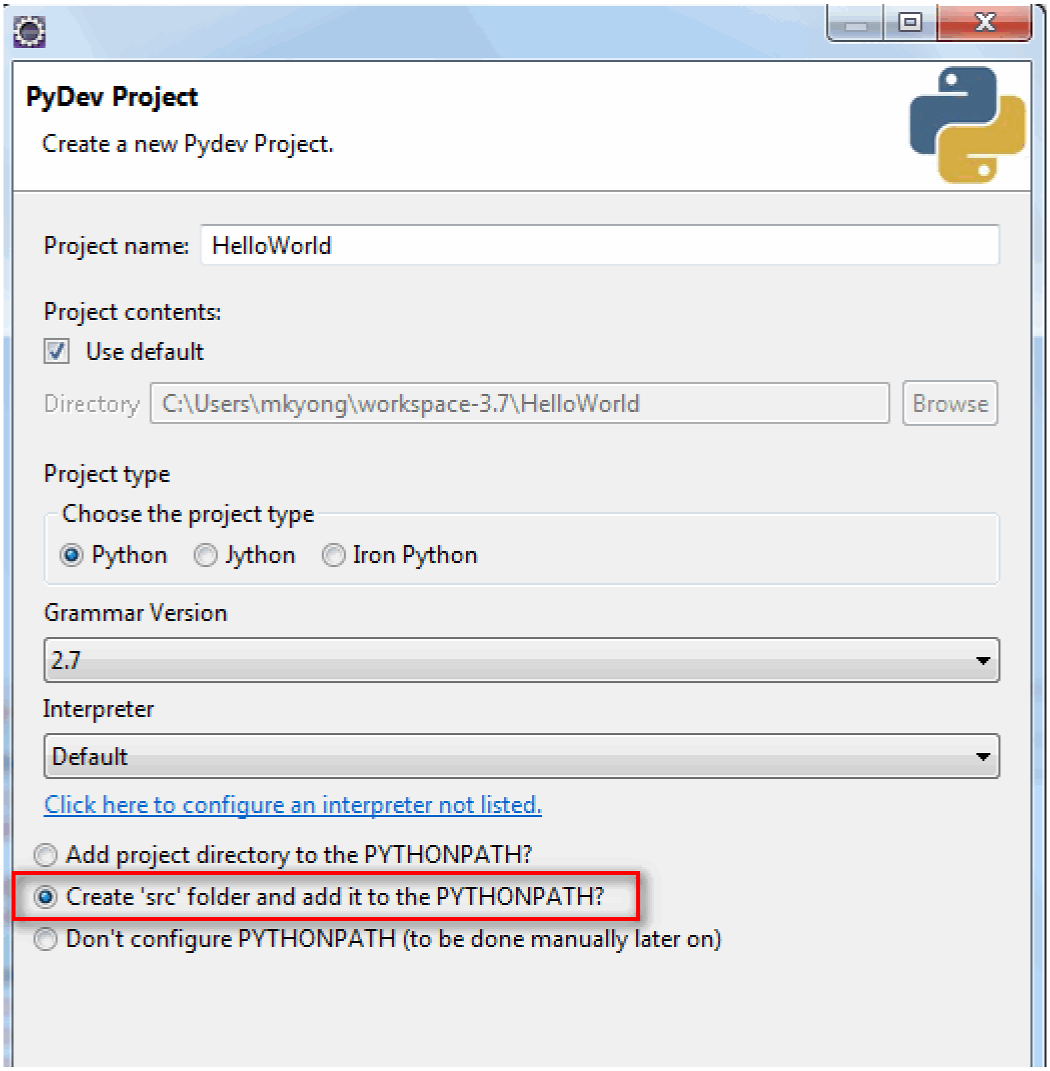


Figure 4.2 – Type project name, if the interpreter is not configure yet (in step 2), you can do it now. And select

this option – **“Create ‘src’ folder and add it to PYTHONPATH**“.



PyDev Project

Create a new Pydev Project.

Project name: HelloWorld

Project contents:

☑ Use default

Directory C:\Users\mkyong\workspace-3.7\HelloWorld

Project type

Choose the project type

● Python ○ Jython ○ Iron Python

Grammar Version

2.7

Interpreter

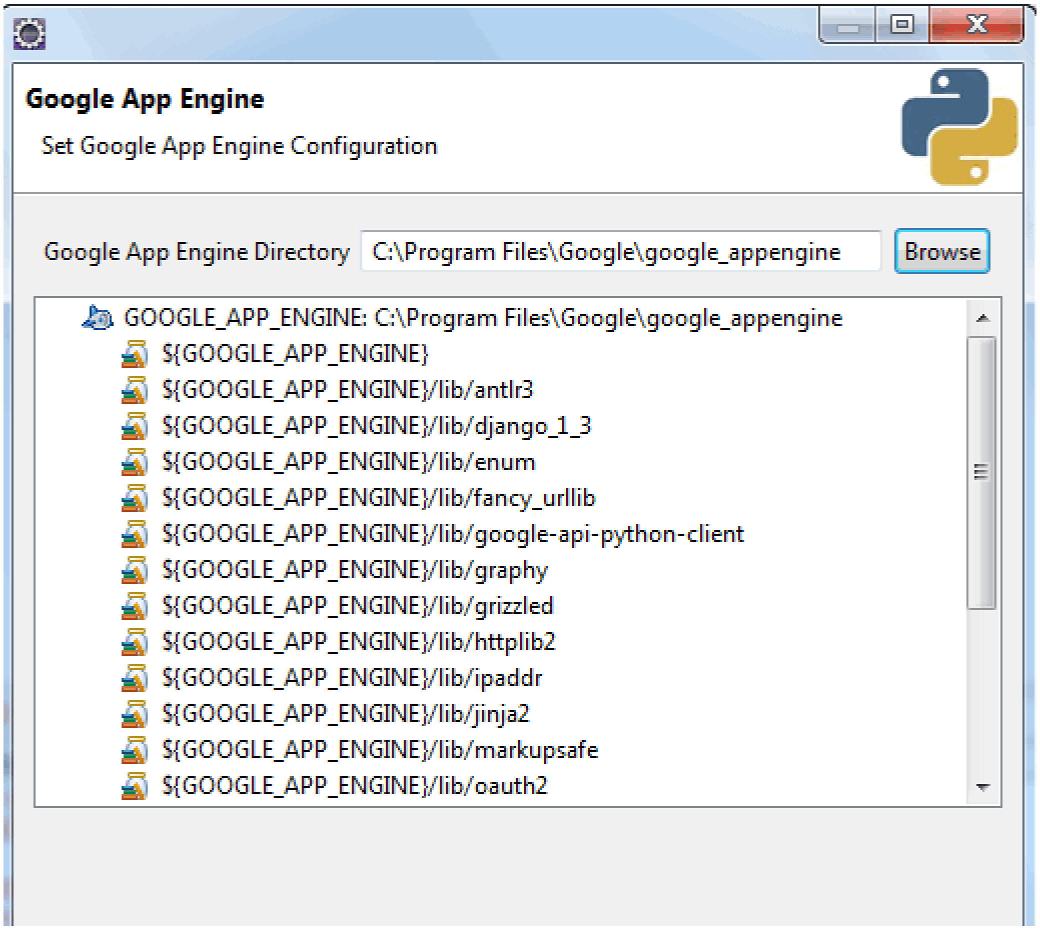
Default

○ Add project directory to the PYTHONPATH?

● Create 'src' folder and add it to the PYTHONPATH?

○ Don't configure PYTHONPATH (to be done manually later on)

Figure 4.3 – Click "Browse" button and point it to the Google App Engine installed directory (in step 3).



Google App Engine

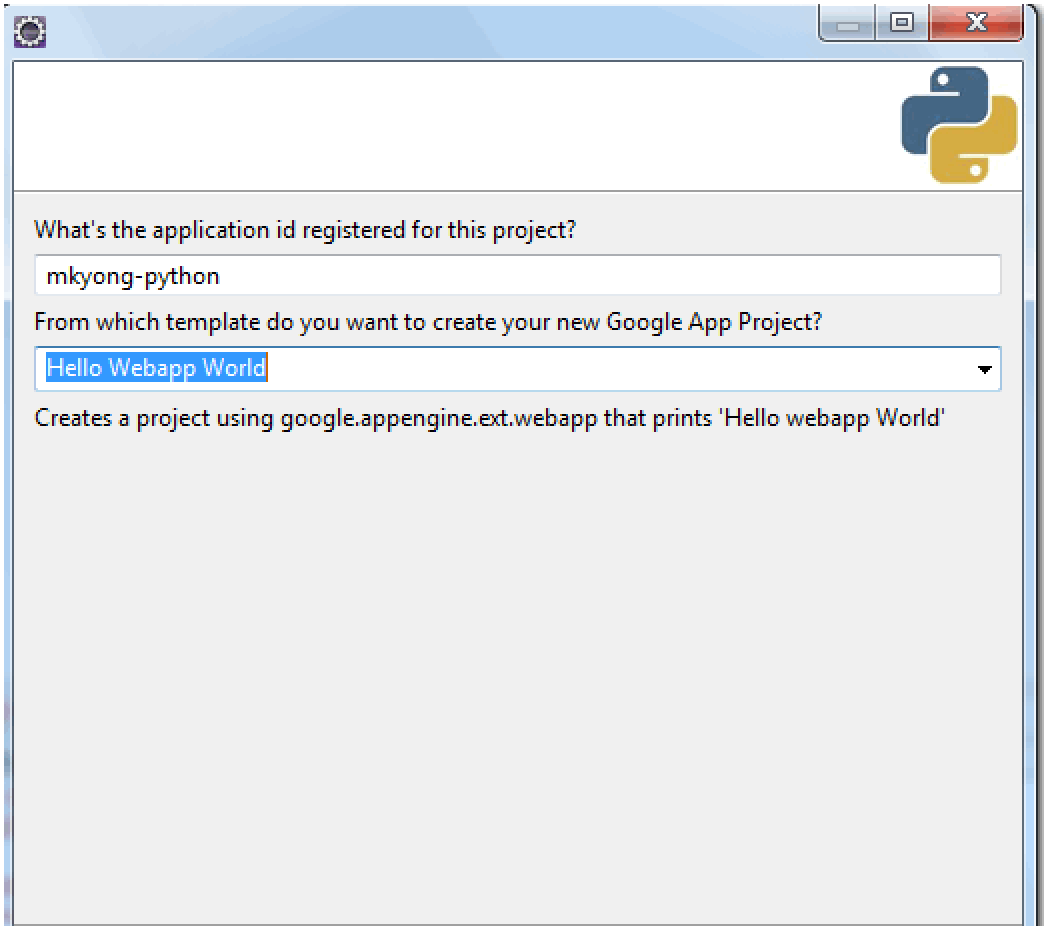
Set Google App Engine Configuration

Google App Engine Directory: C:\Program Files\Google\google_appengine

```
GOOGLE_APP_ENGINE: C:\Program Files\Google\google_appengine
    ${GOOGLE_APP_ENGINE}
    ${GOOGLE_APP_ENGINE}/lib/antlr3
    ${GOOGLE_APP_ENGINE}/lib/django_1_3
    ${GOOGLE_APP_ENGINE}/lib/enum
    ${GOOGLE_APP_ENGINE}/lib/fancy_urllib
    ${GOOGLE_APP_ENGINE}/lib/google-api-python-client
    ${GOOGLE_APP_ENGINE}/lib/graphy
    ${GOOGLE_APP_ENGINE}/lib/grizzled
    ${GOOGLE_APP_ENGINE}/lib/httplib2
    ${GOOGLE_APP_ENGINE}/lib/ipaddr
    ${GOOGLE_APP_ENGINE}/lib/jinja2
    ${GOOGLE_APP_ENGINE}/lib/markupsafe
    ${GOOGLE_APP_ENGINE}/lib/oauth2
```

Figure 4.4 – Name your application id in GAE, type anything, you can change it later. And choose “**Hello**

Webapp World” template to generate the sample files.



What's the application id registered for this project?

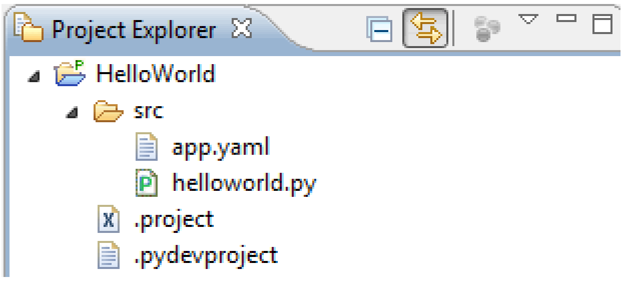
mkyong-python

From which template do you want to create your new Google App Project?

Hello Webapp World

Creates a project using google.appengine.ext.webapp that prints 'Hello webapp World'

Figure 4.5 – Done, 4 files are generated, Both "`.pydevproject`" and "`.project`" are Eclipse project files, ignore it.



```
HelloWorld
    src
        app.yaml
        helloworld.py
    .project
    .pydevproject
```

Review the generated Python's files :

File : helloworld.py – Just output a hello world.

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app

class MainPage(webapp.RequestHandler):

    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write('Hello, webapp World!')

application = webapp.WSGIApplication([('/', MainPage)], debug=True)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

Copy

File : app.yaml – GAE need this file to run and deploy your Python project, it's quite self-explanatory, for

detail syntax and configuration, visit yaml and app.yaml reference.

```
application: mkyong-python
version: 1
runtime: python
api_version: 1

handlers:
- url: /.*
  script: helloworld.py
```

Copy

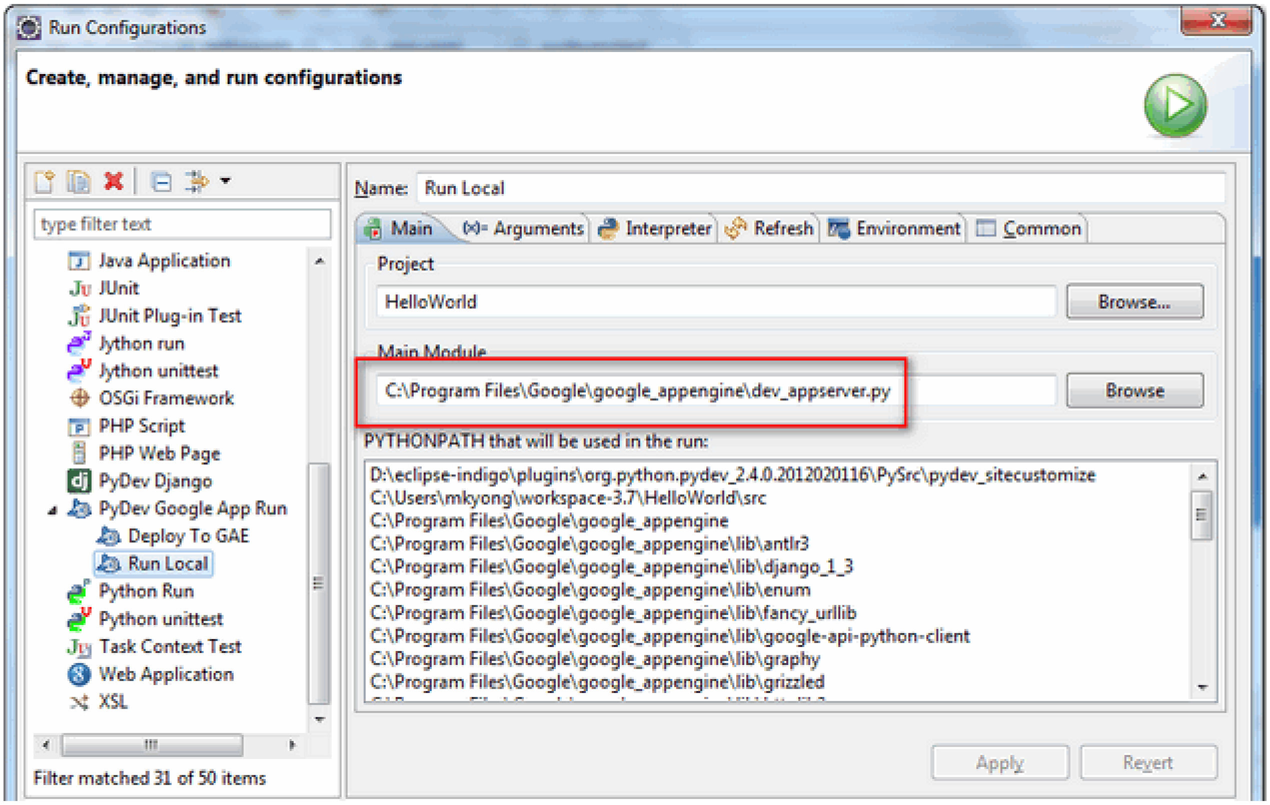
Step:5. Run it locally

To run it locally, right click on the `helloworld.py`, choose "Run As" –> "Run Configuration", create a new

"**PyDev Google App Run**".

Figure 5.1 – In Main tab -> Main module, manually type the directory path of "**dev_appserver.py**".

"Browse" button is not able to help you, type manually.



Name: Run Local

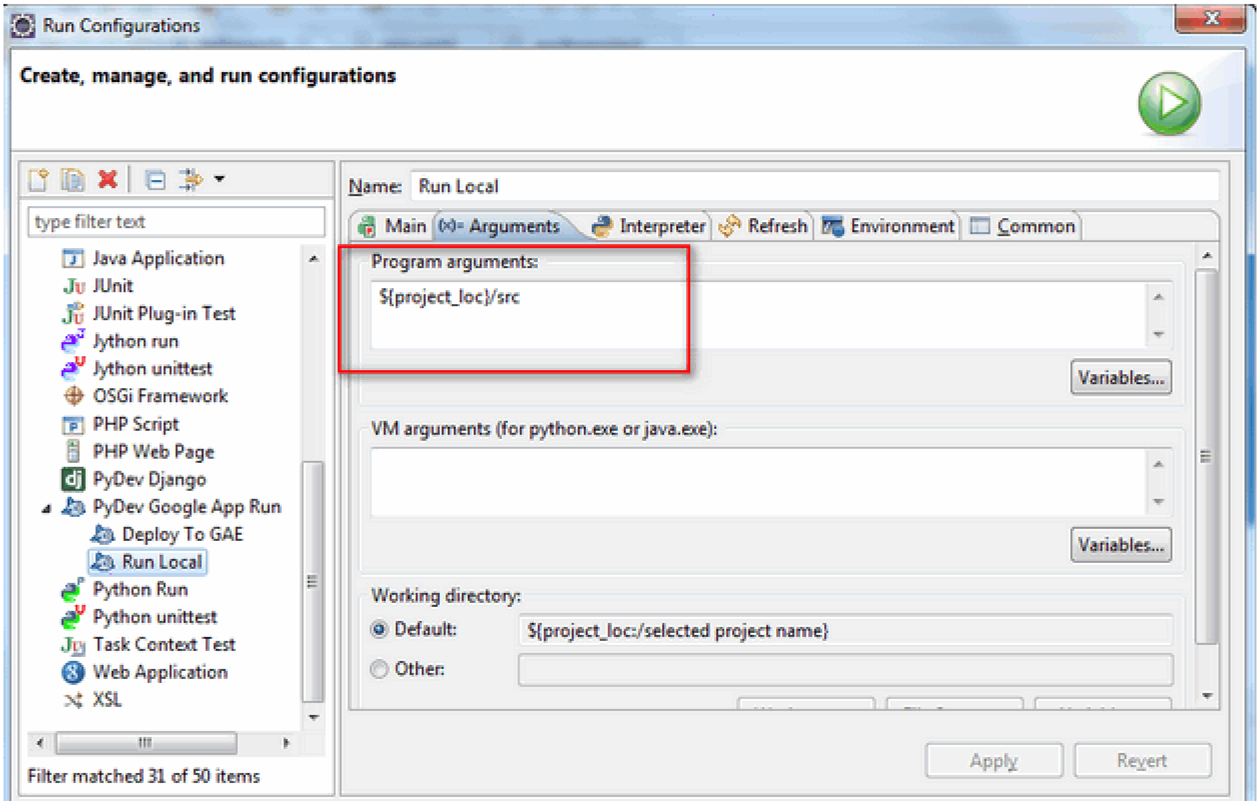
Project: HelloWorld

Main Module: C:\Program Files\Google\google_appengine\dev_appserver.py

PYTHONPATH that will be used in the run:

```
D:\eclipse-indigo\plugins\org.python.pydev_2.4.0.2012020116\PySrc\pydev_sitecustomize
C:\Users\mkyong\workspace-3.7\HelloWorld\src
C:\Program Files\Google\google_appengine
C:\Program Files\Google\google_appengine\lib\antlr3
C:\Program Files\Google\google_appengine\lib\django_1_3
C:\Program Files\Google\google_appengine\lib\enum
C:\Program Files\Google\google_appengine\lib\fancy_urllib
C:\Program Files\Google\google_appengine\lib\google-api-python-client
C:\Program Files\Google\google_appengine\lib\graphy
C:\Program Files\Google\google_appengine\lib\grizzled
```

Figure 5.2 – In Arguments tab -> Program arguments, put "**${project_loc}/src**".



Name: Run Local

Program arguments: ${project_loc}/src

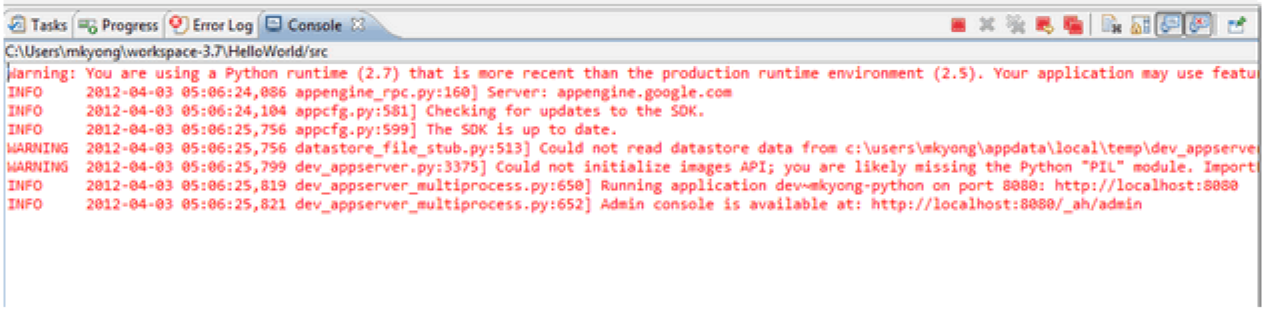
VM arguments (for python.exe or java.exe):

Working directory:

● Default: ${project_loc:/selected project name}

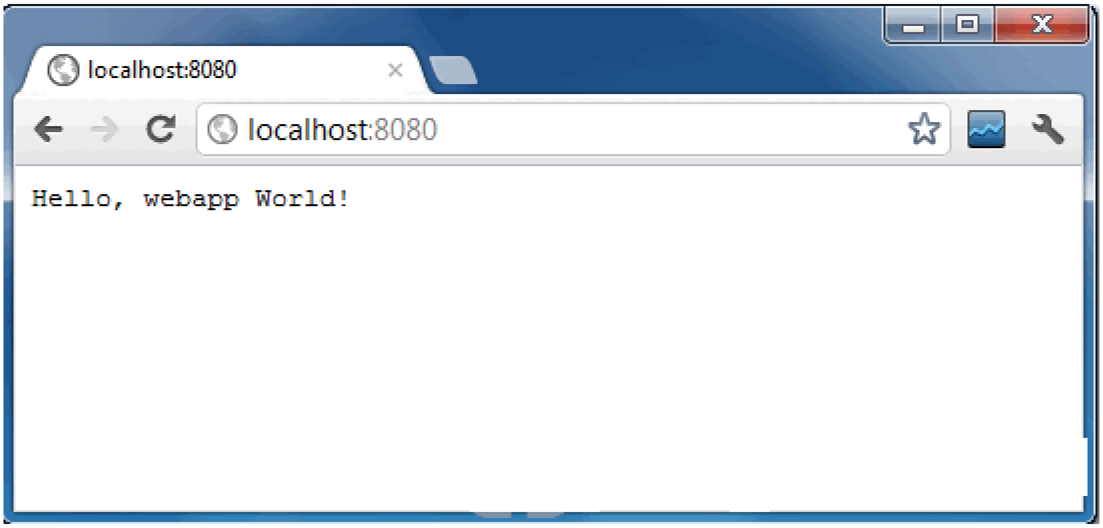
○ Other:

Figure 5.3 – Run it. By default, it will deploy to *http://localhost:8080*.



```
C:\Users\mkyong\workspace-3.7\HelloWorld/src
Warning: You are using a Python runtime (2.7) that is more recent than the production runtime environment (2.5). Your application may use featu
INFO     2012-04-03 05:06:24,086 appengine_rpc.py:160] Server: appengine.google.com
INFO     2012-04-03 05:06:24,104 appcfg.py:581] Checking for updates to the SDK.
INFO     2012-04-03 05:06:25,756 appcfg.py:599] The SDK is up to date.
WARNING  2012-04-03 05:06:25,756 datastore_file_stub.py:513] Could not read datastore data from c:\users\mkyong\appdata\local\temp\dev_appserve
WARNING  2012-04-03 05:06:25,799 dev_appserver.py:3375] Could not initialize images API; you are likely missing the Python "PIL" module. Import
INFO     2012-04-03 05:06:25,819 dev_appserver_multiprocess.py:650] Running application dev~mkyong-python on port 8080: http://localhost:8080
INFO     2012-04-03 05:06:25,821 dev_appserver_multiprocess.py:652] Admin console is available at: http://localhost:8080/_ah/admin
```

Figure 5.4 – Done.



```
Hello, webapp World!
```

5. Deploy to Google App Engine

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

application. Review "`app.yaml`" again, this web app will be deployed to GAE with application ID

"**mkyong-python**".

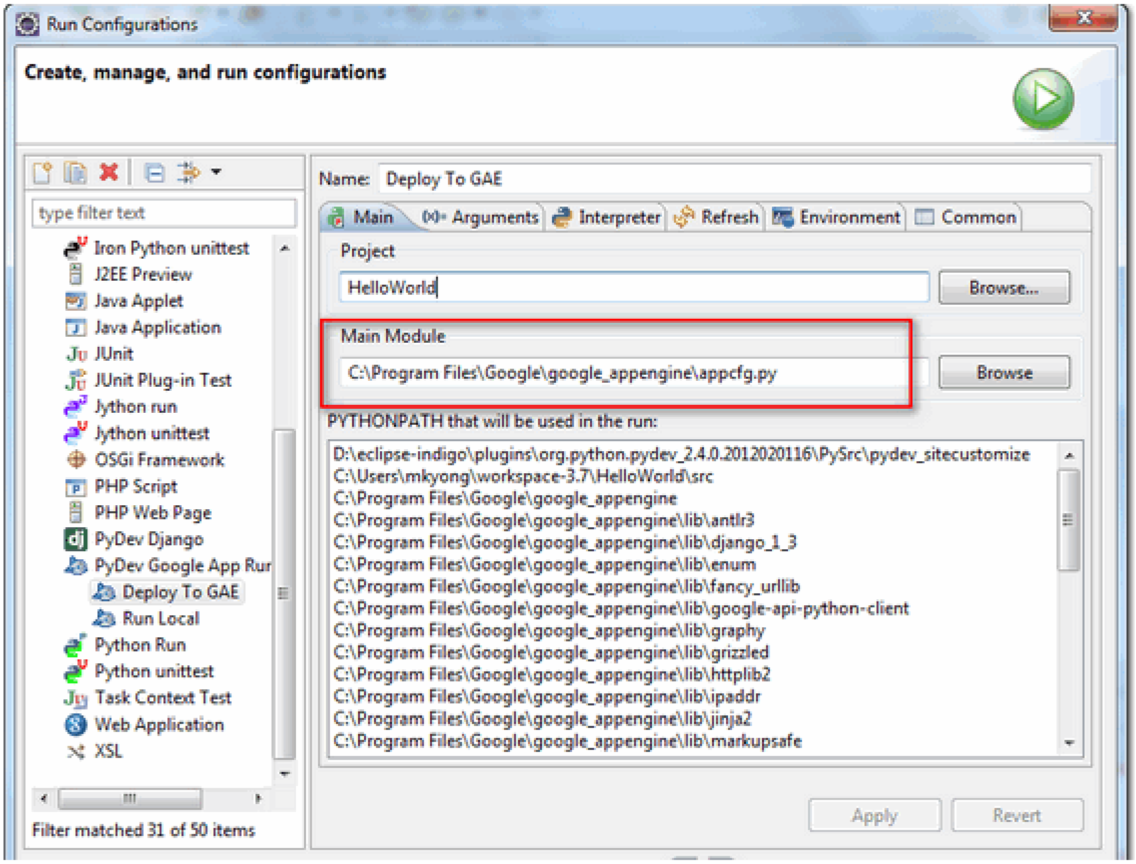
File : app.yaml

```
application: mkyong-python
version: 1
runtime: python
api_version: 1

handlers:
- url: /.*
  script: helloworld.py
```
Copy

To deploy to GAE, see following steps :

Figure 5.1 – Create another new “PyDev Google App Run”, In Main tab -> Main module, manually type

the directory path of “**appcfg.py**“.



Name: Deploy To GAE

Project: HelloWorld

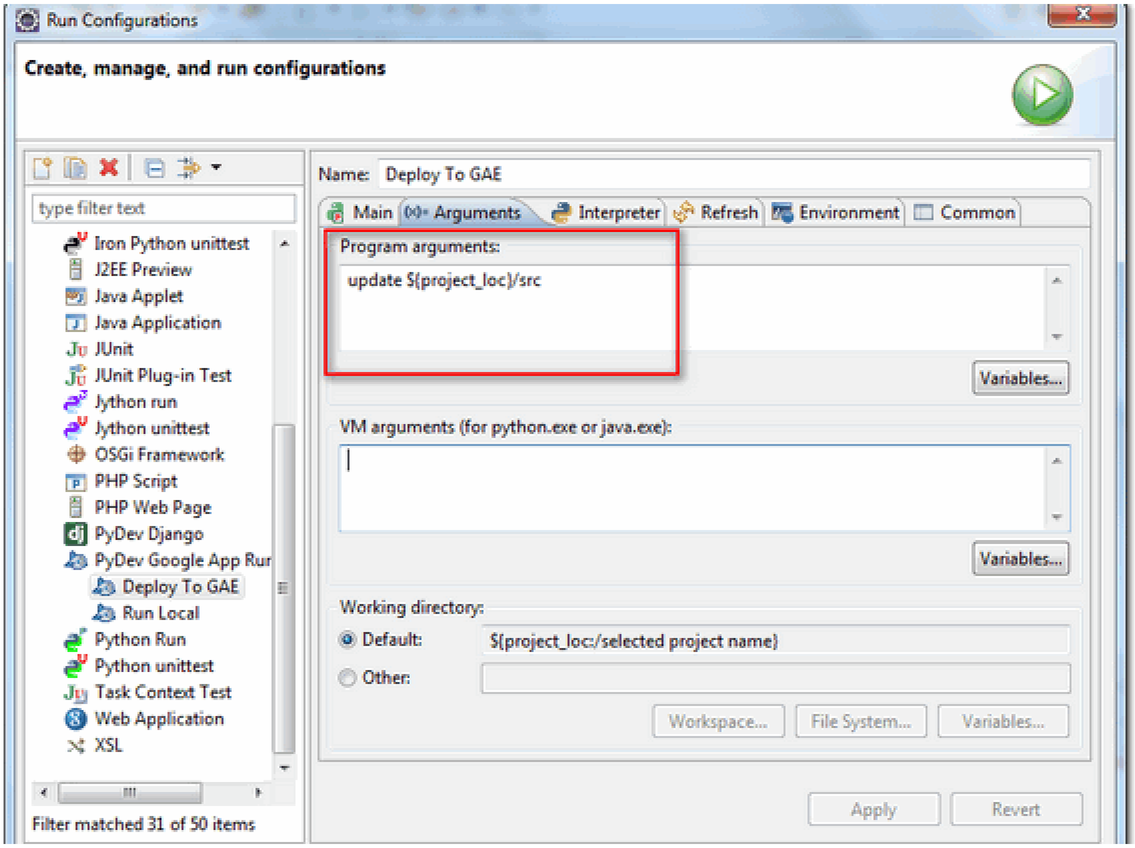
Main Module: C:\Program Files\Google\google_appengine\appcfg.py

PYTHONPATH that will be used in the run:

```
D:\eclipse-indigo\plugins\org.python.pydev_2.4.0.2012020116\PySrc\pydev_sitecustomize
C:\Users\mkyong\workspace-3.7\HelloWorld\src
C:\Program Files\Google\google_appengine
C:\Program Files\Google\google_appengine\lib\antlr3
C:\Program Files\Google\google_appengine\lib\django_1_3
C:\Program Files\Google\google_appengine\lib\enum
C:\Program Files\Google\google_appengine\lib\fancy_urllib
C:\Program Files\Google\google_appengine\lib\google-api-python-client
C:\Program Files\Google\google_appengine\lib\graphy
C:\Program Files\Google\google_appengine\lib\grizzled
C:\Program Files\Google\google_appengine\lib\httplib2
C:\Program Files\Google\google_appengine\lib\ipaddr
C:\Program Files\Google\google_appengine\lib\jinja2
C:\Program Files\Google\google_appengine\lib\markupsafe
```

Filter matched 31 of 50 items

Figure 5.2 – In Arguments tab -> Program arguments, put “**update ${project_loc}/src**“.



Name: Deploy To GAE

Program arguments:

```
update ${project_loc}/src
```

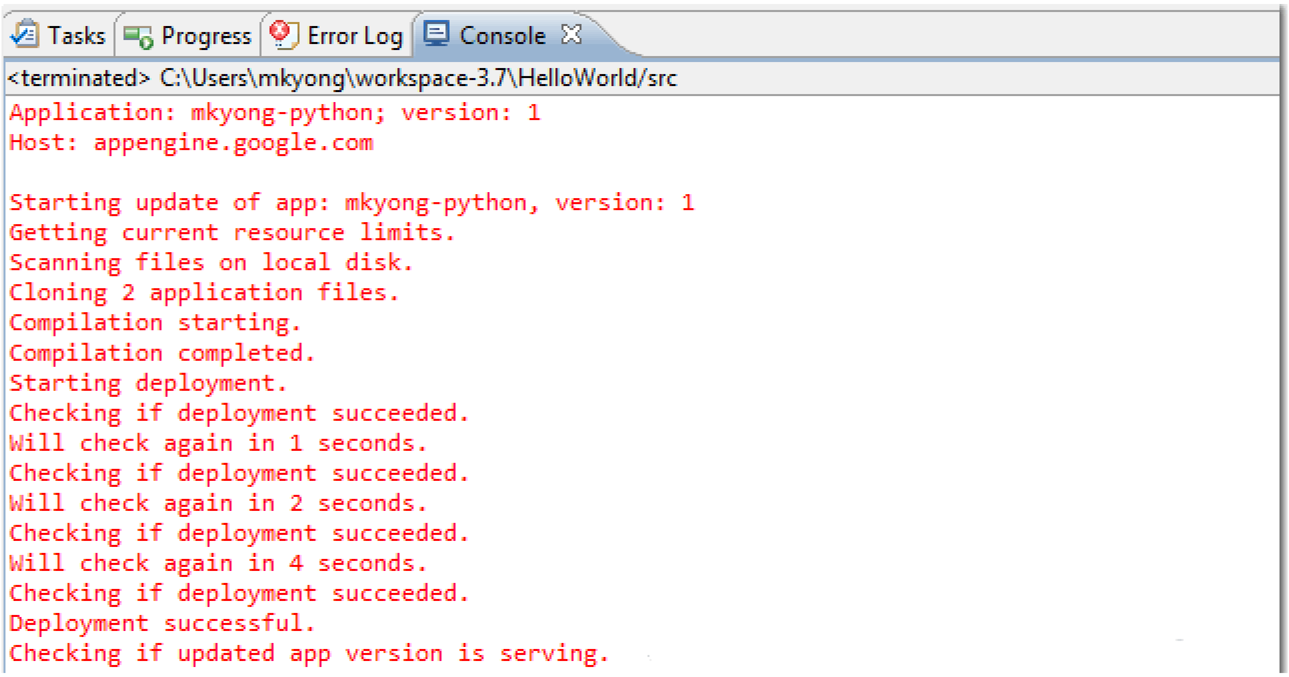
VM arguments (for python.exe or java.exe):

Working directory:

● Default: ${project_loc:/selected project name}

○ Other:

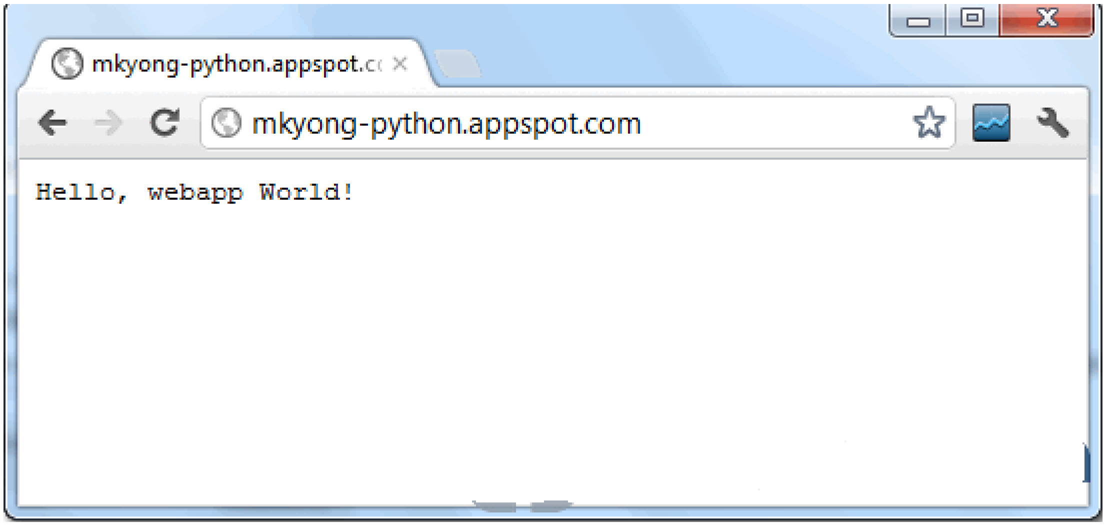
Figure 5.3 – During deploying process, you need to type your GAE email and password for authentication.



```
<terminated> C:\Users\mkyong\workspace-3.7\HelloWorld/src
Application: mkyong-python; version: 1
Host: appengine.google.com

Starting update of app: mkyong-python, version: 1
Getting current resource limits.
Scanning files on local disk.
Cloning 2 application files.
Compilation starting.
Compilation completed.
Starting deployment.
Checking if deployment succeeded.
Will check again in 1 seconds.
Checking if deployment succeeded.
Will check again in 2 seconds.
Checking if deployment succeeded.
Will check again in 4 seconds.
Checking if deployment succeeded.
Deployment successful.
Checking if updated app version is serving.
```

Figure 5.4 – If success, the web app will be deployed to – http://mkyong-python.appspot.com/.

OUTPUT:



```
Hello, webapp World!
```

RESULT:

Thus a hello world web application has been launched using GAE.

EX.NO:5	SIMULATE A CLOUD SCENARIO USING CLOUDSIM AN RUN A SCHEDULING ALGORITHM

Aim:

To Simulate a cloud scenario using CloudSim and run a scheduling algorithm

Procedure:

What is Cloudsim?

CloudSim is a simulation toolkit that supports the modeling and simulation of the core functionality of cloud, like job/task queue, processing of events, creation of cloud entities(datacenter, datacenter brokers, etc), communication between different entities, implementation of broker policies, etc. This toolkit allows to:

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

Core features of CloudSim are:

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

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

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

6. 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), stor
```

7. The third step is to create a broker:

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

8. 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())
```

9. Submit the VM list to the broker:

```
broker.submitVmList(vmlist)
```

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

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

11. Submit the cloudlet list to the broker:

```
broker.submitCloudletList(cloudletList)
```

12. Start the simulation:

```
CloudSim.startSimulation()
```

Sample Output from the Existing Example:

```
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>null
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
```

OUTPUT:

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

CloudSimExample1 finished!
```

RESULT:

Thus a hello world web application has been launched using GAE.

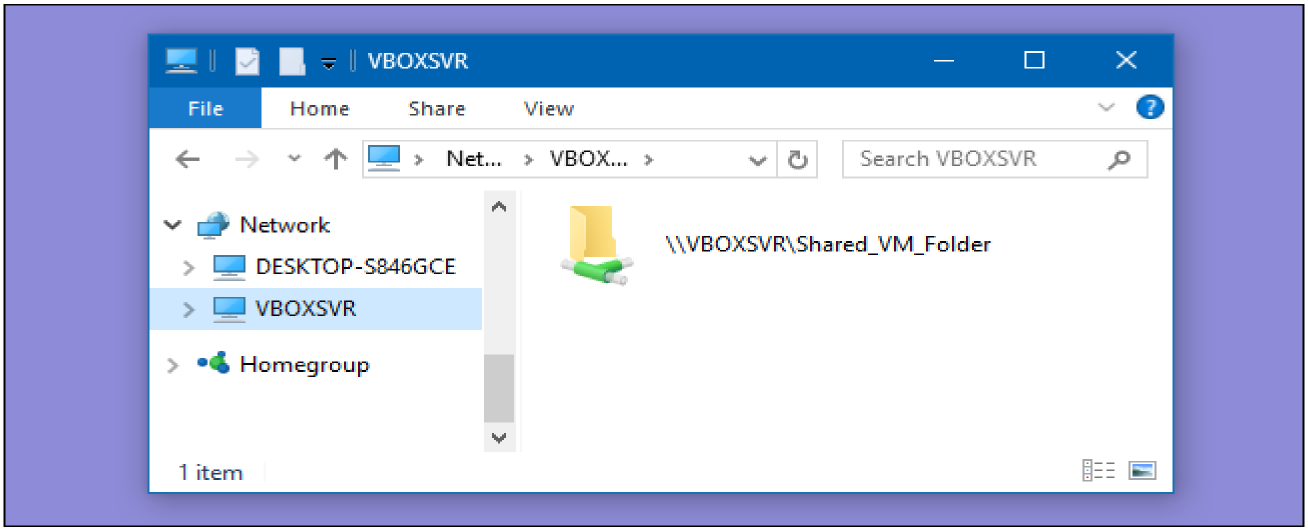
EX.NO:6	**COPY FILES FROM ONE VIRTUAL MACHINE TO ANOTHER**

AIM:

To write a procedure to copy files from one virtual machine to another

PROCEDURE:

1.Create one shared folder in virtual box



\\VBOXSVR\Shared_VM_Folder

VirtualBox

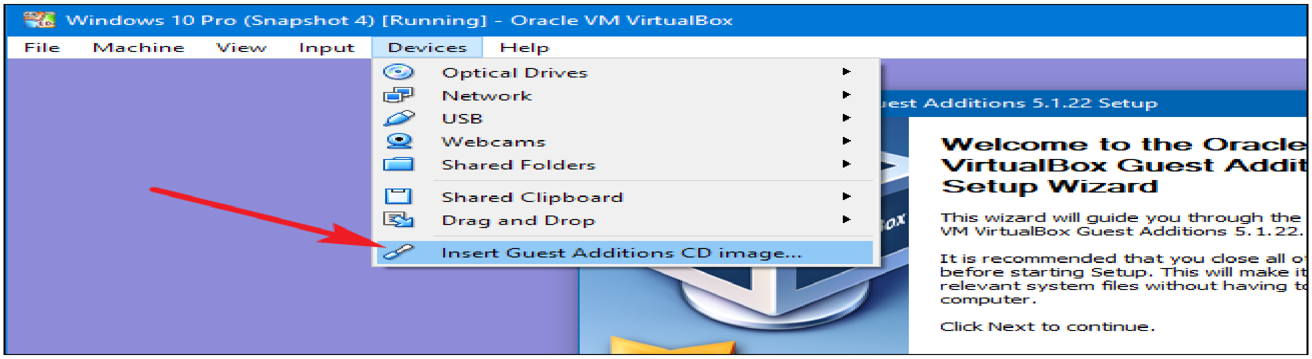
VirtualBox's Shared Folders feature works with both Windows and Linux guest operating systems. To use

the feature, you first need to install VirtualBox's Guest Additions in the guest virtual machine.

With the virtual machine running, click the "Devices" menu and choose the "Insert Guest Additions CD

image" option. This inserts a virtual CD that you can use within the guest operating system to install the

Guest Additions.


Windows 10 Pro (Snapshot 4) [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Optical Drives
Network
USB
Webcams
Shared Folders
Shared Clipboard
Drag and Drop
Insert Guest Additions CD image...
est Additions 5.1.22 Setup

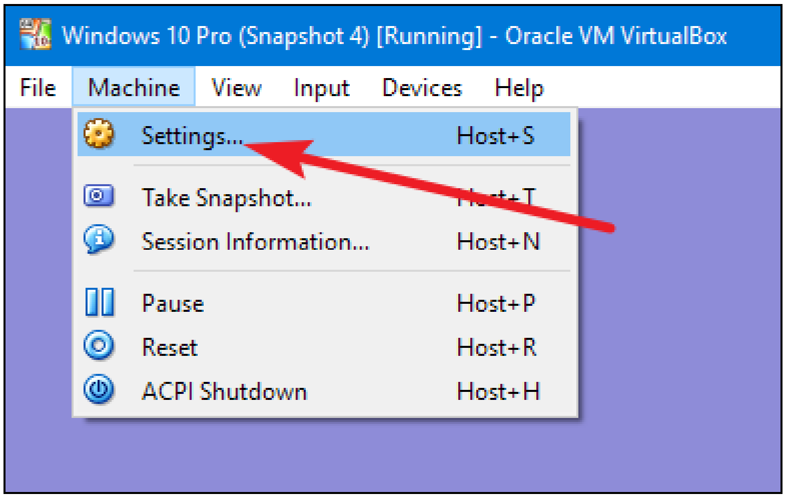
Welcome to the Oracle
VirtualBox Guest Addit
Setup Wizard

This wizard will guide you through the
VM VirtualBox Guest Additions 5.1.22.

It is recommended that you close all o
before starting Setup. This will make it
relevant system files without having t
computer.

Click Next to continue.

After the Guest Additions are installed, open the “Machine” menu and click the “Settings” option.



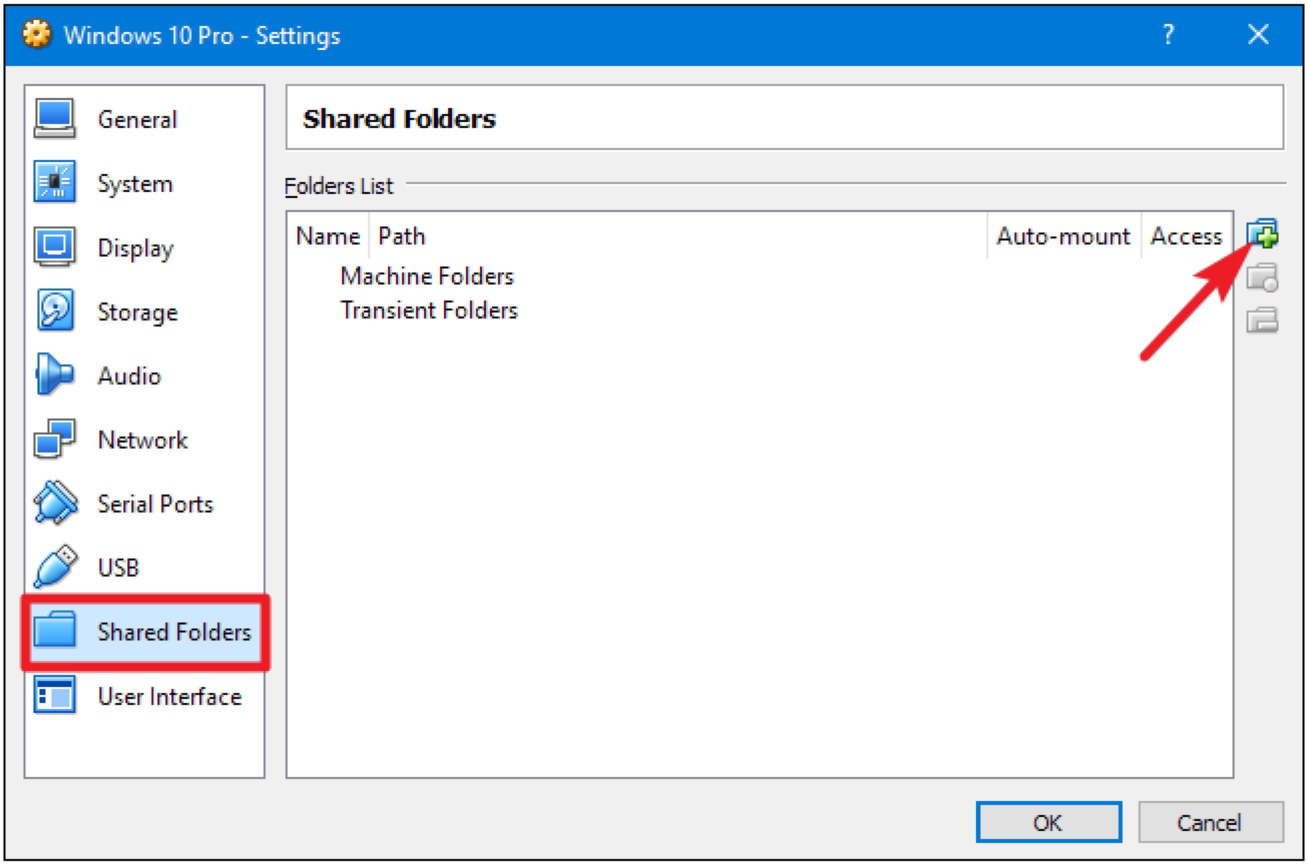
In the “Settings” window, switch to the “Shared Folders” tab. Here you can see any shared folders you’ve

set up. There are two types of shared folders. Machine Folders are permanent folders that are shared until

you remove them. Transient Folders are temporary and are automatically removed when you restart or shut down

the virtual machine.

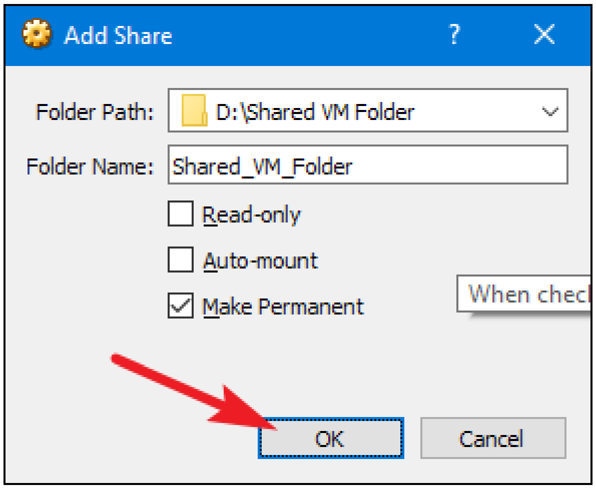
Click the “Add” button (the folder with a plus on it) to create a new shared folder.



In the "Add Share" window, you can specify the following:

- **Folder Path:** This is the location of the shared folder on your host operating system (your real PC).
- **Folder Name:** This is how the shared folder will appear inside the guest operating system.
- **Read-only:** By default, the virtual machine has full read-write access to the shared folder. Enable the
- "Read-only" checkbox if you want the virtual machine only to be able to read files from the shared
- folder, but not modify them.
- **Auto-mount:** This option makes the guest operating system attempt to automatically mount the folder
- when it boots.
- **Make Permanent:** This option makes the shared folder a Machine Folder. If you don't select this
- option, it becomes a transient folder that is removed with the virtual machine restarts.

Make all your choices and then hit the "OK" button.



Add Share

Folder Path: D:\Shared VM Folder

Folder Name: Shared_VM_Folder

☐ Read-only

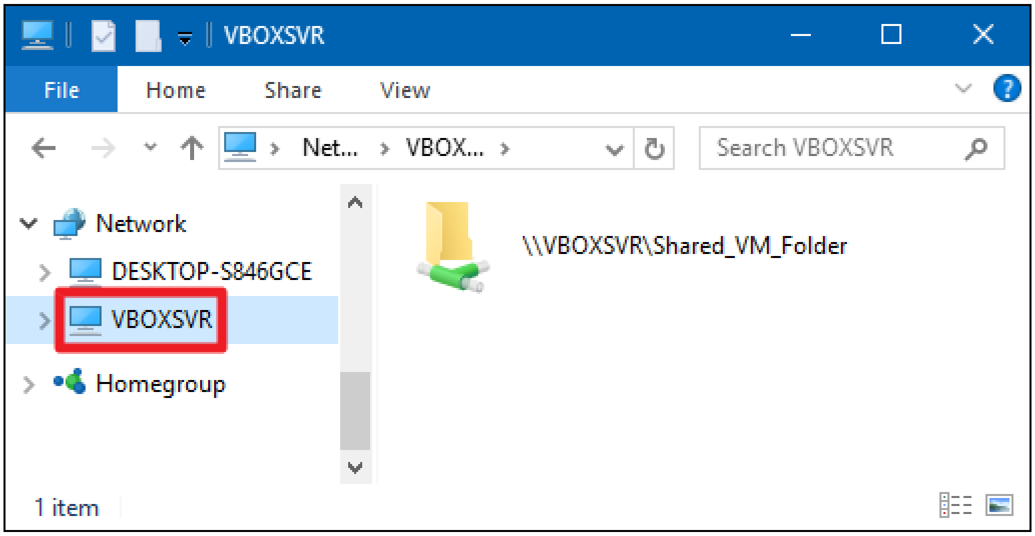
☐ Auto-mount

☑ Make Permanent

You should now see the shared folders appear as network file shares. If you're using a Windows guest operating system, open File Explorer, select "Network", and then look under the "VBOXSRV" computer.

OUTPUT:

"VBOXSRV" contains shared files.



\\VBOXSVR\Shared_VM_Folder

1 item

RESULT:

Thus a procedure to copy files from one virtual ,machine to another virtual machine was executed successfully.

Ex.No.7	**CREATE, DEPLOY AND LAUNCH VIRTUAL MACHINES IN OPENSTACK**

AIM:

To Write a procedure to create ,deploy and launch a virtul machine s in open stack.

REQUIREMENTS

1. Install OpenStack in RHEL and CentOS 7
2. Configure OpenStack Networking Service

PROCEDURE:

Step 1: Allocate Floating IP to OpenStack

1. Before you deploy an **OpenStack** image, first you need to assure that all pieces are in place and we'll
2. start by allocating floating IP.

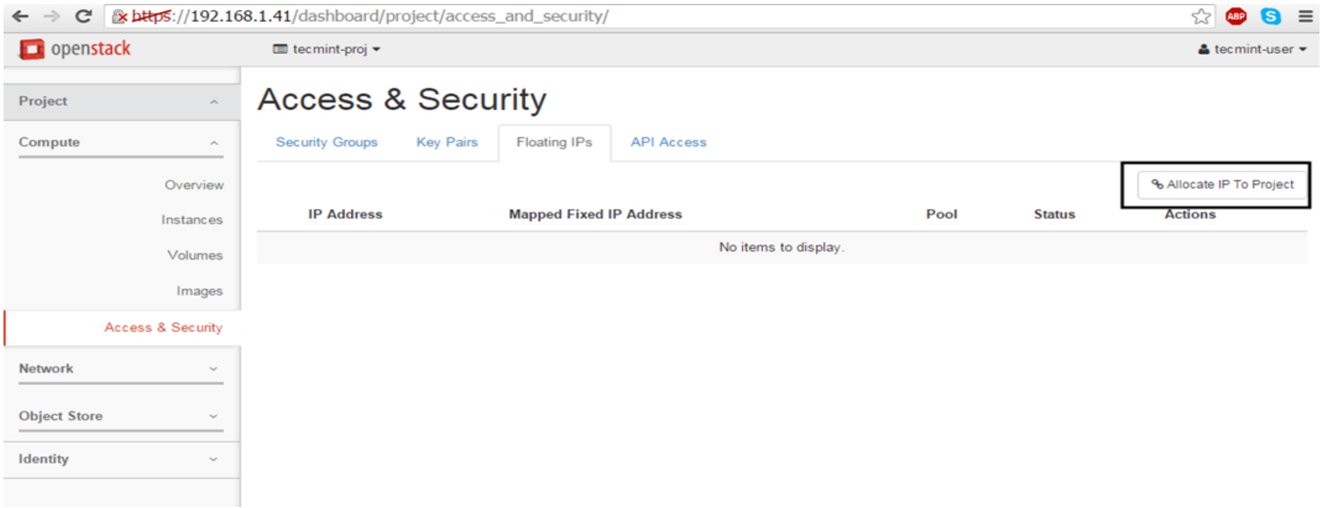
Floating IP allows external access from outside networks or internet to an Openstack virtual machine. In

order to create floating IPs for your project, login with your **user** credentials and go to **Project** ->

Compute -> **Access & Security** -> **Floating IPs** tab and click on **Allocate IP** to The Project.

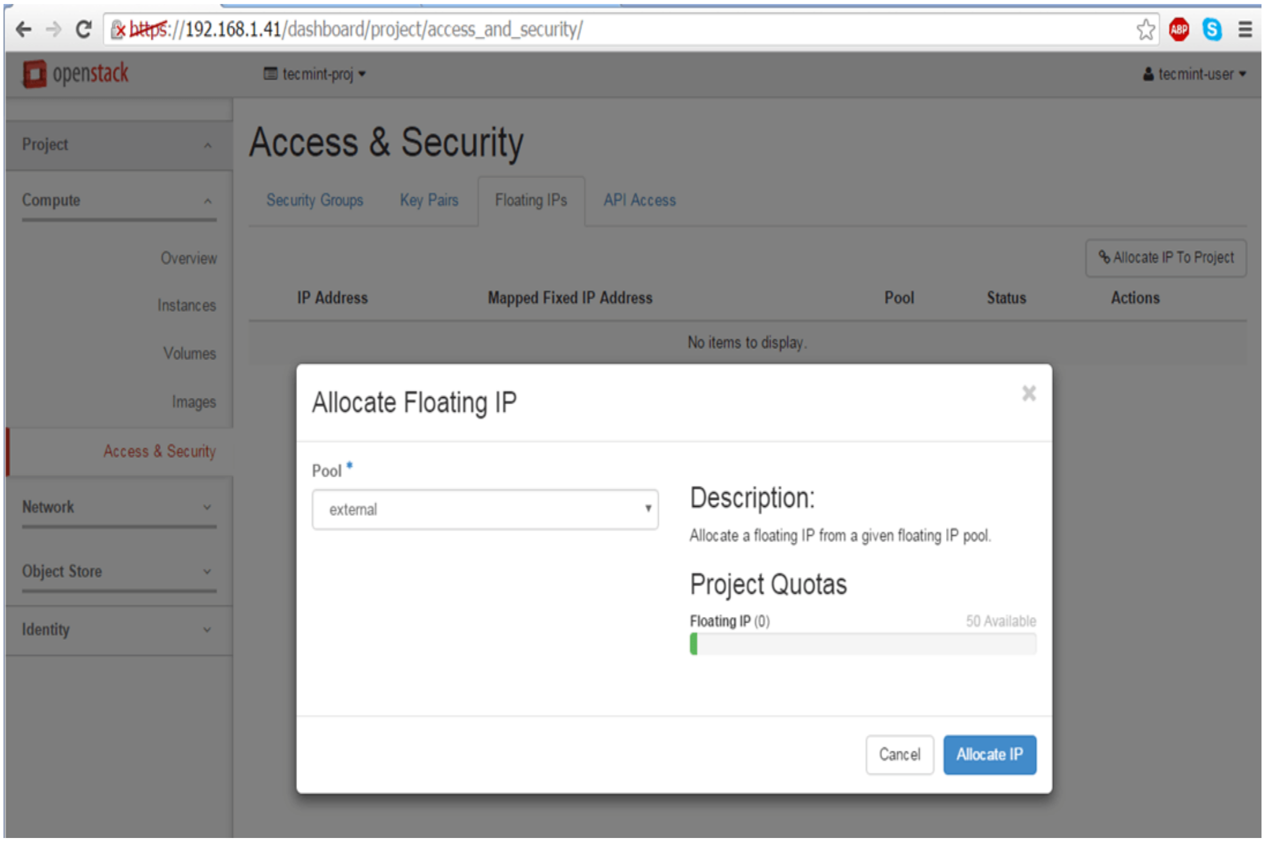
Choose external **Pool** and hit on **Allocate IP** button and the IP address should appear in dashboard. It's a good

idea to allocate a Floating IP for each instance you run.



IP Address	Mapped Fixed IP Address	Pool	Status	Actions
No items to display.				

Allocate Floating IP to Project in OpenStack



IP Address	Mapped Fixed IP Address	Pool	Status	Actions

No items to display.

Allocate Floating IP

Pool *: external

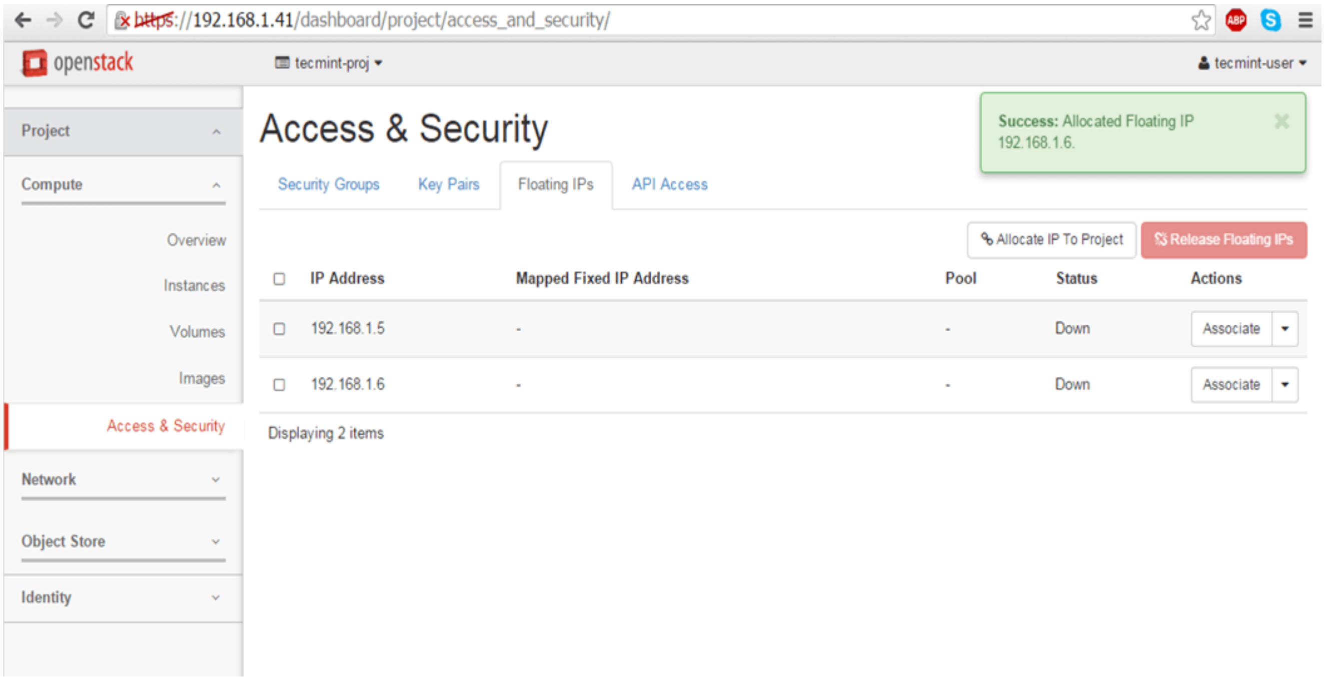
Description:

Allocate a floating IP from a given floating IP pool.

Project Quotas

Floating IP (0) 50 Available

Allocate Floating IP to External Pool



Success: Allocated Floating IP 192.168.1.6.

□	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
□	192.168.1.5	-	-	Down	Associate
□	192.168.1.6	-	-	Down	Associate

Displaying 2 items

Confirmation of Adding Floating IP

Step 2: Create an OpenStack Image

3. OpenStack images are just virtual machines already created by third-parties. You can create your own
4. customized images on your machine by installing an Linux OS in a virtual machine using a
5. virtualization tool, such as KVM, VirtualBox, VMware or Hyper-V.

Once you have installed the OS, just convert the file to raw and upload it to your OpenStack cloud infrastructure.

To deploy official images provided by major Linux distributions use the following links to download the

latest packaged images:

1. **CentOS 7** – http://cloud.centos.org/centos/7/images/
2. **CentOS 6** – http://cloud.centos.org/centos/6/images/
3. **Fedora 23** – https://download.fedoraproject.org/pub/fedora/linux/releases/23/Cloud/
4. **Ubuntu** – http://cloud-images.ubuntu.com/
5. **Debian** – http://cdimage.debian.org/cdimage/openstack/current/
6. **Windows Server 2012 R2** – https://cloudbase.it/windows-cloud-images/#download

Official images additionally contain the **cloud-init** package which is responsible with SSH key pair and user

data injection.

On this guide we'll deploy a test image, for demonstration purposes, based on a lightweight Cirros cloud

image which can be obtained by visiting the following link http://download.cirros-cloud.net/0.3.4/.

The image file can be used directly from the HTTP link or downloaded locally on your machine and

uploaded to OpenStack cloud.

To create an image, go **OpenStack** web panel and navigate to **Project** -> **Compute** -> **Images** and hit on

Create Image button. On the image prompt use the following settings and hit on **Create Image** when done.

Name: **tecmint-test**
Description: **Cirros test image**

Image Source: **Image Location** #Use Image File if you've downloaded the file locally on your hard disk
Image Location: **http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-i386-disk.img**
Format: **QCOWW2 – QEMU Emulator**
Architecture: **leave blank**
Minimum Disk: leave blank
Minimum RAM: **leave blank**
Image Location: **checked**
Public: **unchecked**
Protected: **unchecked**

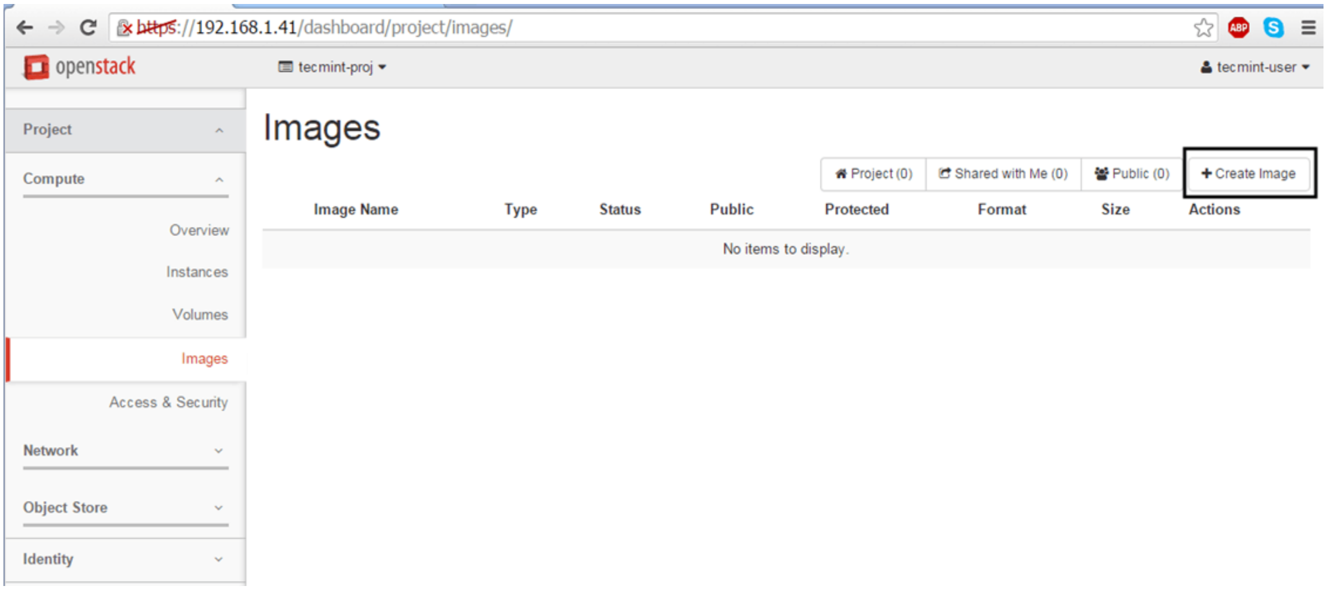
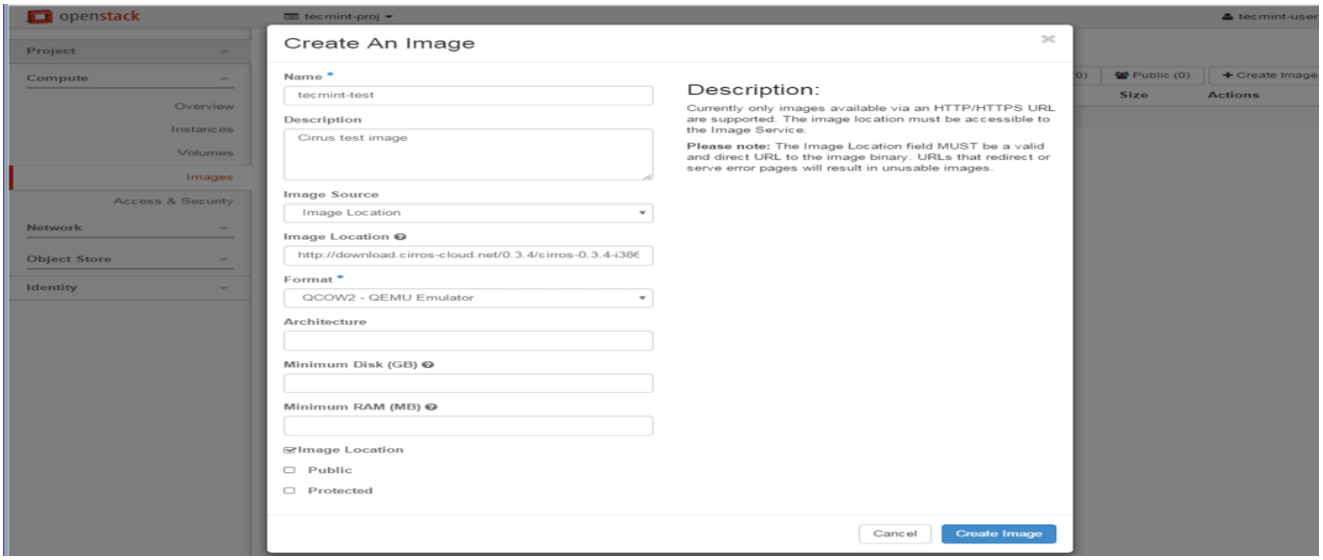


Image Name	Type	Status	Public	Protected	Format	Size	Actions
No items to display.							

Create Images in OpenStack



Name *
tecmint-test

Description
Cirrus test image

Image Source
Image Location

Image Location
http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-i38

Format *
QCOW2 - QEMU Emulator

Architecture

Minimum Disk (GB)

Minimum RAM (MB)

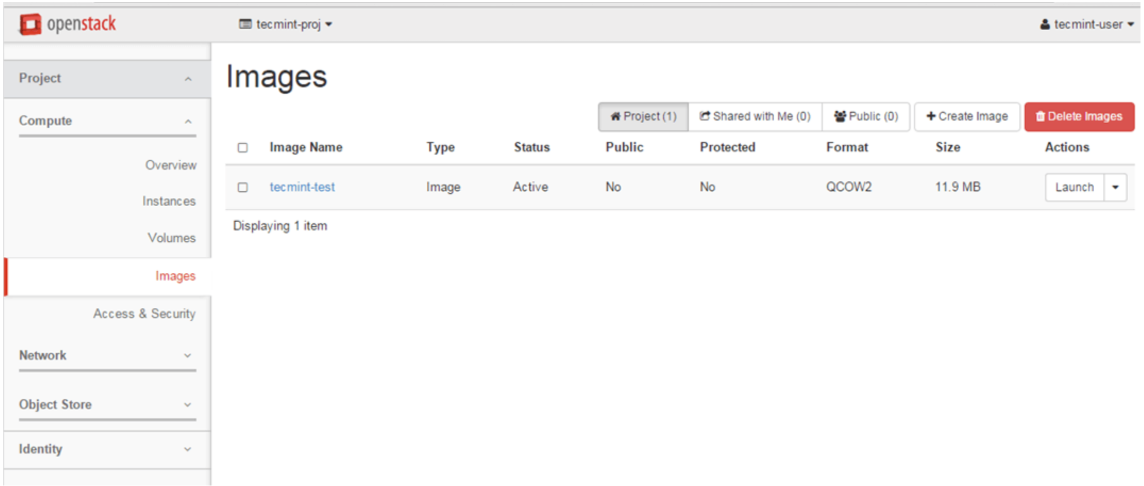
☑ Image Location
□ Public
□ Protected

Description:

Currently only images available via an HTTP/HTTPS URL are supported. The image location must be accessible to the Image Service.

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Add OpenStack Image Details



□	Image Name	Type	Status	Public	Protected	Format	Size	Actions
□	tecmint-test	Image	Active	No	No	QCOW2	11.9 MB	Launch

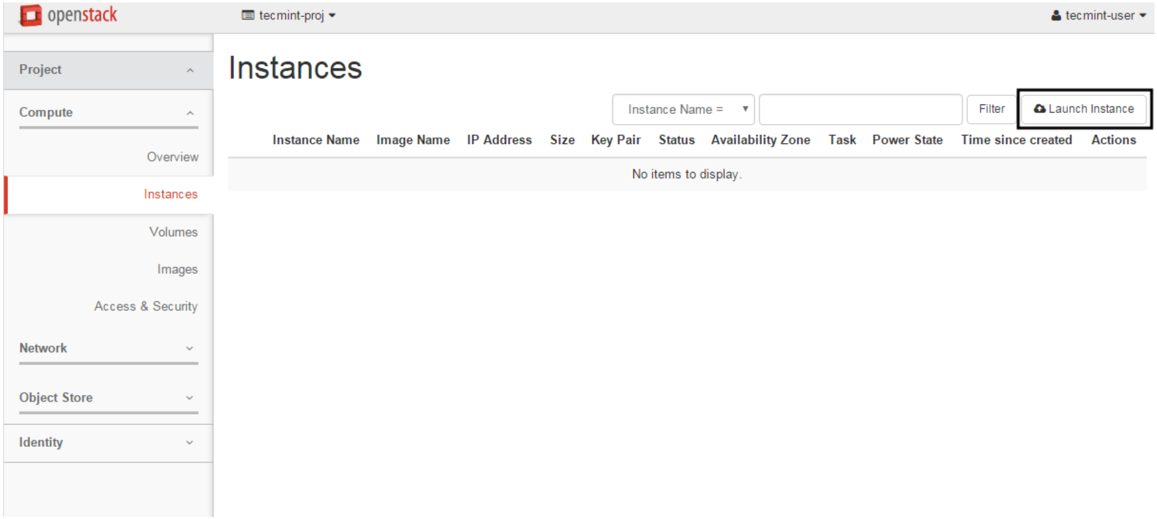
Displaying 1 item

OpenStack Images

Step 3: Launch an Image Instance in OpenStack

3. Once you've created an image you're good to go. Now you can run the virtual machine based on the
4. image created earlier in your cloud environment.

Move to **Project** -> **Instances** and hit on **Launch Instance** button and a new window will appear.



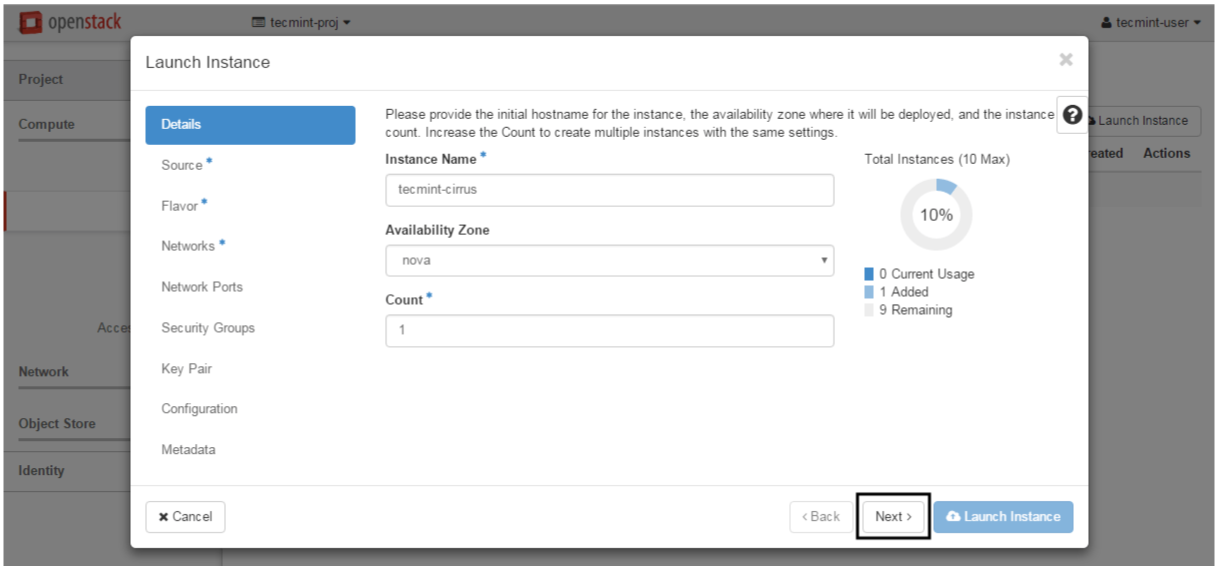
Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions

No items to display.

Launch Image Instance in Openstack

5. On the first screen add a name for your instance, leave the **Availability Zone** to nova, use one instance count
6. and hit on **Next** button to continue.

Choose a descriptive **Instance Name** for your instance because this name will be used to form the virtual machine hostname.



Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name * tecmint-cirrus

Availability Zone nova

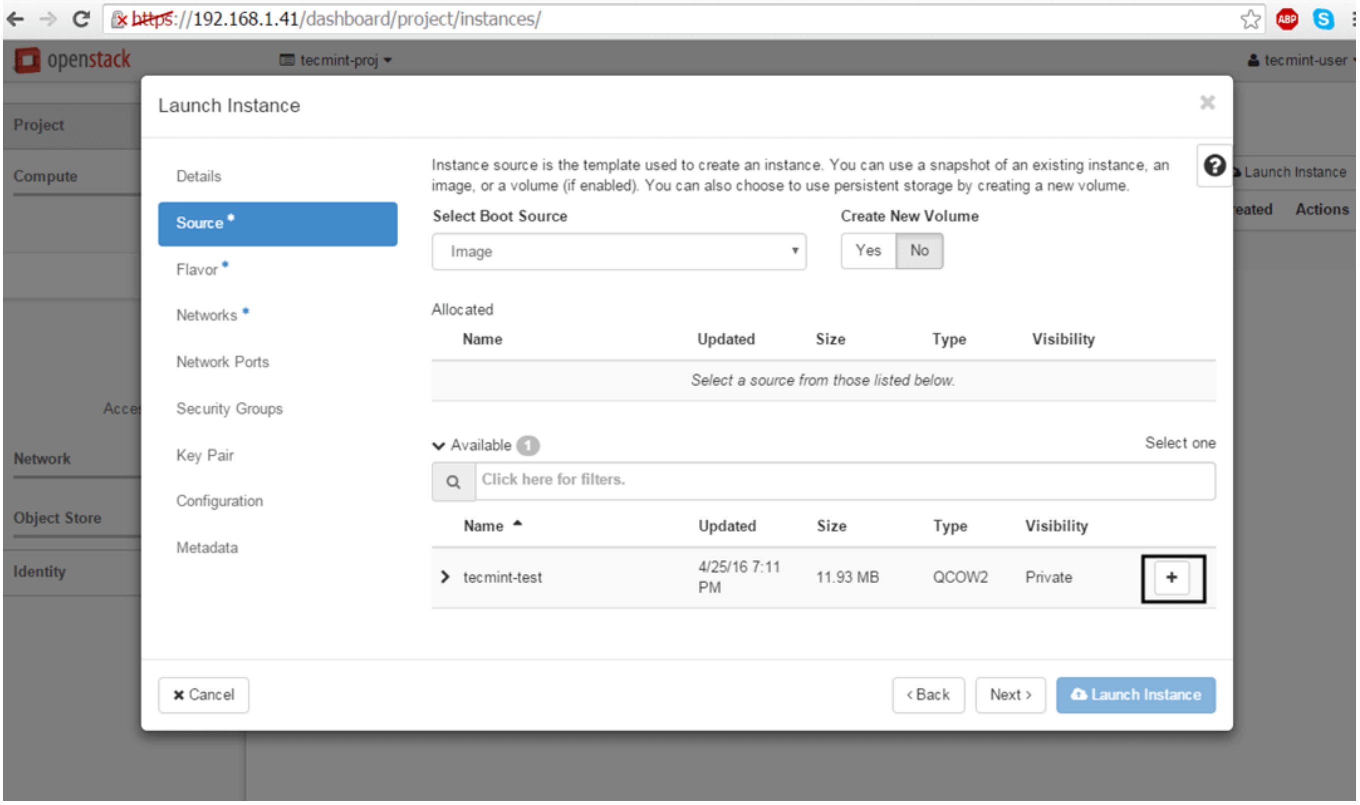
Count * 1

Total Instances (10 Max) 10%

- 0 Current Usage
- 1 Added
- 9 Remaining

Add Hostname to OpenStack Instance

6. Next, select Image as a **Boot Source**, add the **Cirros** test image created earlier by hitting the + button
7. and hit
8. **Next** to proceed further.



Instance source is the template used to create an instance. You can use a snapshot of an existing instance, an image, or a volume (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Create New Volume: Yes [✓ No]

Allocated

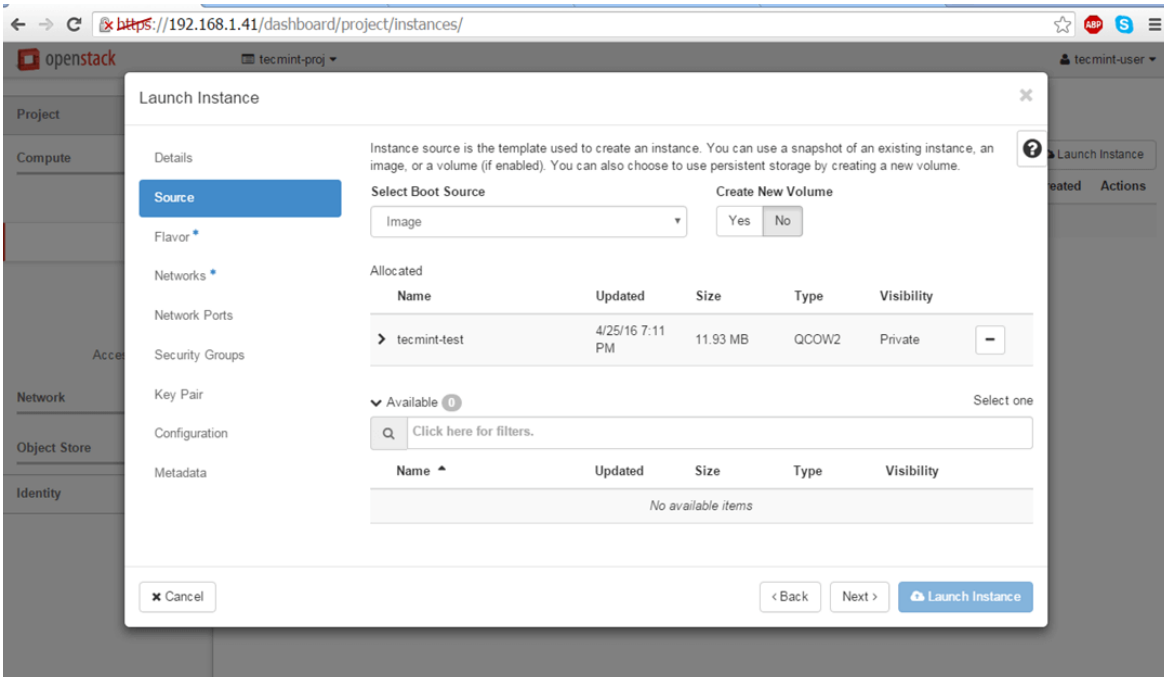
Name	Updated	Size	Type	Visibility
Select a source from those listed below.				

Available 1 — Select one

Click here for filters.

Name	Updated	Size	Type	Visibility	
tecmint-test	4/25/16 7:11 PM	11.93 MB	QCOW2	Private	+

Select OpenStack Instance Boot Source



Launch Instance

Instance source is the template used to create an instance. You can use a snapshot of an existing instance, an image, or a volume (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Create New Volume: Yes [✓ No]

Allocated

Name	Updated	Size	Type	Visibility
tecmint-test	4/25/16 7:11 PM	11.93 MB	QCOW2	Private

Available 0 — Select one

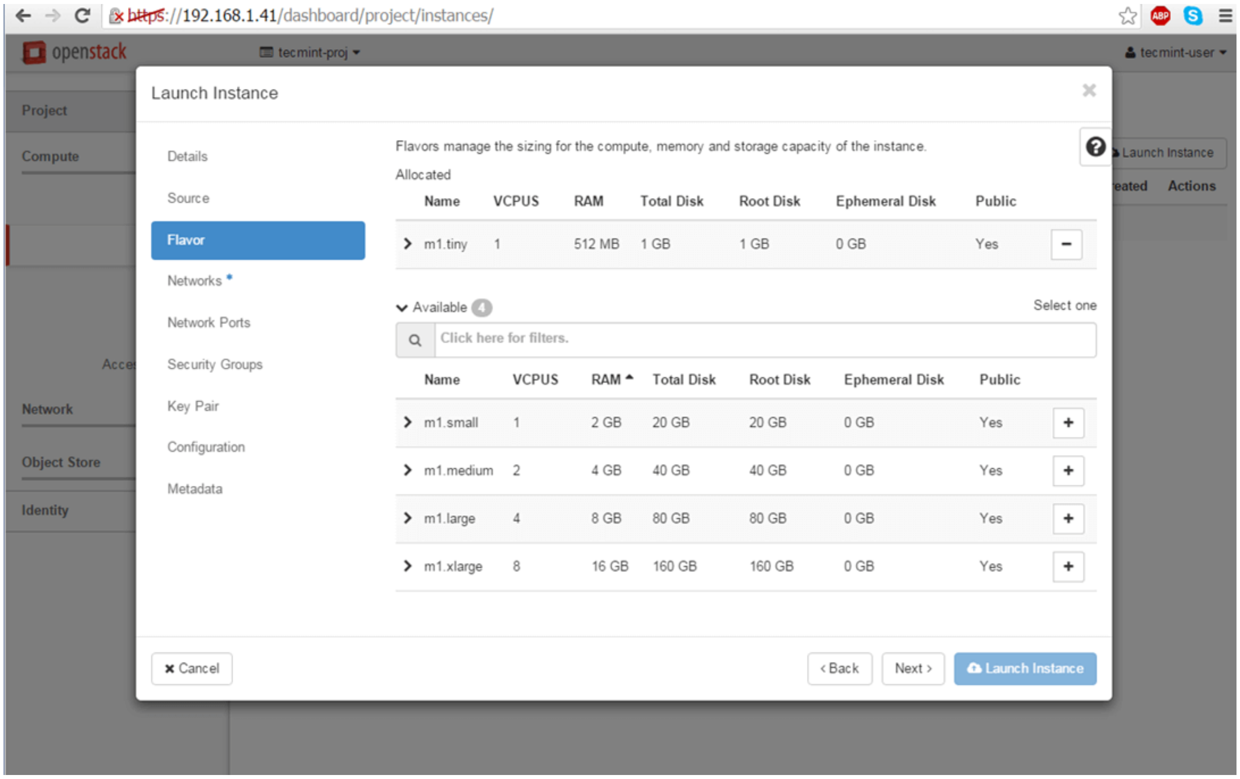
Click here for filters.

Name	Updated	Size	Type	Visibility

No available items

Add Cirros Text Image

7. Allocate the virtual machine resources by adding a flavor best suited for your needs and click on **Next** to move on.



Launch Instance

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes

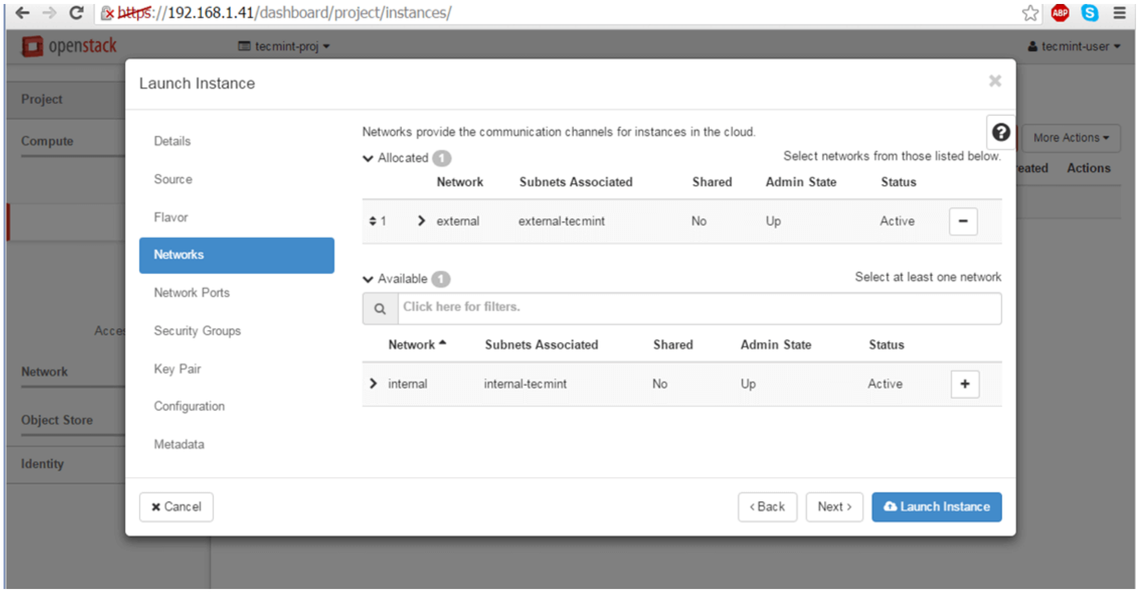
Available 4 — Select one

Click here for filters.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes
m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes

Add Resources to OpenStack Instance

7. Finally, add one of the OpenStack available networks to your instance using the + button and hit on
8. **Launch Instance** to start the virtual machine.



Networks provide the communication channels for instances in the cloud.

Allocated 1

Select networks from those listed below.

	Network	Subnets Associated	Shared	Admin State	Status	
1	external	external-tecmint	No	Up	Active	−

Available 1

Select at least one network

Click here for filters.

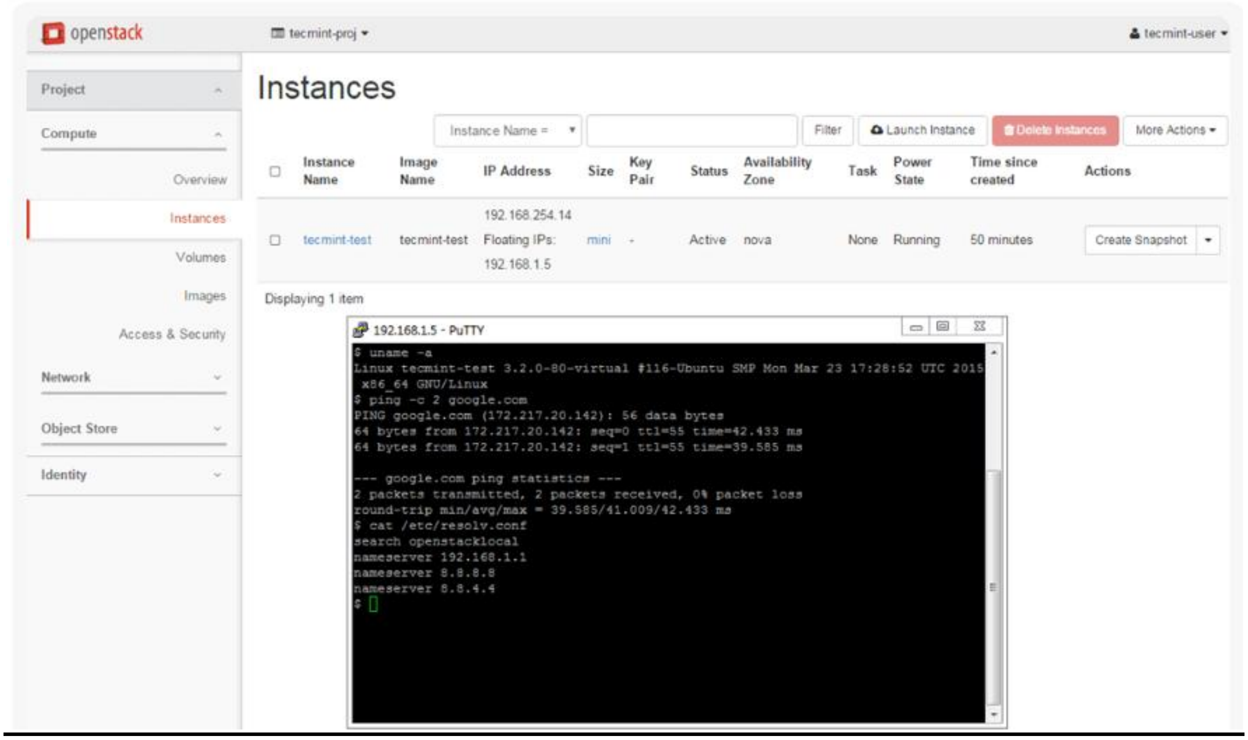
Network	Subnets Associated	Shared	Admin State	Status	
internal	internal-tecmint	No	Up	Active	+

Add Network to OpenStack Instance

9. Once the instance has been started, hit on the right arrow from **Create Snapshot** menu button and
10. choose **Associate Floating IP**.

Select one of the floating IP created earlier and hit on **Associate** button in order to make the instance reachable from your internal LAN.

OUTPUT:



□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	tecmint-test	tecmint-test	192.168.254.14 Floating IPs: 192.168.1.5	mini	-	Active	nova	None	Running	50 minutes	Create Snapshot ▾

Displaying 1 item

```
$ uname -a
Linux tecmint-test 3.2.0-80-virtual #116-Ubuntu SMP Mon Mar 23 17:28:52 UTC 2015
 x86_64 GNU/Linux
$ ping -c 2 google.com
PING google.com (172.217.20.142): 56 data bytes
64 bytes from 172.217.20.142: seq=0 ttl=55 time=42.433 ms
64 bytes from 172.217.20.142: seq=1 ttl=55 time=39.585 ms

--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 39.585/41.009/42.433 ms
$ cat /etc/resolv.conf
search openstacklocal
nameserver 192.168.1.1
nameserver 8.8.8.8
nameserver 8.8.4.4
$ 
```

RESULT:

Thus the procedure to deploy virtual machined in open stack was successfully executed.

Ex.No.8	**INSTALL HADOOP SINGLE NODE CLUSTER**

AIM:

To write a procedure to install a single node Hadoop cluster

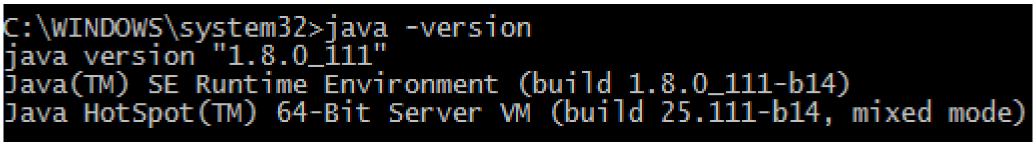
PROCEDURE:.

Java

Download the Java 1.8 from https://java.com/en/download/

Once installed confirm that you're running the correct version from command line using 'java -version'

command, output of which you can confirm in command line like this:



```
C:\WINDOWS\system32>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)
```

WinRAR

I've downloaded and installed WinRAR 64 bit release from http://www.rarlab.com/download.htm that will

later allow me to decompress Linux type tar.gz packages on Windows.

Hadoop

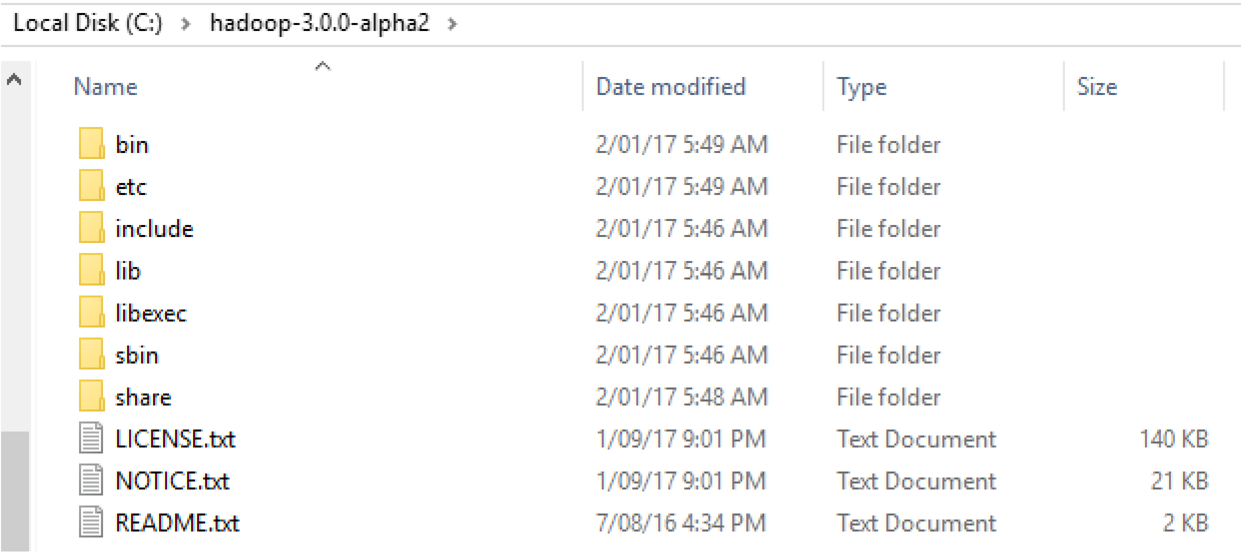
The next step was to install a Hadoop distribution. To do so, I've decided to download the most recent

release **Hadoop 3.0.0-alpha2** (25 Jan, 2017) in a binary form, from the Apache Download

Mirror at http://hadoop.apache.org/releases.html

Once the **hadoop-3.0.0-alpha2.tar.gz** (250 MB) downloaded, I've extracted it by using WinRAR (installed in

the previous step) into C:\hadoop-3.0.0-alpha2 folder:



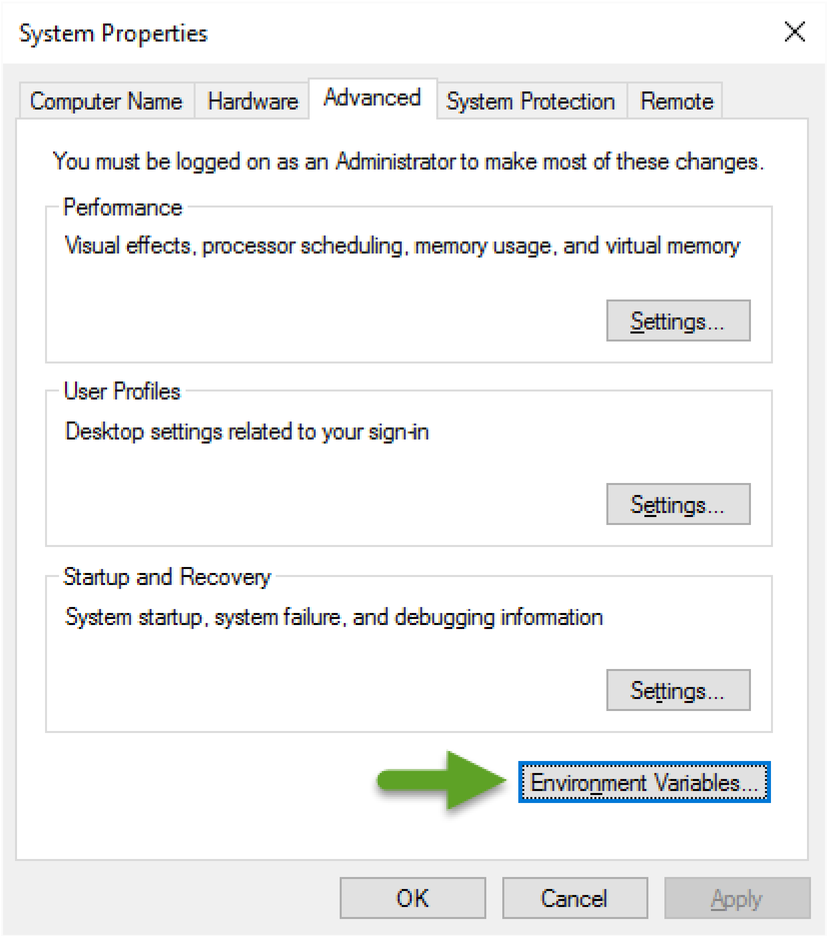
Local Disk (C:) › hadoop-3.0.0-alpha2 ›

Name	Date modified	Type	Size
bin	2/01/17 5:49 AM	File folder	
etc	2/01/17 5:49 AM	File folder	
include	2/01/17 5:46 AM	File folder	
lib	2/01/17 5:46 AM	File folder	
libexec	2/01/17 5:46 AM	File folder	
sbin	2/01/17 5:46 AM	File folder	
share	2/01/17 5:48 AM	File folder	
LICENSE.txt	1/09/17 9:01 PM	Text Document	140 KB
NOTICE.txt	1/09/17 9:01 PM	Text Document	21 KB
README.txt	7/08/16 4:34 PM	Text Document	2 KB

Now that I had Hadoop downloaded, it was time to start the Hadoop cluster with a single node.

Setup Environmental Variables

In Windows 10 I’ve opened System Properties windows and clicked on Environment Variables button:



System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

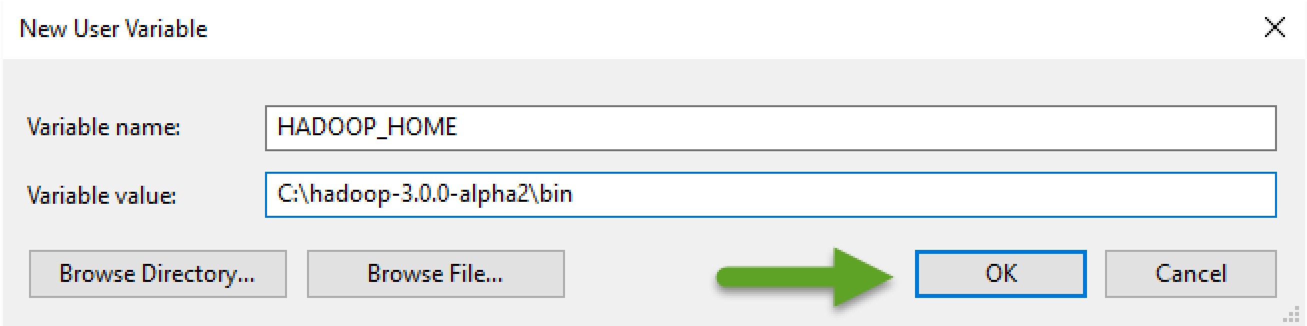
Settings...

Environment Variables...

OK | Cancel | Apply

hen created a new HADOOP_HOME variable and pointed the path to C:\hadoop-3.0.0-alpha2\bin folder on my

PC:

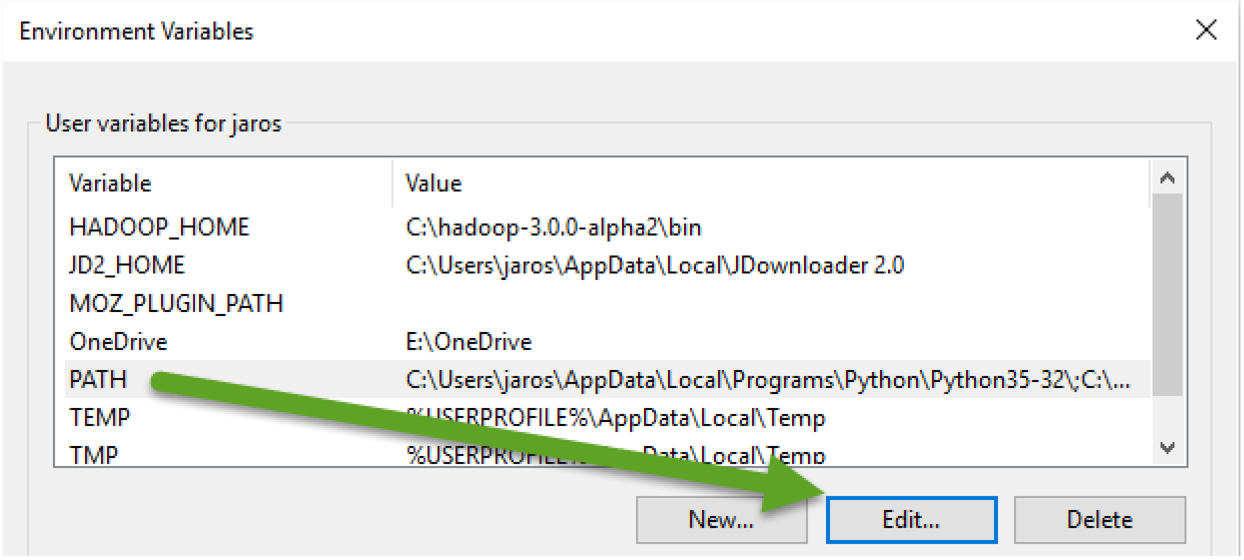


New User Variable

Variable name: HADOOP_HOME

Variable value: C:\hadoop-3.0.0-alpha2\bin

Next step was to add a Hadoop bin directory path to PATH variable. Clicked on PATH and pressed edit:

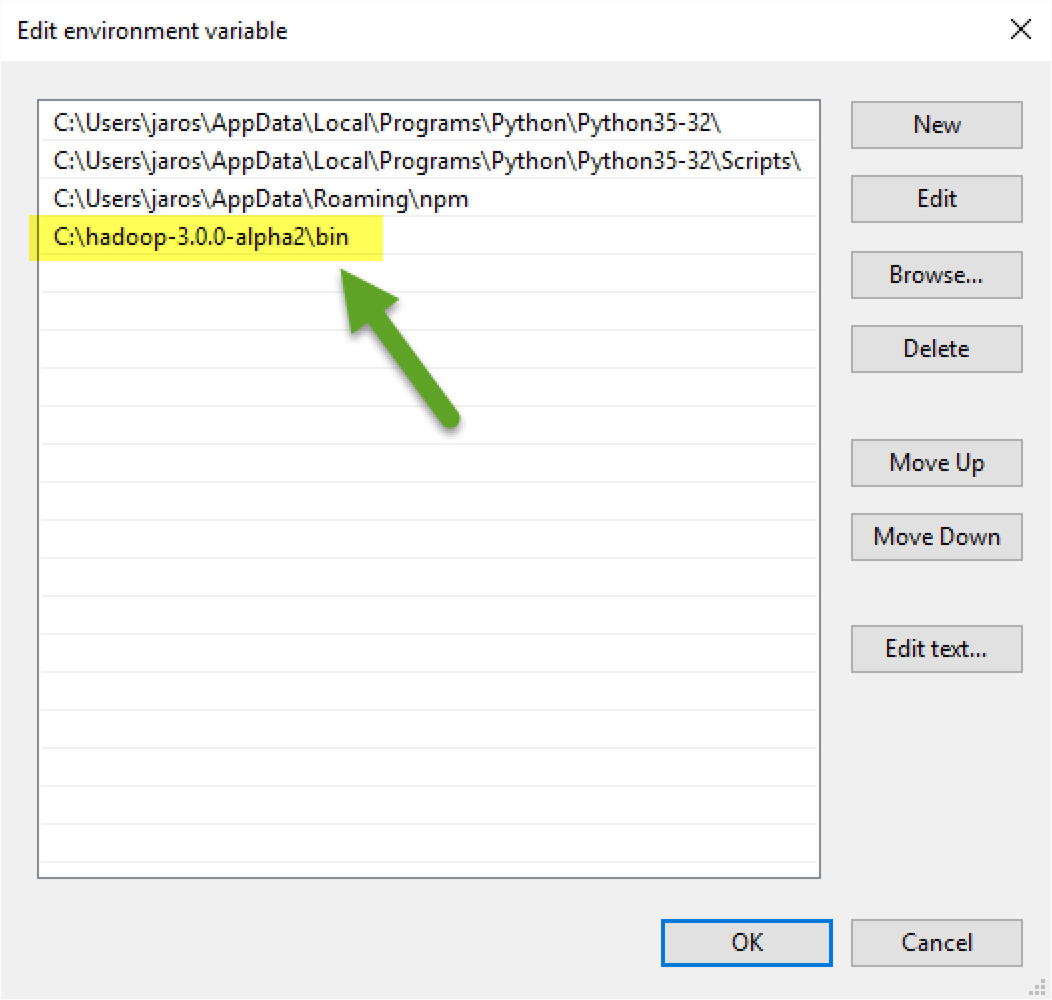


Environment Variables

User variables for jaros

Variable	Value
HADOOP_HOME	C:\hadoop-3.0.0-alpha2\bin
JD2_HOME	C:\Users\jaros\AppData\Local\JDownloader 2.0
MOZ_PLUGIN_PATH	
OneDrive	E:\OneDrive
PATH	C:\Users\jaros\AppData\Local\Programs\Python\Python35-32\;C:\...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

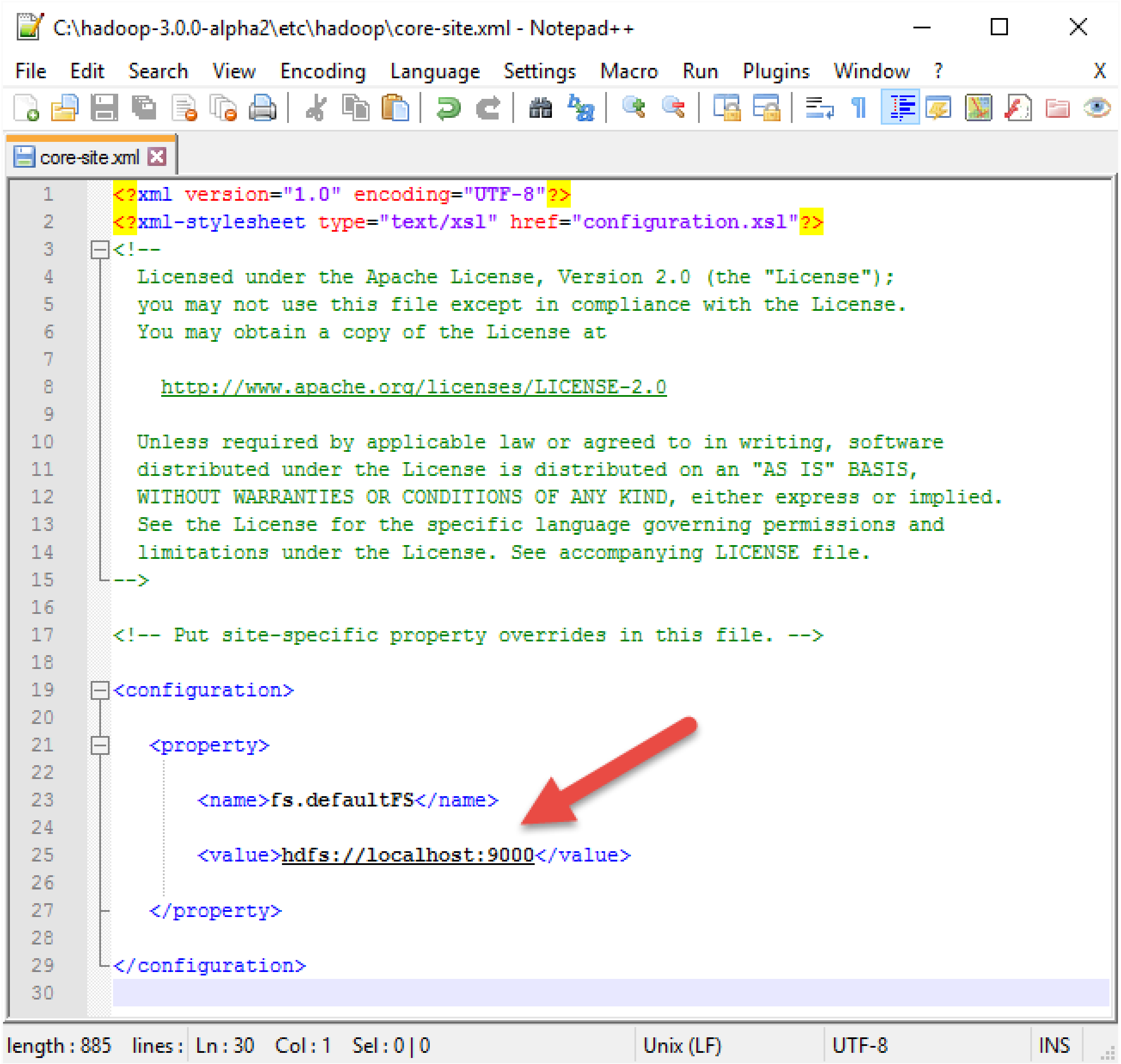
Then added a ‘C:\hadoop-3.0.0-alpha2\bin’ path like this and pressed OK:



C:\Users\jaros\AppData\Local\Programs\Python\Python35-32\
C:\Users\jaros\AppData\Local\Programs\Python\Python35-32\Scripts\
C:\Users\jaros\AppData\Roaming\npm
C:\hadoop-3.0.0-alpha2\bin

Edit Hadoop Configuration

C:\hadoop-3.0.0-alpha2\etc\hadoop\core-site.xml file, just like this:



C:\hadoop-3.0.0-alpha2\etc\hadoop\core-site.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
X
core-site.xml
length : 885 lines : Ln : 30 Col : 1 Sel : 0 | 0
Unix (LF)
UTF-8
INS

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

   <property>

       <name>fs.defaultFS</name>

       <value>hdfs://localhost:9000</value>

   </property>

</configuration>

```

Next go to C:\hadoop-3.0.0-alpha2\etc\hadoop folder and renamed mapred-site.xml.template to mapred-site.xml.

edited the mapred-site.xml file adding the following XML Yarn configuration for Mapreduce:

<configuration>

<property>

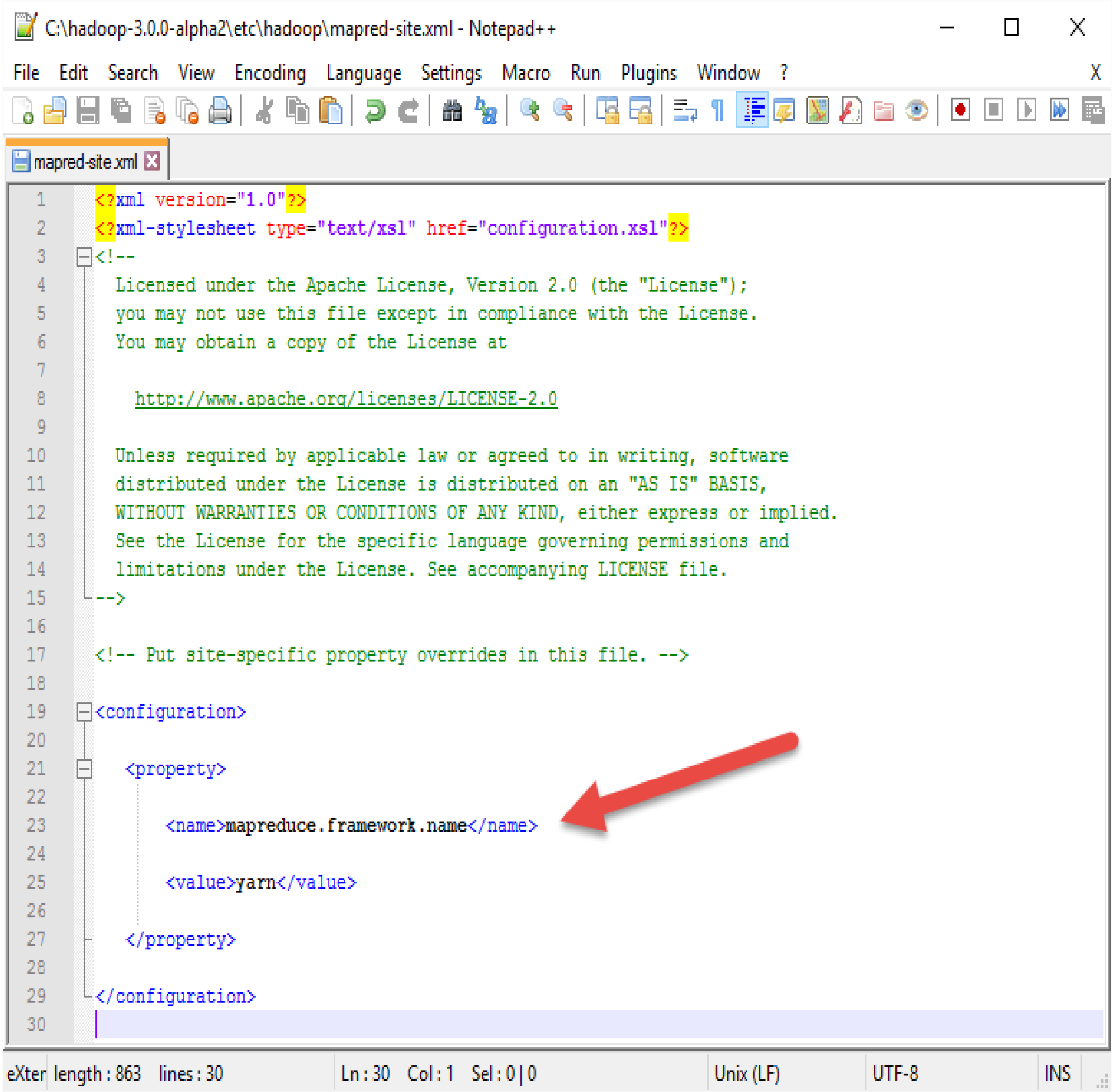
<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

This is what the file looks like when configured:



C:\hadoop-3.0.0-alpha2\etc\hadoop\mapred-site.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
mapred-site.xml
eXter length : 863 lines : 30
Ln : 30 Col : 1 Sel : 0 | 0
Unix (LF)
UTF-8
INS

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

   <property>

       <name>mapreduce.framework.name</name>

       <value>yarn</value>

   </property>

</configuration>

```

The next step was to created a new ‘data’ folder in Hadoop’s home directory (C:\hadoop-3.0.0-alpha2\data).

Once done, the next step was to add a data node and name node to Hadoop, by editing

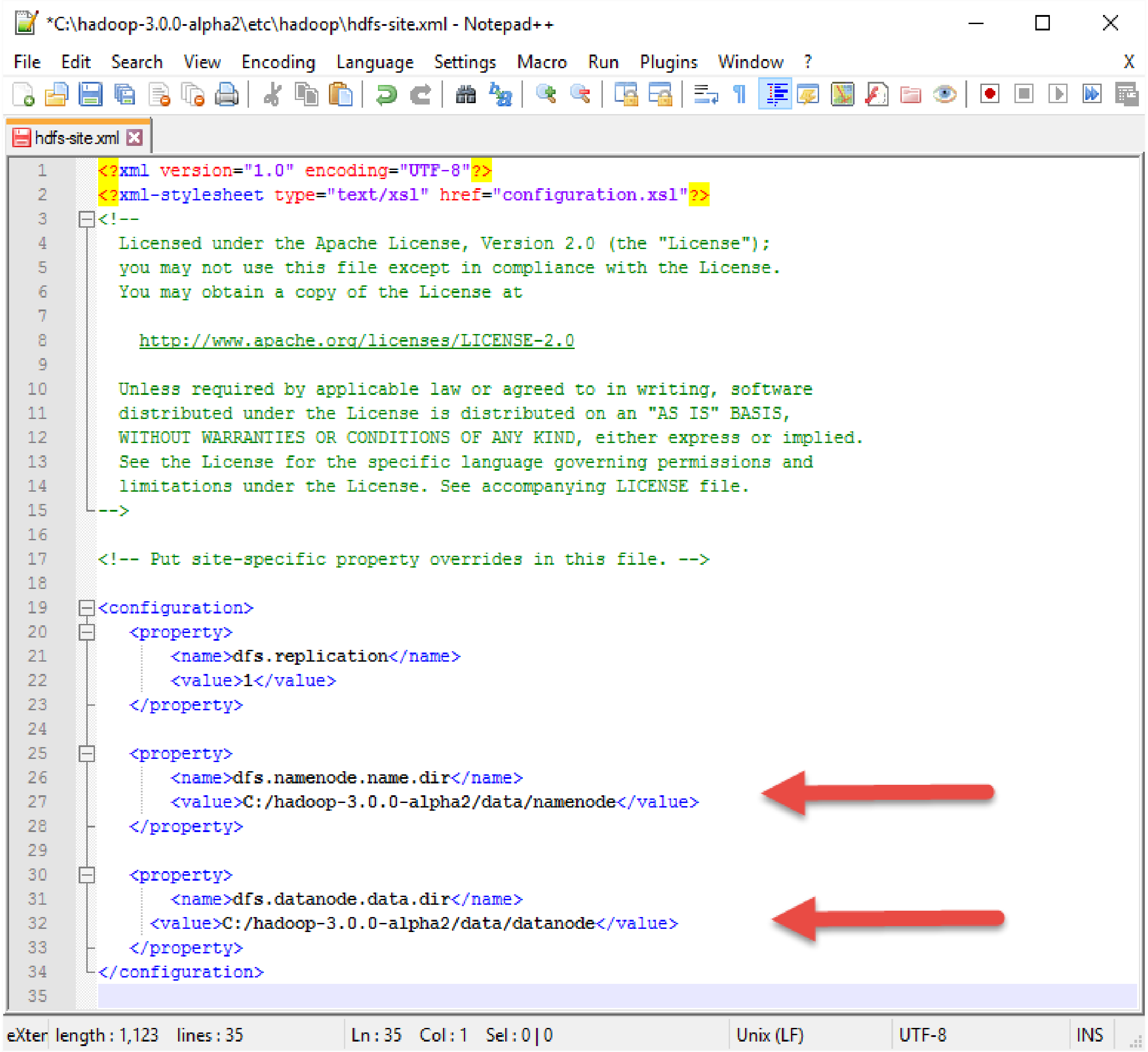
c:\hadoop-3.0.0-alpha2\etc\hadoop\hdfs-site.xml file.

And added following configuration to this XML file:

```
configuration>



<property> <name>dfs.replication</name> <value>1</value> </property> <property>
<name>dfs.namenode.name.dir</name> <value>C:/hadoop-3.0.0-
alpha2/data/namenode</value> </property><property> <name>dfs.datanode.data.dir</name>
<value>C:/hadoop-3.0.0-alpha2/data/datanode</value> </property></configuration>
```

In above step, I had to make sure that I am pointing to location of my newly created data folder and append the datanode and namenode as shown in example.

This is what hdfs-site.xml file looked like once completed:



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
   <property>
       <name>dfs.replication</name>
       <value>1</value>
   </property>

   <property>
       <name>dfs.namenode.name.dir</name>
       <value>C:/hadoop-3.0.0-alpha2/data/namenode</value>
   </property>

   <property>
       <name>dfs.datanode.data.dir</name>
     <value>C:/hadoop-3.0.0-alpha2/data/datanode</value>
   </property>
</configuration>

```

The next step was to add site specific YARN configuration properties by editing yarn-site.xml at

C:\hadoop-3.0.0-alpha2\etc\hadoop\yarn-site.xml, like this:

<configuration> <property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value> </property>

<property>

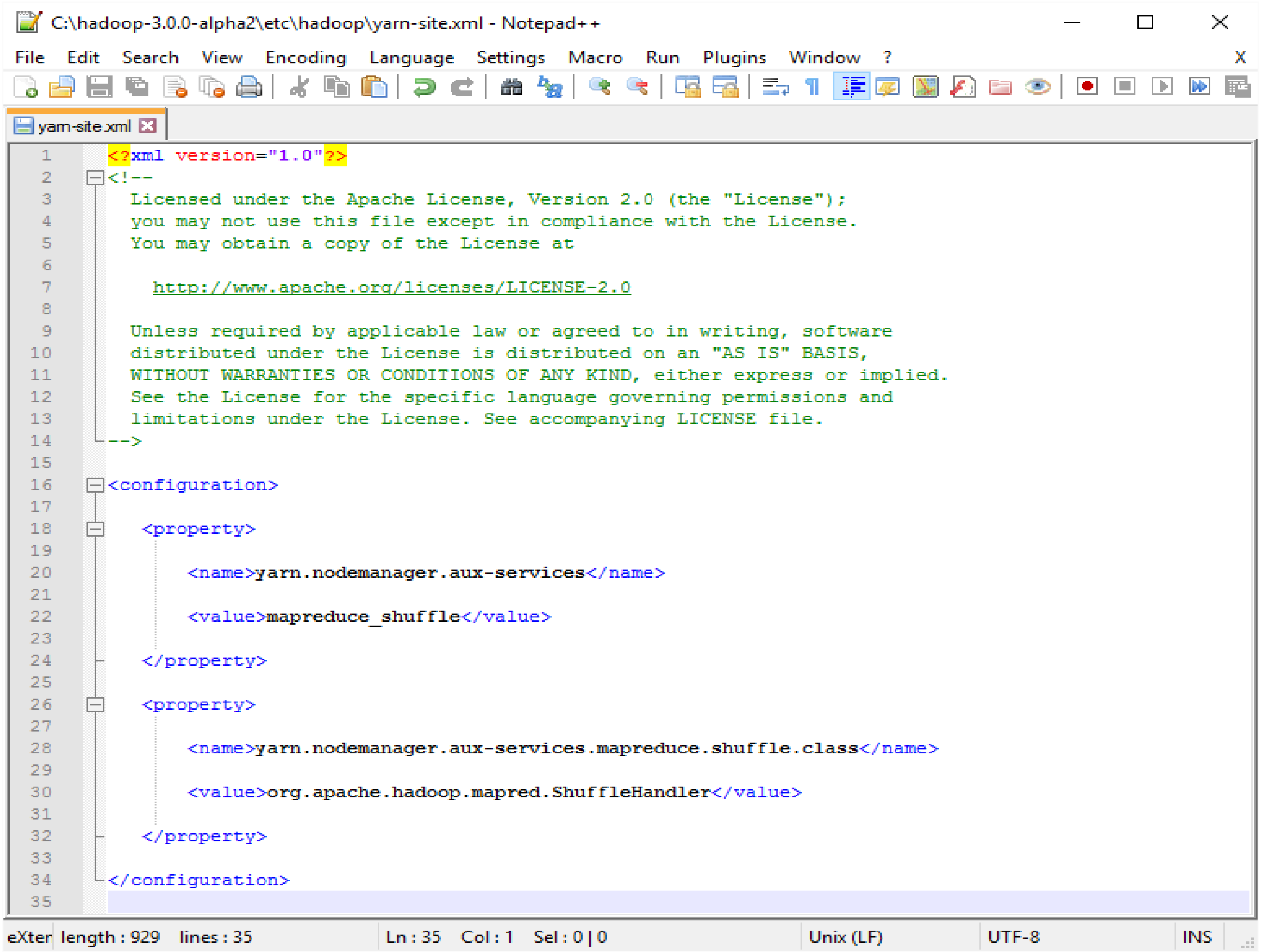
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

This is what yarn-site.xml file looked like once completed:



```xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<configuration>

   <property>

       <name>yarn.nodemanager.aux-services</name>

       <value>mapreduce_shuffle</value>

   </property>

   <property>

       <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

       <value>org.apache.hadoop.mapred.ShuffleHandler</value>

   </property>

</configuration>

```

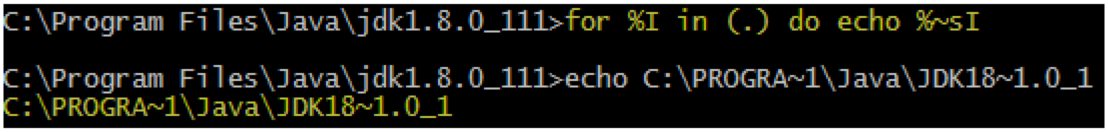
Then I continued by editing hadoop-env.cmd in C:\hadoop-3.0.0-alpha2\etc\hadoop\hadoop-env.cmd. Then

changed the line for JAVA_HOME=%JAVA_HOME% and added a path to my JAVA

folder: C:\PROGRA~1\Java\JDK18~1.0_1

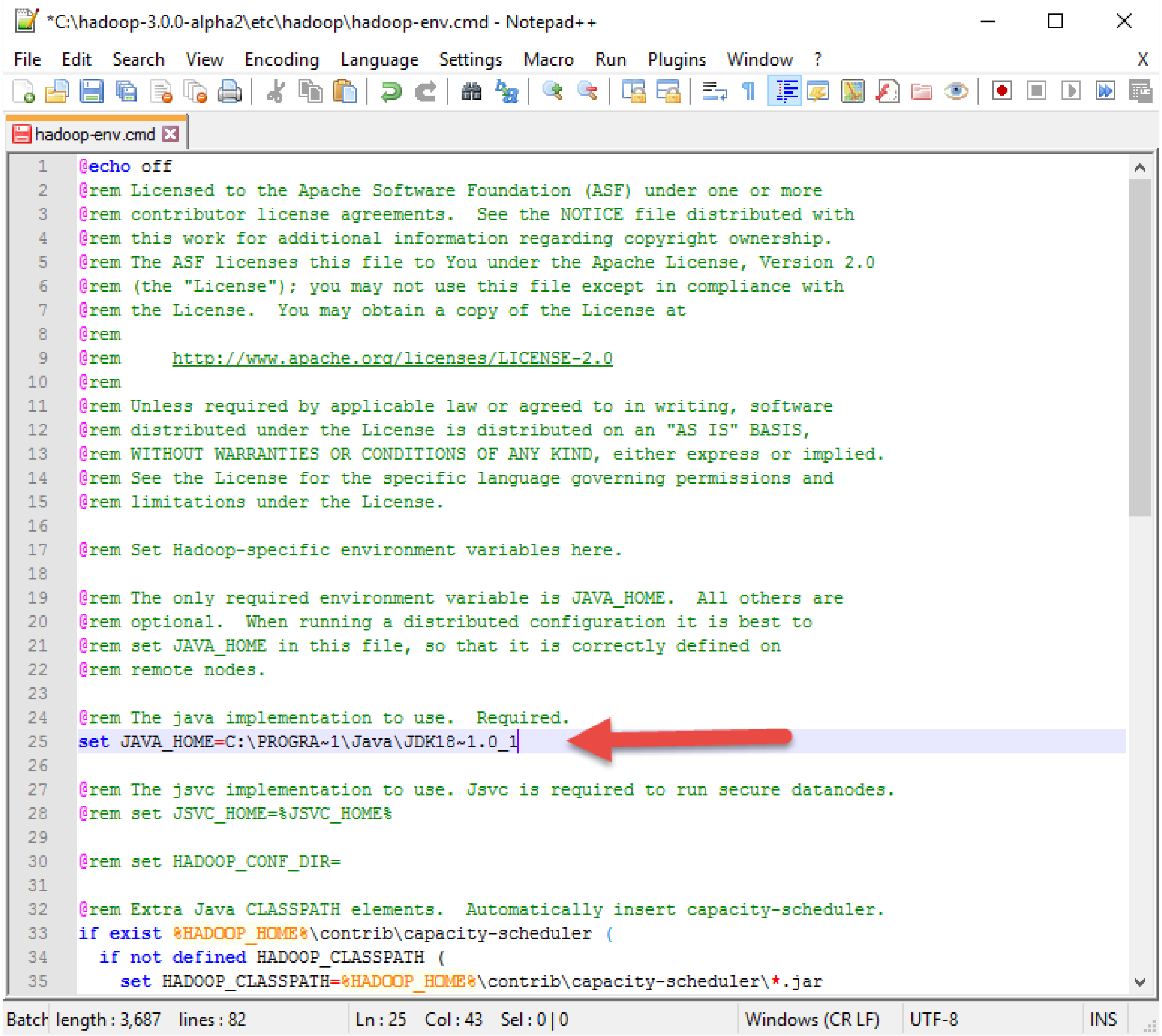
Go to C:\Program Files\Java\jdk1.8.0_111 where my Java JDK is installed and converted a long path to

windows short name:



```
C:\Program Files\Java\jdk1.8.0_111>for %I in (.) do echo %~sI

C:\Program Files\Java\jdk1.8.0_111>echo C:\PROGRA~1\Java\JDK18~1.0_1
C:\PROGRA~1\Java\JDK18~1.0_1
```

Next step was to open hadoop-env.cmd and add it in there, as shown in this screenshot:



*C:\hadoop-3.0.0-alpha2\etc\hadoop\hadoop-env.cmd - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
hadoop-env.cmd

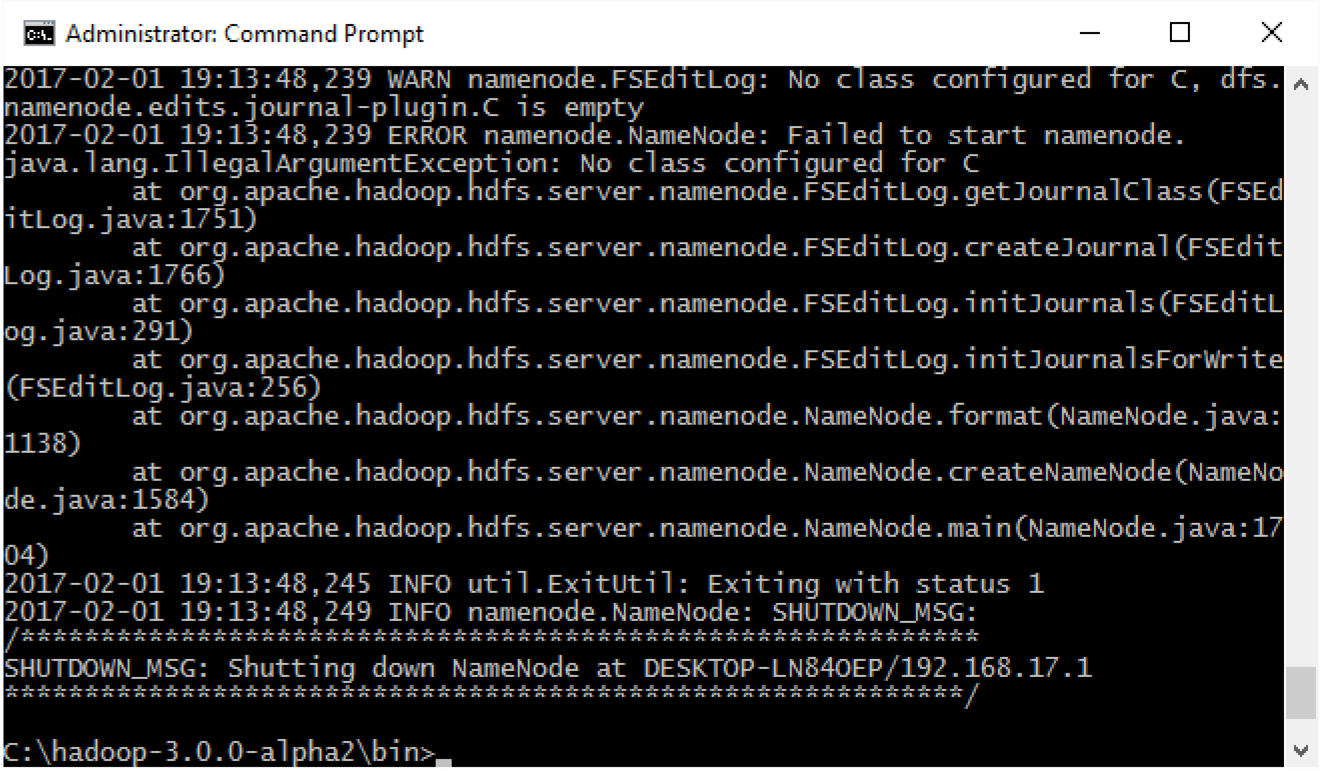
```
@echo off
@rem Licensed to the Apache Software Foundation (ASF) under one or more
@rem contributor license agreements.  See the NOTICE file distributed with
@rem this work for additional information regarding copyright ownership.
@rem The ASF licenses this file to You under the Apache License, Version 2.0
@rem (the "License"); you may not use this file except in compliance with
@rem the License.  You may obtain a copy of the License at
@rem
@rem     http://www.apache.org/licenses/LICENSE-2.0
@rem
@rem Unless required by applicable law or agreed to in writing, software
@rem distributed under the License is distributed on an "AS IS" BASIS,
@rem WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
@rem See the License for the specific language governing permissions and
@rem limitations under the License.

@rem Set Hadoop-specific environment variables here.

@rem The only required environment variable is JAVA_HOME.  All others are
@rem optional.  When running a distributed configuration it is best to
@rem set JAVA_HOME in this file, so that it is correctly defined on
@rem remote nodes.

@rem The java implementation to use.  Required.
set JAVA_HOME=C:\PROGRA~1\Java\JDK18~1.0_1

@rem The jsvc implementation to use. Jsvc is required to run secure datanodes.
@rem set JSVC_HOME=%JSVC_HOME%

@rem set HADOOP_CONF_DIR=

@rem Extra Java CLASSPATH elements.  Automatically insert capacity-scheduler.
if exist %HADOOP_HOME%\contrib\capacity-scheduler (
  if not defined HADOOP_CLASSPATH (
    set HADOOP_CLASSPATH=%HADOOP_HOME%\contrib\capacity-scheduler\*.jar
```

Batch length : 3,687 lines : 82 Ln : 25 Col : 43 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Next in C:\hadoop-3.0.0-alpha2\bin using windows command prompt as admin run:

‘hdfs namenode -format’ command.

Output looked like this:

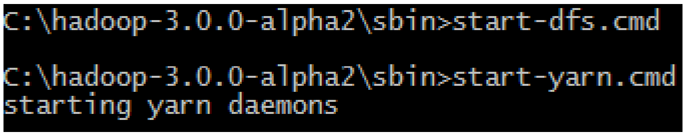


```
2017-02-01 19:13:48,239 WARN namenode.FSEditLog: No class configured for C, dfs.
namenode.edits.journal-plugin.C is empty
2017-02-01 19:13:48,239 ERROR namenode.NameNode: Failed to start namenode.
java.lang.IllegalArgumentException: No class configured for C
        at org.apache.hadoop.hdfs.server.namenode.FSEditLog.getJournalClass(FSEd
itLog.java:1751)
        at org.apache.hadoop.hdfs.server.namenode.FSEditLog.createJournal(FSEdit
Log.java:1766)
        at org.apache.hadoop.hdfs.server.namenode.FSEditLog.initJournals(FSEditL
og.java:291)
        at org.apache.hadoop.hdfs.server.namenode.FSEditLog.initJournalsForWrite
(FSEditLog.java:256)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.format(NameNode.java:
1138)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNo
de.java:1584)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:17
04)
2017-02-01 19:13:48,245 INFO util.ExitUtil: Exiting with status 1
2017-02-01 19:13:48,249 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-LN84OEP/192.168.17.1
************************************************************/

C:\hadoop-3.0.0-alpha2\bin>
```

Then I’ve finally started Hadoop. I’ve opened command prompt as admin in C:\hadoop-3.0.0-alpha2\sbin and

ran

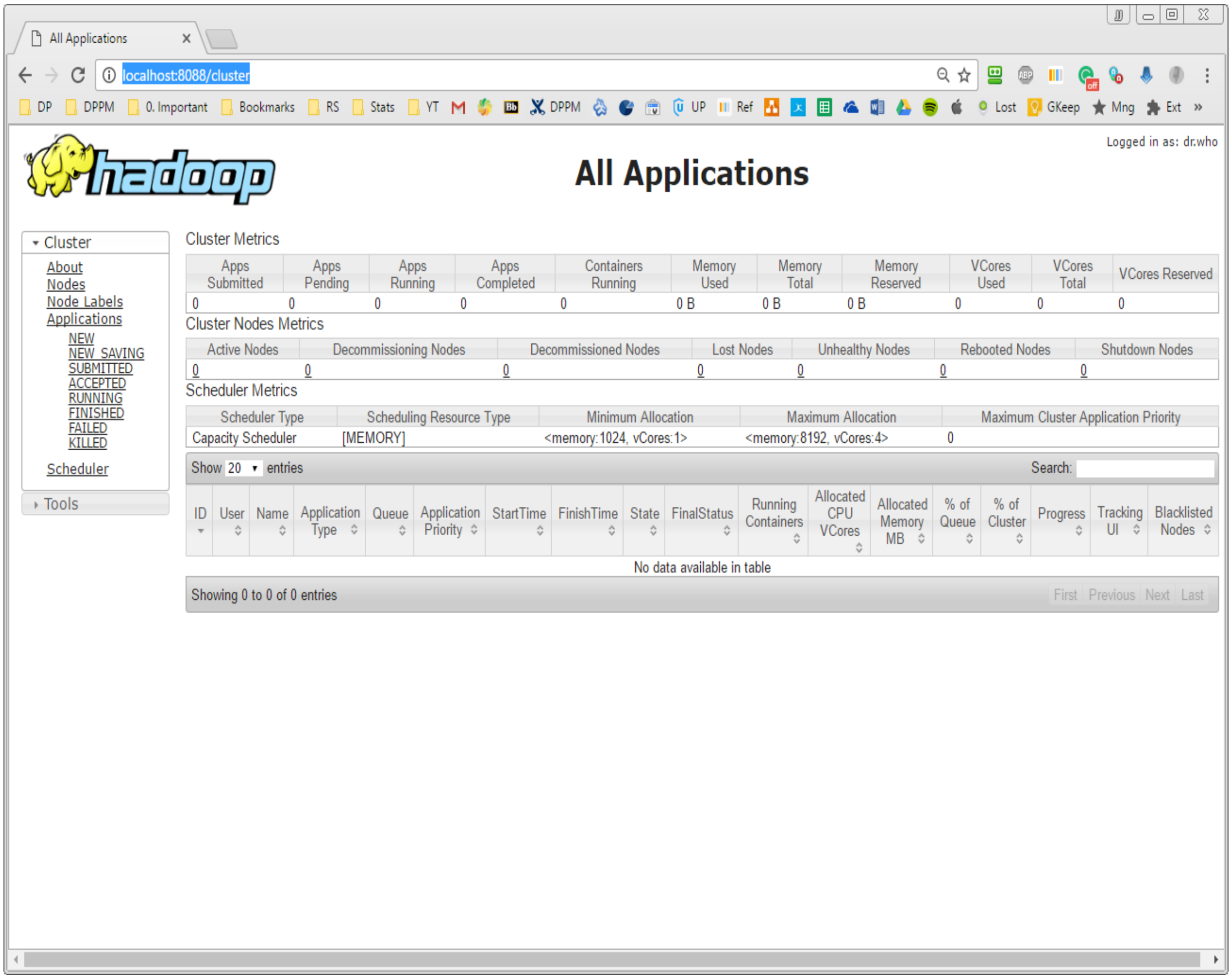
start-dfs.cmd and also **start-yarn.cmd**, like this:



```
C:\hadoop-3.0.0-alpha2\sbin>start-dfs.cmd

C:\hadoop-3.0.0-alpha2\sbin>start-yarn.cmd
starting yarn daemons
```

Open Hadoop GUI

Once all above steps were completed, I’ve opened browser and navigated to: http://localhost:8088/cluster



Logged in as: dr.who

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	0 B	0 B	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
0	0	0	0	0	0	0

Scheduler Metrics

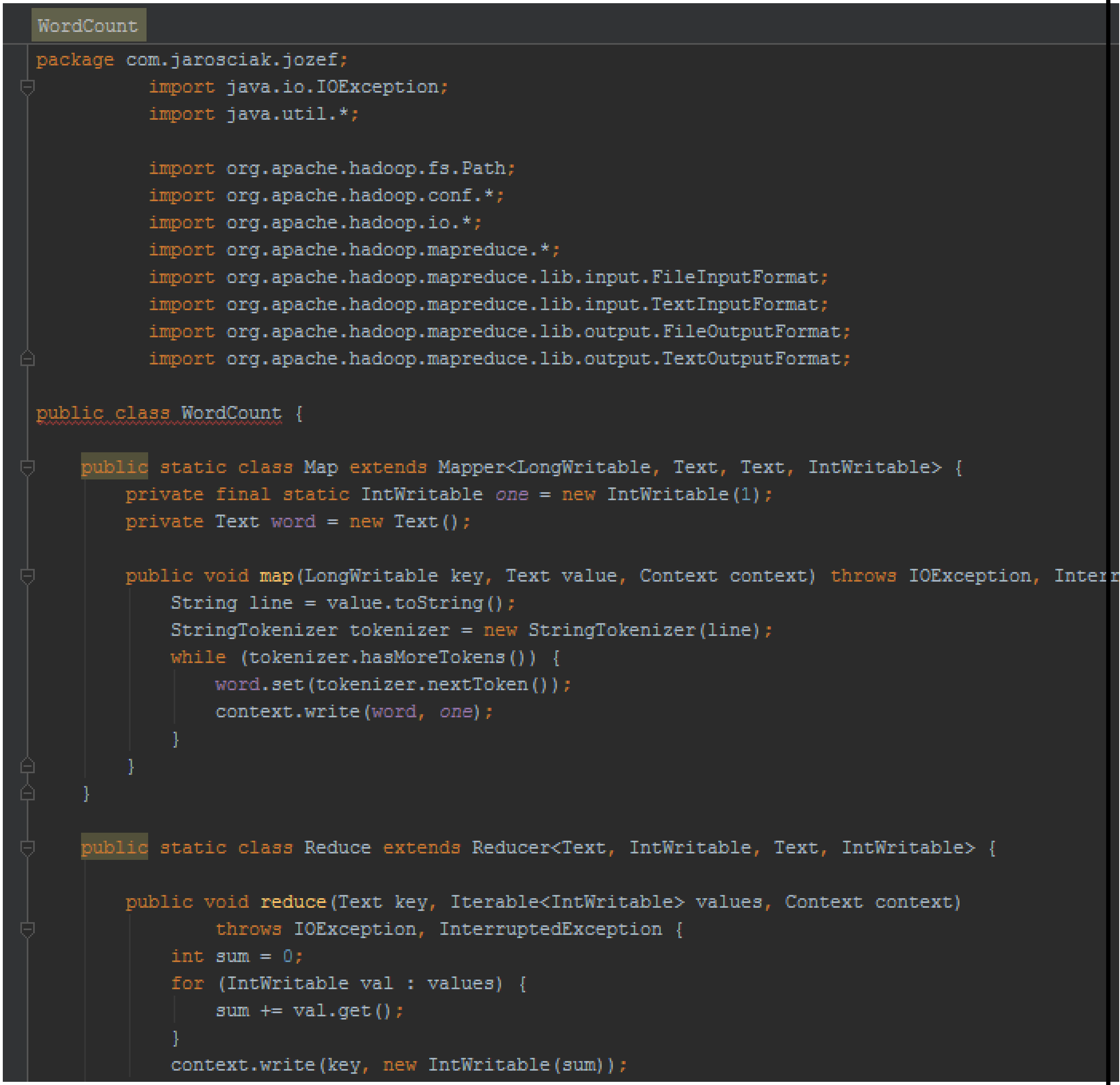
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																	

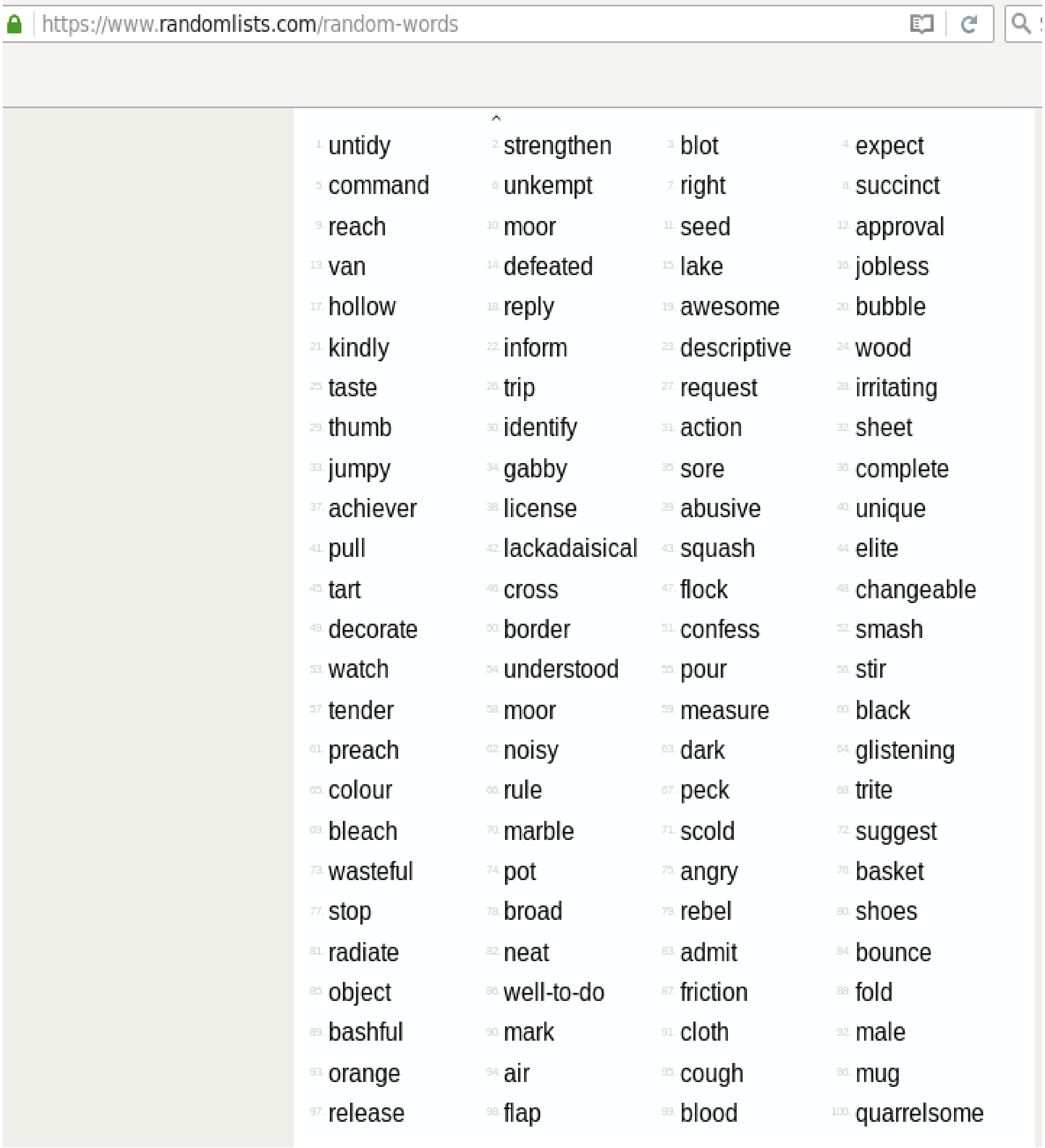
Showing 0 to 0 of 0 entries

WORD COUNT PROGRAM:



```
package com.jarosciak.jozef;
          import java.io.IOException;
          import java.util.*;

          import org.apache.hadoop.fs.Path;
          import org.apache.hadoop.conf.*;
          import org.apache.hadoop.io.*;
          import org.apache.hadoop.mapreduce.*;
          import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
          import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
          import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
          import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {

    public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value, Context context) throws IOException, Interr
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                word.set(tokenizer.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {

        public void reduce(Text key, Iterable<IntWritable> values, Context context)
                throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            context.write(key, new IntWritable(sum));
```

Then go to https://www.randomlists.com/random-words to create couple of random words:



https://www.randomlists.com/random-words

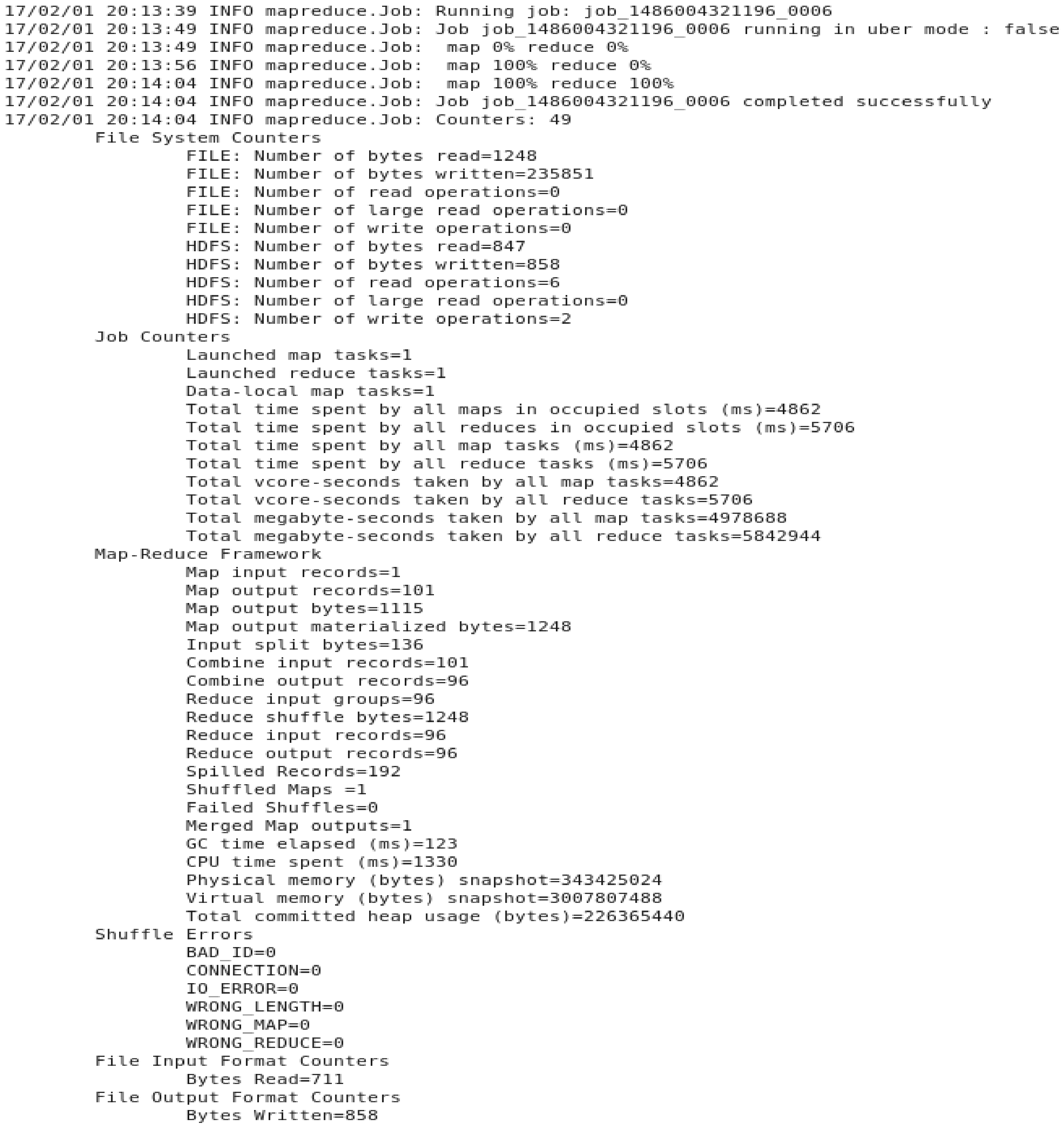
1. untidy
2. strengthen
3. blot
4. expect
5. command
6. unkempt
7. right
8. succinct
9. reach
10. moor
11. seed
12. approval
13. van
14. defeated
15. lake
16. jobless
17. hollow
18. reply
19. awesome
20. bubble
21. kindly
22. inform
23. descriptive
24. wood
25. taste
26. trip
27. request
28. irritating
29. thumb
30. identify
31. action
32. sheet
33. jumpy
34. gabby
35. sore
36. complete
37. achiever
38. license
39. abusive
40. unique
41. pull
42. lackadaisical
43. squash
44. elite
45. tart
46. cross
47. flock
48. changeable
49. decorate
50. border
51. confess
52. smash
53. watch
54. understood
55. pour
56. stir
57. tender
58. moor
59. measure
60. black
61. preach
62. noisy
63. dark
64. glistening
65. colour
66. rule
67. peck
68. trite
69. bleach
70. marble
71. scold
72. suggest
73. wasteful
74. pot
75. angry
76. basket
77. stop
78. broad
79. rebel
80. shoes
81. radiate
82. neat
83. admit
84. bounce
85. object
86. well-to-do
87. friction
88. fold
89. bashful
90. mark
91. cloth
92. male
93. orange
94. air
95. cough
96. mug
97. release
98. flap
99. blood
100. quarrelsome

Then save words to words.txt,

Running Wordlist against Hadoop’s MapReduce

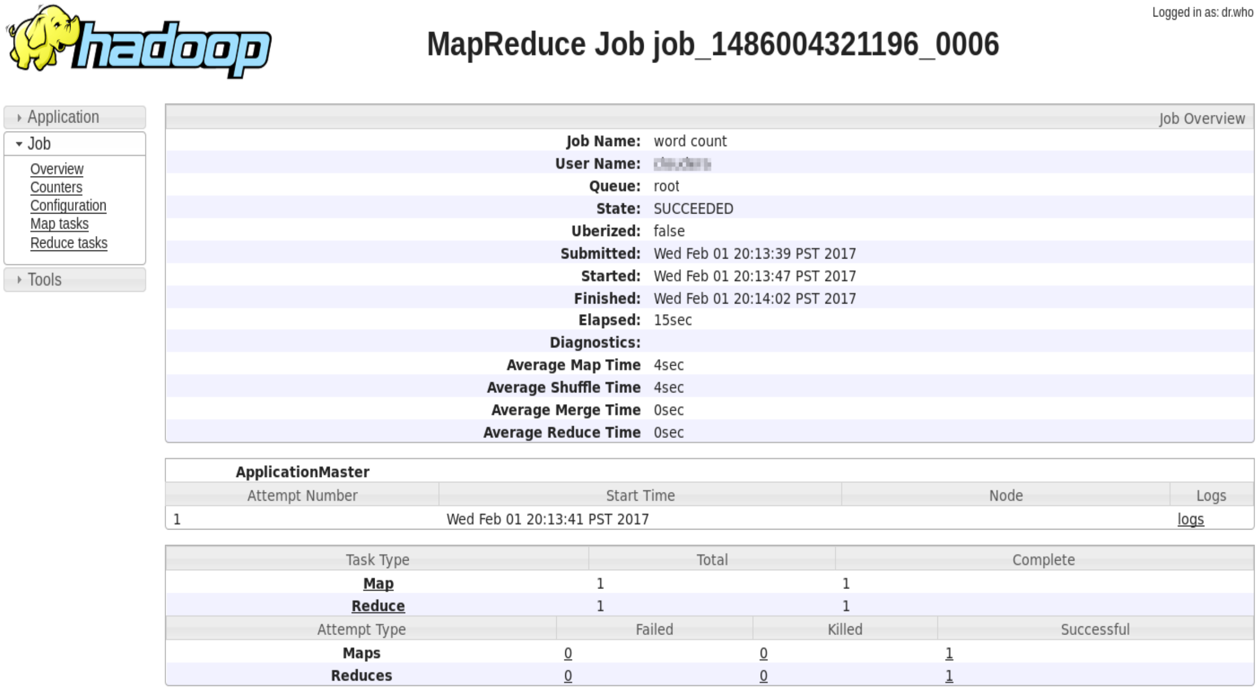
Once I ran my code, it executed and started processing the words.txt file that was prior to execution copied to input folder (which I created earlier together with the output folder for the outcome files).

Following was the result of Hadoop’s processing job:



```
17/02/01 20:13:39 INFO mapreduce.Job: Running job: job_1486004321196_0006
17/02/01 20:13:49 INFO mapreduce.Job: Job job_1486004321196_0006 running in uber mode : false
17/02/01 20:13:49 INFO mapreduce.Job:  map 0% reduce 0%
17/02/01 20:13:56 INFO mapreduce.Job:  map 100% reduce 0%
17/02/01 20:14:04 INFO mapreduce.Job:  map 100% reduce 100%
17/02/01 20:14:04 INFO mapreduce.Job: Job job_1486004321196_0006 completed successfully
17/02/01 20:14:04 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=1248
                FILE: Number of bytes written=235851
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=847
                HDFS: Number of bytes written=858
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=4862
                Total time spent by all reduces in occupied slots (ms)=5706
                Total time spent by all map tasks (ms)=4862
                Total time spent by all reduce tasks (ms)=5706
                Total vcore-seconds taken by all map tasks=4862
                Total vcore-seconds taken by all reduce tasks=5706
                Total megabyte-seconds taken by all map tasks=4978688
                Total megabyte-seconds taken by all reduce tasks=5842944
        Map-Reduce Framework
                Map input records=1
                Map output records=101
                Map output bytes=1115
                Map output materialized bytes=1248
                Input split bytes=136
                Combine input records=101
                Combine output records=96
                Reduce input groups=96
                Reduce shuffle bytes=1248
                Reduce input records=96
                Reduce output records=96
                Spilled Records=192
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=123
                CPU time spent (ms)=1330
                Physical memory (bytes) snapshot=343425024
                Virtual memory (bytes) snapshot=3007807488
                Total committed heap usage (bytes)=226365440
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=711
        File Output Format Counters
                Bytes Written=858
```

We can see the job progress in the browser as well:



MapReduce Job job_1486004321196_0006

	Job Overview
Job Name:	word count
User Name:	[redacted]
Queue:	root
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 01 20:13:39 PST 2017
Started:	Wed Feb 01 20:13:47 PST 2017
Finished:	Wed Feb 01 20:14:02 PST 2017
Elapsed:	15sec
Diagnostics:	
Average Map Time	4sec
Average Shuffle Time	4sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 01 20:13:41 PST 2017		logs

Task Type	Total	Complete	
Map	1	1	
Reduce	1	1	
Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1

OUTPUT:

1. abusive 1
2. achiever 1
3. action 1
4. admit 1
5. air 1
6. angry 1
7. approval 1
8. awesome 1
9. bashful 1
10. basket 1
11. black 1
12. bleach 1
13. blood 1
14. blot 1
15. border 1
16. bounce 1
17. broad 1
18. bubble 1
19. changeable 1
20. cloth 1
21. colour 1
22. command 1

23. confess 1
24. cough 1
25. cross 1
26. dark 1
27. decorate 1
28. defeated 1
29. descriptive 1
30. elite 1
31. expect 1
32. flap 1
33. flock 1
34. fold 1
35. friction 1
36. gabby 1
37. hollow 1
38. identify 1
39. inform 1
40. irritating 1
41. **jarosciak 5**
42. jobless 1
43. jumpy 1
44. kindly 1
45. lackadaisical 1
46. lake 1
47. license 1
48. male 1
49. marble 1
50. mark 1
51. measure 1
52. moor 2
53. mug 1
54. neat 1
55. noisy 1
56. object 1
57. orange 1
58. peck 1
59. pot 1
60. pour 1
61. preach 1
62. pull 1
63. quarrelsome 1
64. radiate 1
65. reach 1
66. rebel 1
67. release 1
68. reply 1
69. request 1
70. right 1
71. scold 1

```
72. seed    1
73. sheet   1
74. shoes   1
75. smash   1
76. sore    1
77. squash  1
78. stir    1
79. stop    1
80. strengthen  1
81. succinct    1
82. suggest 1
83. tart    1
84. taste   1
85. thumb   1
86. trip    1
87. trite   1
88. understood  1
89. unique  1
90. unkempt 1
91. untidy  1
92. van 1
93. wasteful    1
94. watch   1
95. well-to-do  1
96. wood    1
```

RESULT:

Thus a procedure to install single node Hadoop cluster was successfully executed.