NPR
COLLEGE OF ENGINEERING AND TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
NBA Accredited (B.E. - CSE, ECE, EEE & Mechanical Engg.) | Accredited by NAAC with 'A' Grade | Recognized by UGC under 2 (f)
ISO 9001:2015 Certified | Approved by All India Council for Technical Education, New Delhi | Affiliated to Anna University, Chennai
NPR Nagar, Natham, Dindigul - 624 401. | Phone : 04544 - 246500, 501, 502 | Email : nprgi@nprcolleges.org | Web : www.nprcolleges.org

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CCS335 CLOUD COMPUTING LABORATORY

Semester VI

Regulation 2021

Name of the Student : _____

Register Number : _____

Year / Semester : _____

Department : _____

Subject Code : _____

Subject Name : _____

NPR COLLEGE OF ENGINEERING AND TECHNOLOGY

Natham, Dindigul -624 401

Name of the Student: ……

Year: …… Semester: ……Branch: ...……

University Register No.

Certified that this Bonafide Record work done by the above Student in the ……laboratory during the year 20…… - 20……..

Signature of Lab. In-charge ***Signature of Head of the Department***

Submitted for the practical examination held on ……..

Internal Examiner **External Examiner**

NPR COLLEGE OF ENGINEERING AND TECHNOLOGY

VISION

- To develop students with intellectual curiosity and technical expertise to meet the global needs.

MISSION

- To achieve academic excellence by offering quality technical education using best teaching techniques.
- To improve Industry – Institute interactions and expose industrial atmosphere.
- To develop interpersonal skills along with value based education in a dynamic learning environment.
- To explore solutions for real time problems in the society.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Vision

- To develop AI professionals of international relevance to meet the industry and societal needs with future technologies.

Mission

- To collaborate with industry and provide the state-of-the-art infrastructural facilities, with a conducive Teaching-learning ambience.
- To instill in the students the knowledge for world class technical competence, entrepreneurial skill and a spirit of innovation in the area of Artificial Intelligence and Data Science to solve real world problems.
- To encourage students to pursue higher education and research.
- To inculcate right attitude and discipline and develop industry ready professionals for serving the society.

Program Educational Objectives

Graduates will be able to

1. Utilize their proficiencies in the fundamental knowledge of basic sciences, mathematics, Artificial Intelligence, data science, and statistics to build systems that require management and analysis of large volumes of data.
2. Advance their technical skills to pursue pioneering research in the field of AI and Data Science and create disruptive and sustainable solutions for the welfare of ecosystems.
3. Think logically, pursue lifelong learning and collaborate with an ethical attitude in a Multidisciplinary team
4. Design and model AI-based solutions to critical problem domains in the real world.
5. Exhibit innovative thoughts and creative ideas for an effective contribution towards the economy building.

CCS335 CLOUD COMPUTING LABORATORY L T P C
2 0 2 3

Course Objectives

- To understand the principles of cloud architecture, models and infrastructure.
- To understand the concepts of virtualization and virtual machines.
- To gain knowledge about virtualization Infrastructure.
- To explore and experiment with various Cloud deployment environments.
- To learn about the security issues in the cloud environment.

List of Experiments

1. Install Virtual box/VMware Workstation with different flavors of Linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and executes Simple Programs.
3. Install Google App Engine. Create hello world app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Install Hadoop single node cluster and run simple applications like word count.
8. Creating and Executing Your First Container Using Docker.
9. Run a Container from Docker Hub.
10. Cloud for HPC and HTC.
11. Launching of Virtual Machine using Try Stack.

Course Outcomes

The students will be able to

COs	Course Code	Course Outcomes	Knowledge level
CO1	C305.1	Understand the design challenges in the cloud.	K2
CO2	C305.2	Apply the concept of virtualization and its types.	K3

CO3	C305.3	Experiment with virtualization of hardware resources and Docker.	K3
CO4	C305.4	Develop and deploy services on the cloud and set up a cloud environment.	K3
CO5	C305.5	Explain security challenges in the cloud environment.	K2

List of Experiments with COs, POs & PSOs

Exp. No.	Name of the Experiment	COs	POs	PSOs
1.	Install Virtual box/VMware Workstation with different flavors of Linux or windows OS on top of windows 7 or 8.	CO2	1, 2, 3, 4, 5, 9, 10,11 & 12	1, 2 & 3
2.	Install a C compiler in the virtual machine created using virtual box and executes Simple Programs.	CO2	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
3.	Install Google App Engine. Create hello world app and other simple web applications using python/java.	CO4	1, 2, 3, 4, 5, 9, 10,11 & 12	1, 2 & 3
4.	Use GAE launcher to launch the web applications.	CO4	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
5.	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim	CO4	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
6.	Find a procedure to transfer the files from one virtual machine to another virtual machine.	CO3	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
7.	Install Hadoop single node cluster and run simple applications like word count.	CO3	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
8.	Creating and Executing Your First Container Using Docker.	CO3	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
9.	Run a Container from Docker Hub.	CO3	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
Additional Programs				
10.	Cloud for HPC and HTC.	CO1	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
11.	Launching of Virtual Machine using Try Stack.	CO2	1, 2, 3, 4, 5, 9, 10, 11 & 12	1, 2 & 3
				Total: 30 Periods

Program Outcomes

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Conduct Investigations of Complex Problems
5. Modern Tool Usage
6. The Engineer and Society
7. Environment and Sustainability
8. Ethics
9. Individual and Team Work
10. Communication
11. Project Management and Finance
12. Life-long Learning

Program Specific Outcomes

At the end of the program students will be able to

- Develop and implement AI-based processes for effective decision-making in diverse domains, including business and governance, by integrating domain-specific knowledge and advanced techniques.
- Utilize data analysis to derive actionable insights and foresights, enabling the solution of complex business and engineering problems.
- Apply theoretical knowledge of AI and Data Analytics, along with practical tools and techniques, to address societal problems, demonstrating proficiency in data analytics,visualization,and project coordination skills.

CONTENTS

S. No	Date of Experiment	Name of the Experiment	Page No.	Date of Submission	Signature
1.					
2.					
3.					
4.					
5.					
6.					
7.					
8.					
9.					
10.					
11.					

Ex. No: 01 Date:	INSTALL VIRTUALBOX/VMWARE WORKSTATION

Aim:

To Install Virtual box/VMware Workstation with different flavors of Linux or windows OS on top of windows7 or 8.

Procedure to Install

Step 1- Download Link

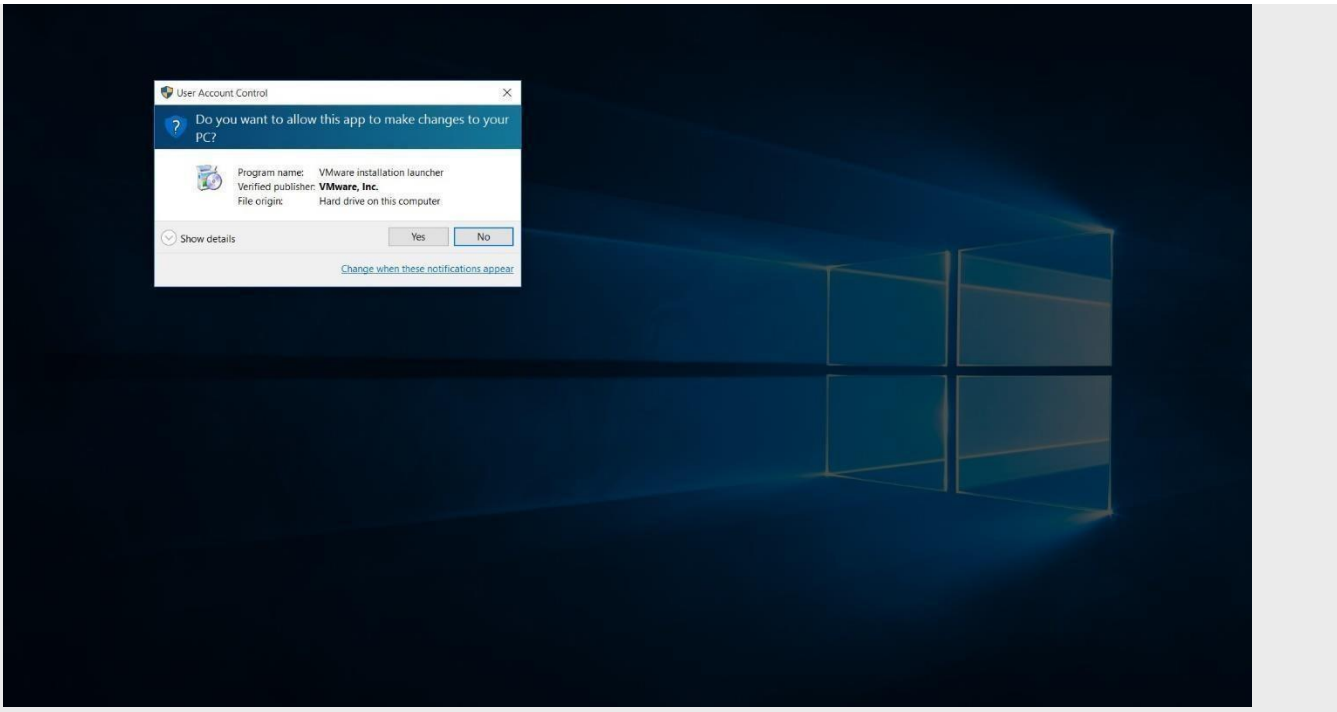
Link for downloading the software is https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html. Download the software for windows. Good thing is that there is no signup process. Click and download begins. Software is around 541 MB.

Step 2- Download the installer file It should probably be in the download folder by default, if you have not changed the settings in your browser. File name should be something like VMware-workstation-full-15.5.1-15018445.exe. This file name can change depending on the version of the software currently available for download. But for now, till the next version is available, they will all be VMware Workstation 15 Pro.

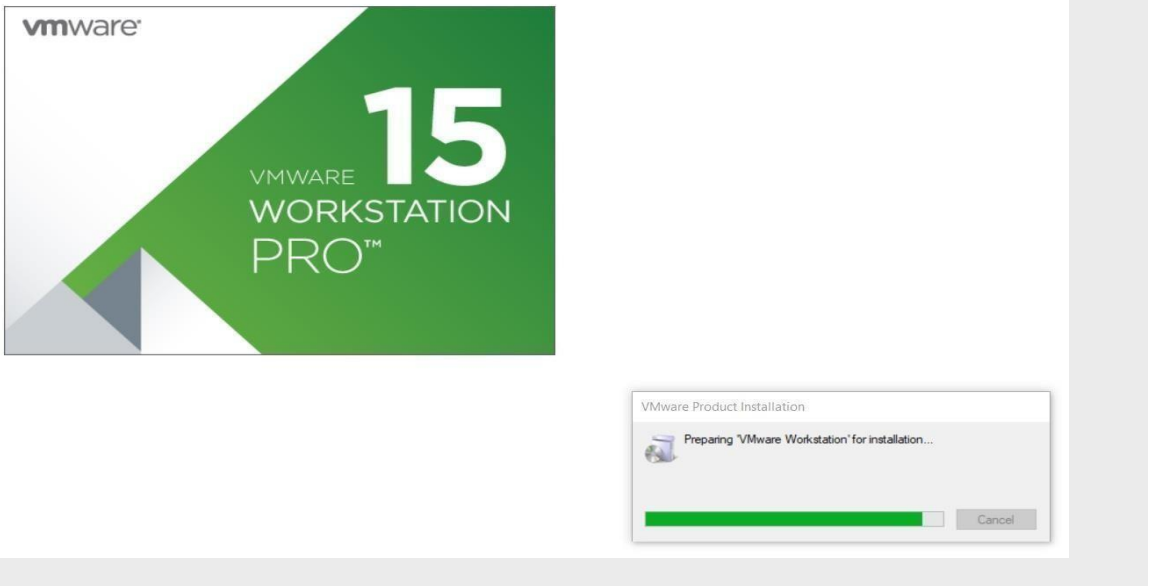
Step 3- Locate the downloaded installer file for demonstration purpose, I have placed the downloaded installer on my desktop. Find the installer on your system and double click to launch the application.

Step 4- User Access Control (UAC) Warning

Now you should see User Access Control (UAC) dialog box. Click yes to continue



Initial Splash screen will appear. Wait for the process to complete.

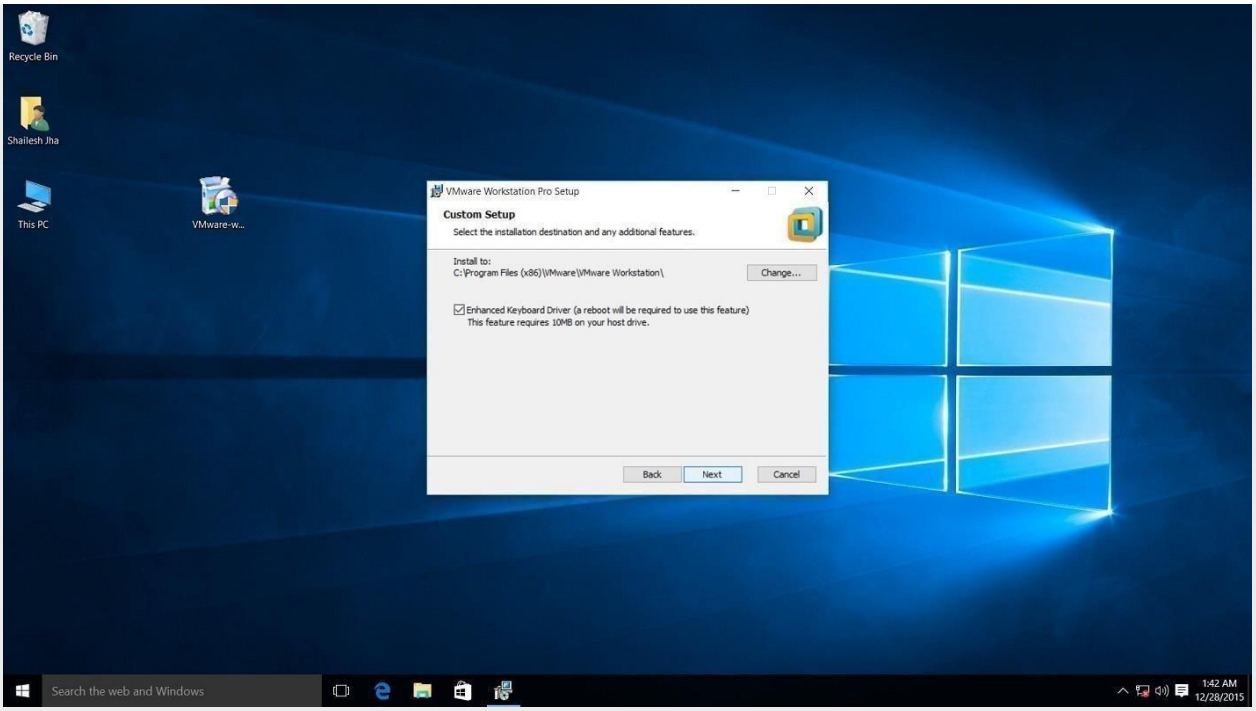


Step 5- VMware Workstation Setup wizard
Now you will see VMware Workstation setup wizard dialog box. Click next to continue.

Step 6- End User License Agreement
This time you should see End User License Agreement dialog box. Check “I accept the terms in theLicense Agreement” box and press next to continue.

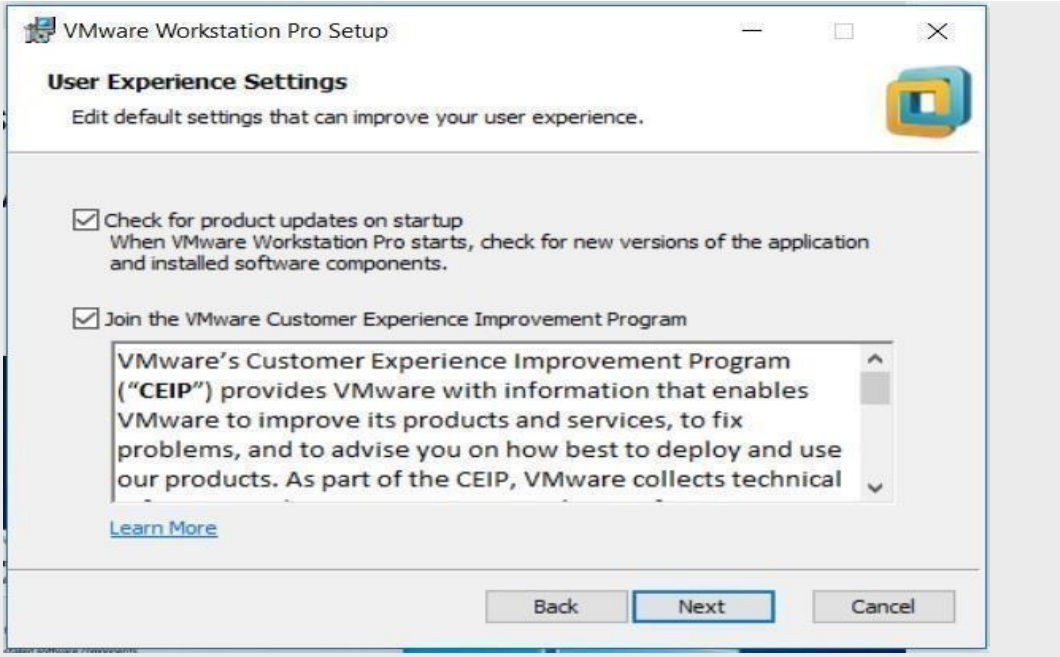
Step 7- Custom Setup options

Select the folder in which you would like to install the application. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.



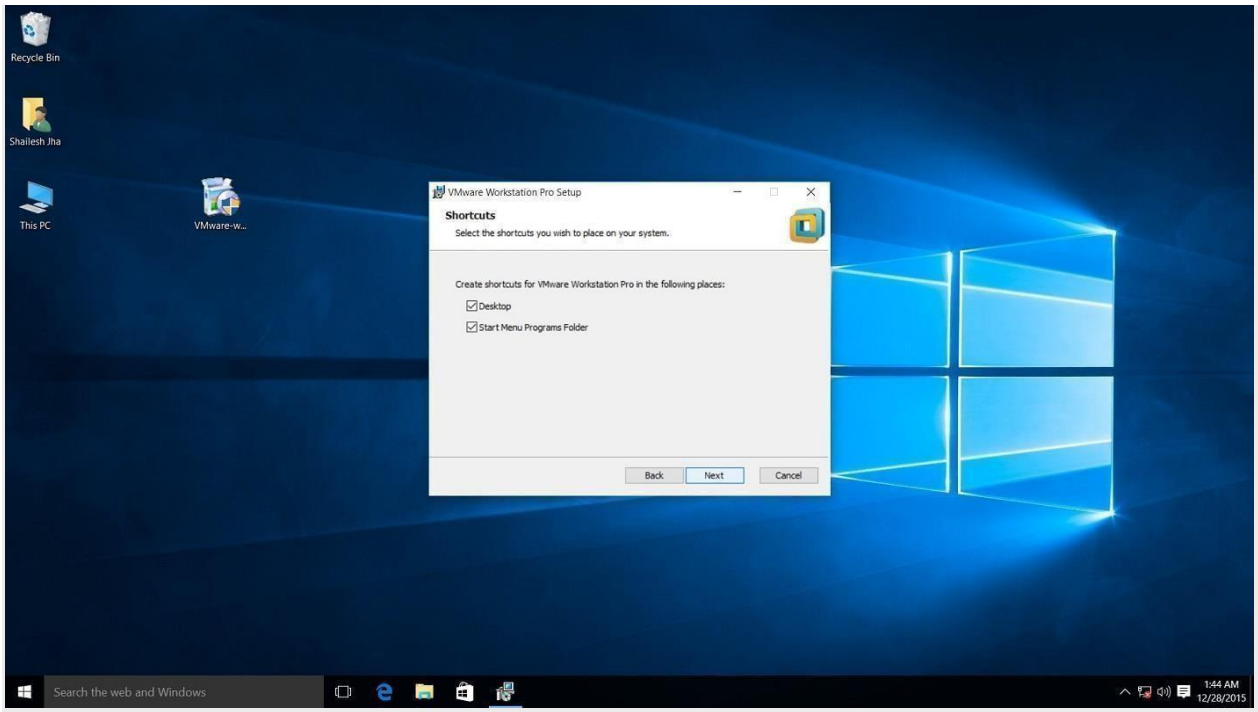
Step 8- User Experience Settings

Next you are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”. Do as you wish. I normally leave it to defaults that are unchecked.



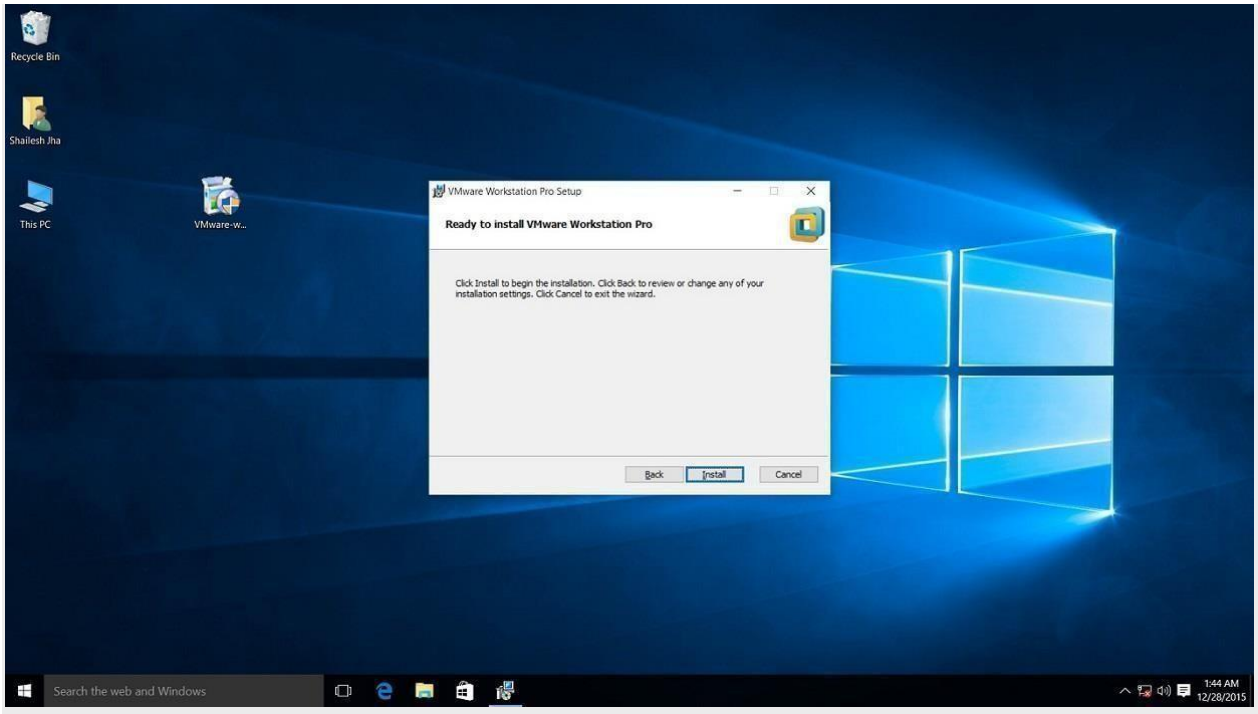
Step 9- Application Shortcuts preference

Next step is to select the place you want the shortcut icons to be placed on your system to launch the application. Please select both the options, desktop and start menu and click next.

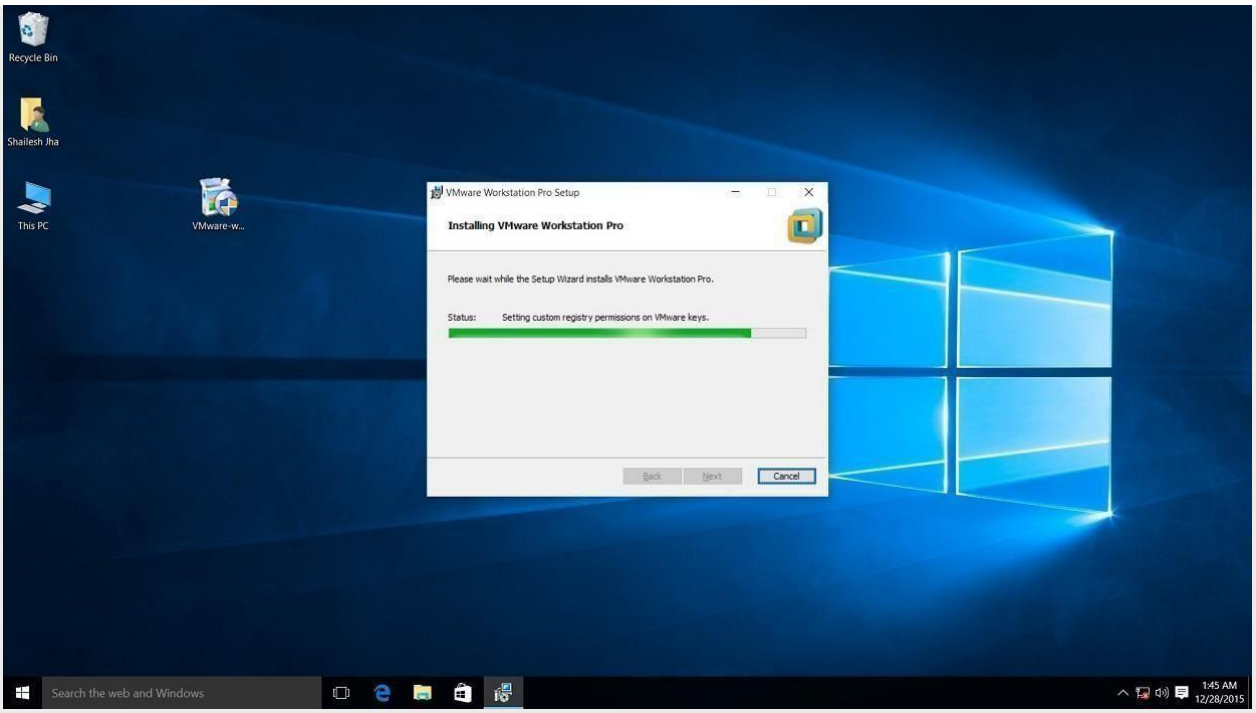


Step 10- Installation begins

Now you see the begin installation dialog box. Click install to start the installation process.



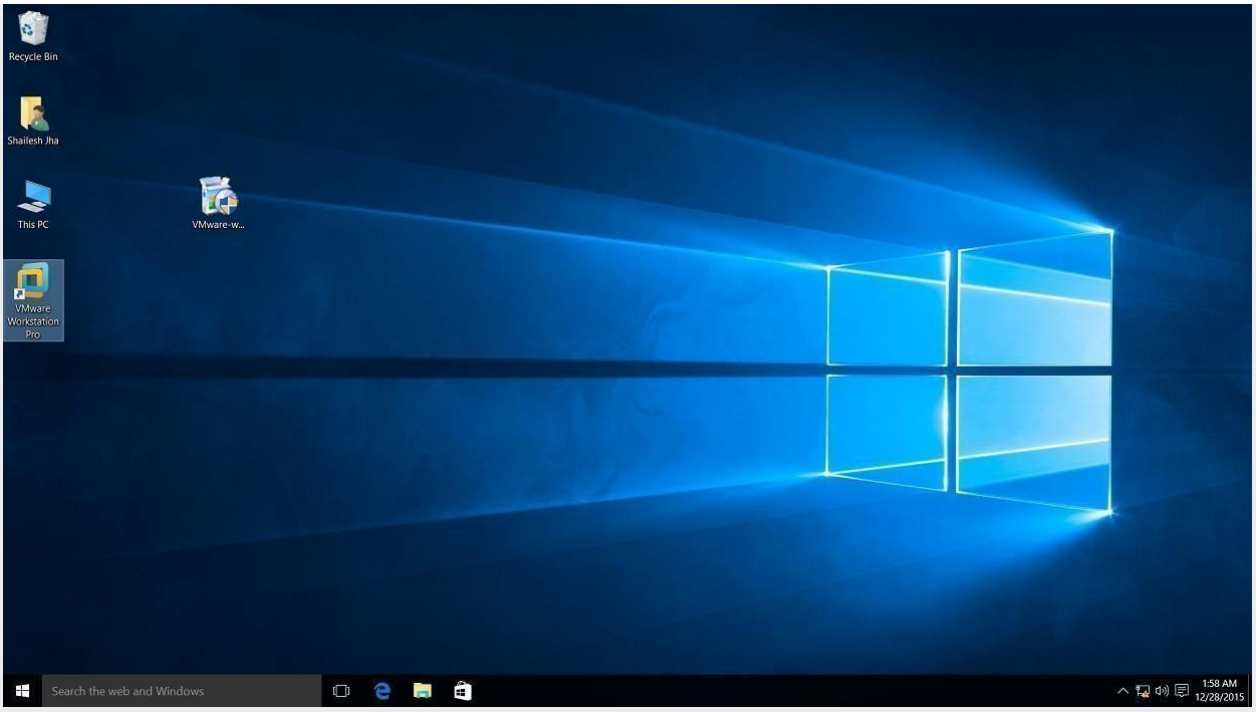
Below screenshot shows Installation in progress. Wait for this to complete



At the end you will see installation complete dialog box. Click finish and you are done with the installation process. You may be asked to restart your computer. Click on Yes to restart.

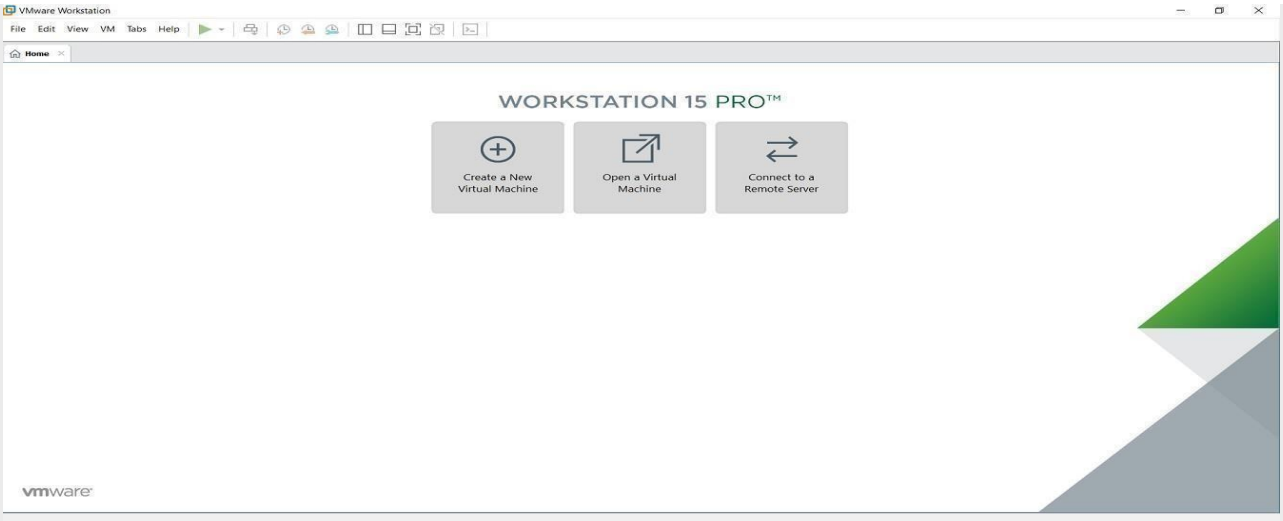
Step 11- Launch VMware Workstation

After the installation completes, you should see VMware Workstation icon on the desktop. Double clickon it to launch the application.



Step 12- License Key

If you see the dialog box asking for license key, click on trial or enter the license key. Then what you have is the VMware Workstation 15 Pro running on your windows 10 desktop. If don‟t have the licensekey, you will have 30 days trial



Step 13- At some point if you decide to buy

At some point of time if you decide to buy the License key, you can enter the License key by goingto **Help->Enter a License Key**You can enter the 25 character license key in the dialog box shown below and click OK. Now you havethe license version of the software.

Result:

Thus, the virtual machine is created and installed successfully.

Ex. No: 02 Date:	INSTALL C COMPILER

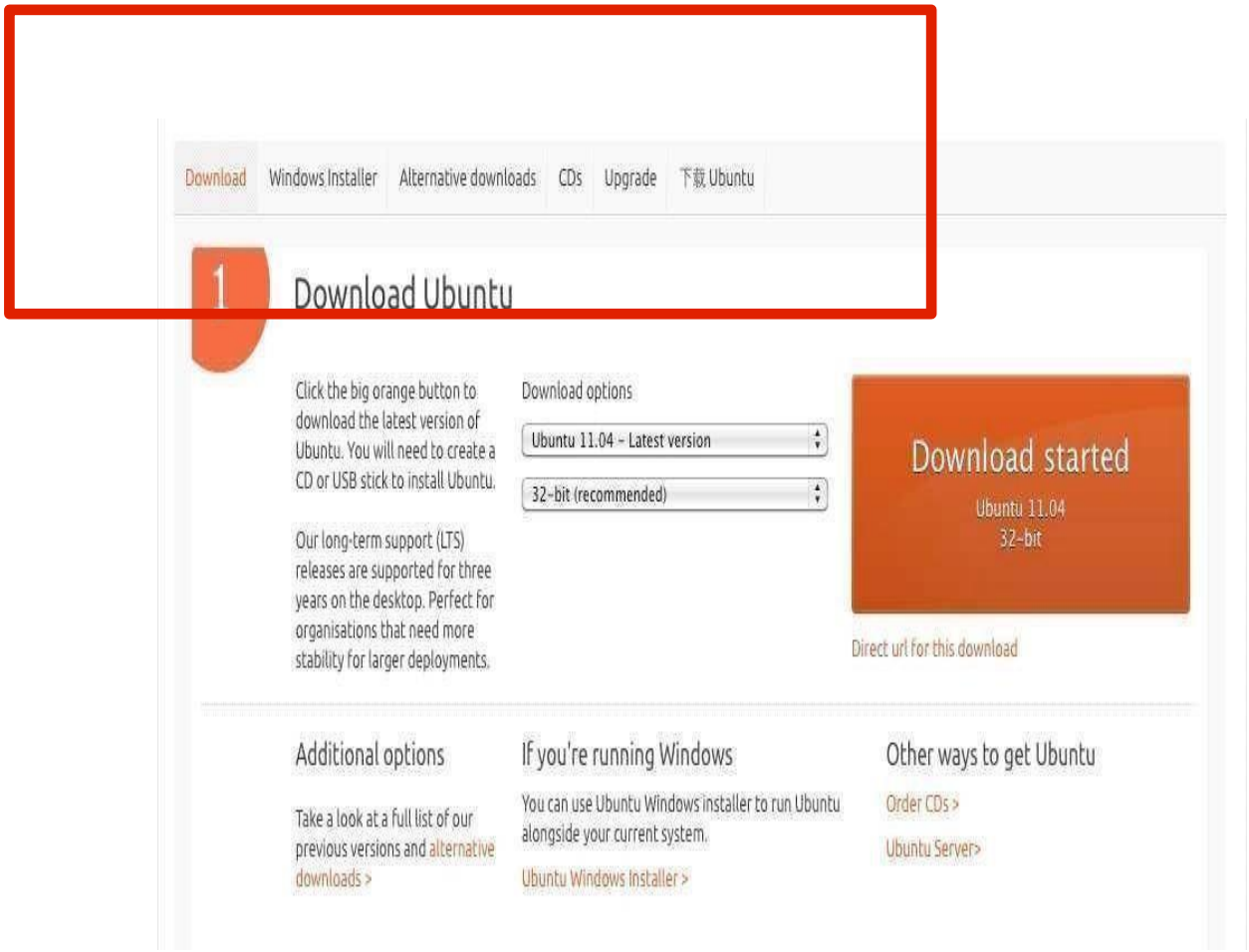
Aim:

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

Procedure:

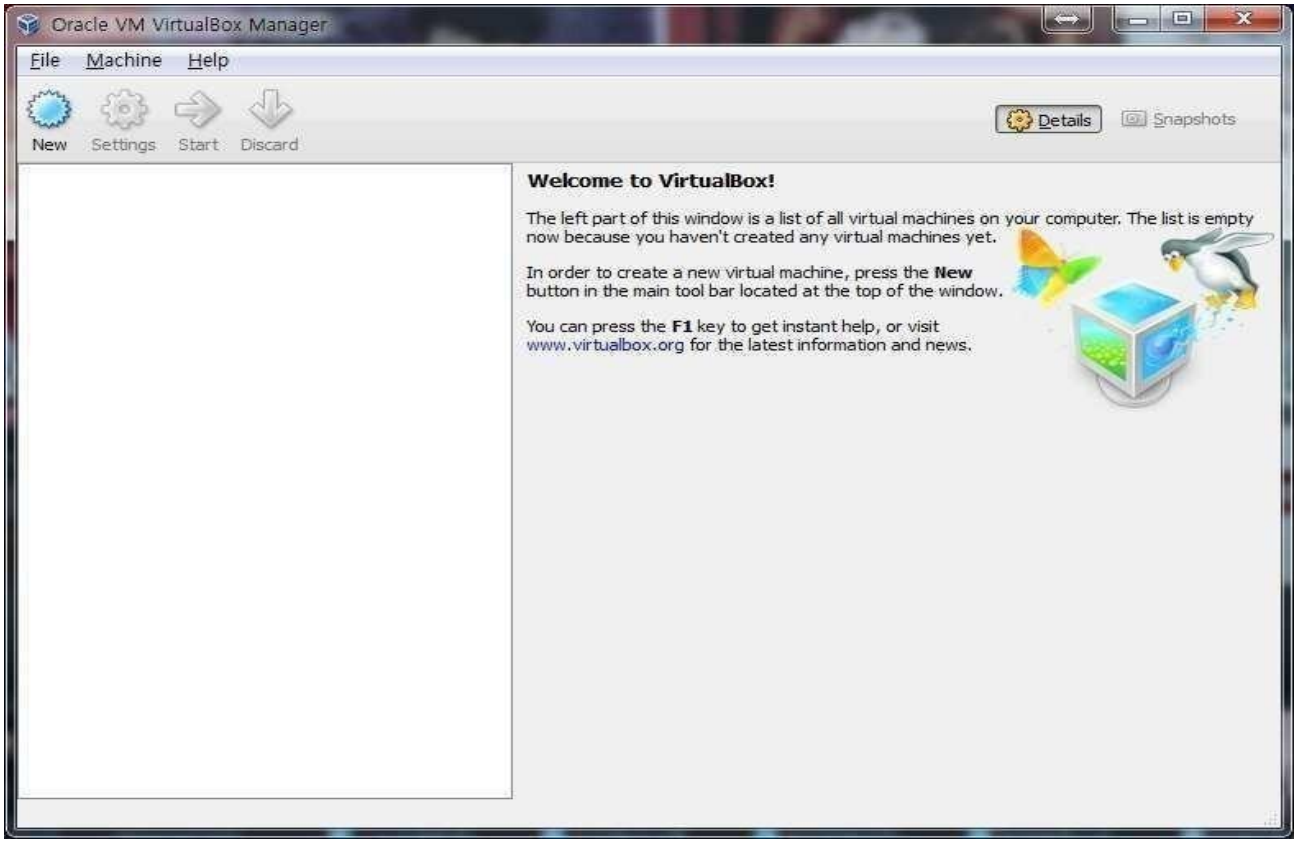
step1:

Choose the Latest version of Ubuntu and 32-bit and click “Start Download”



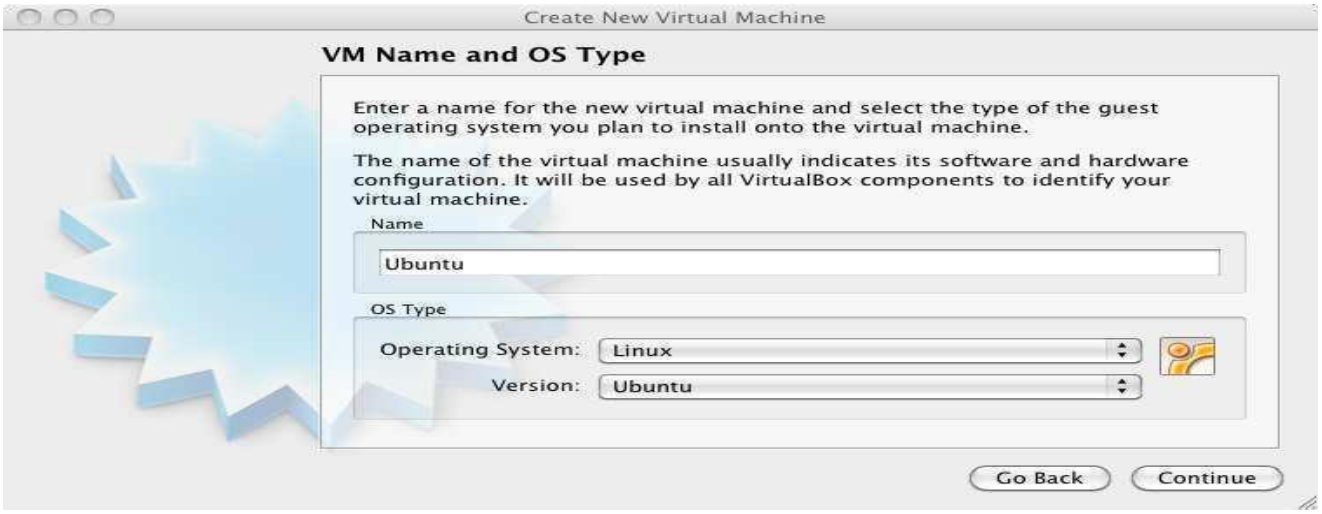
Step 2:

Login into the VM of installed OS.

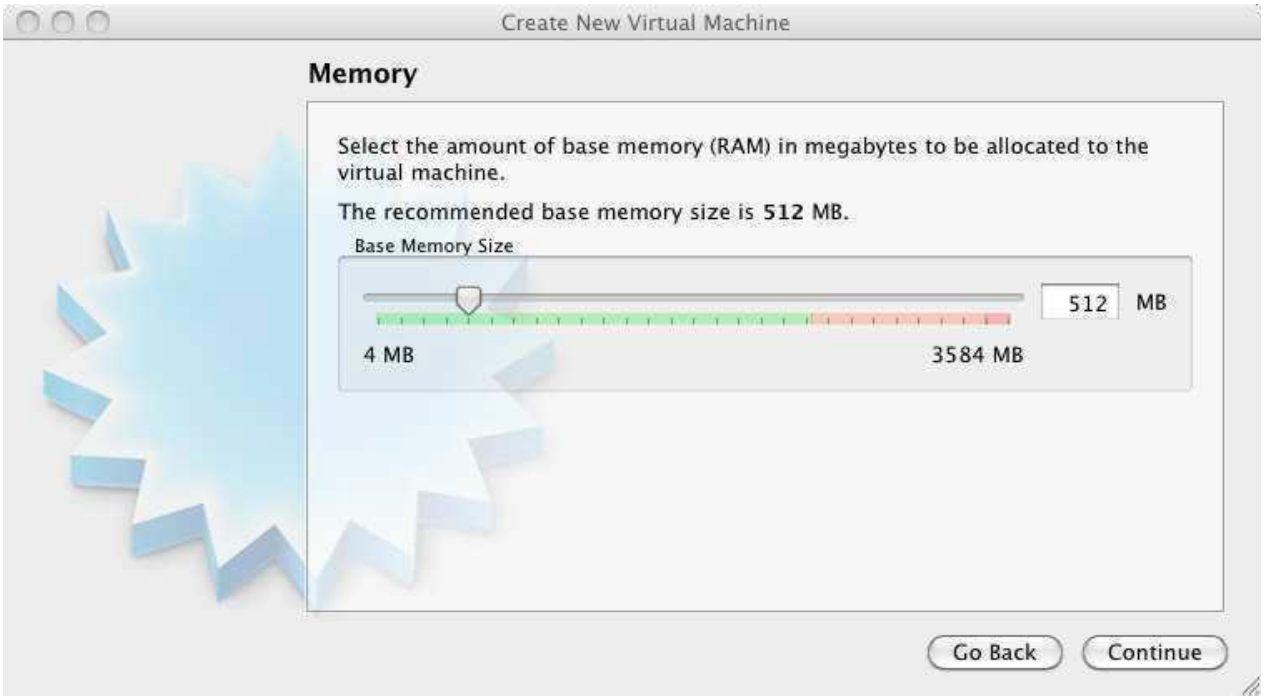


Step 3:

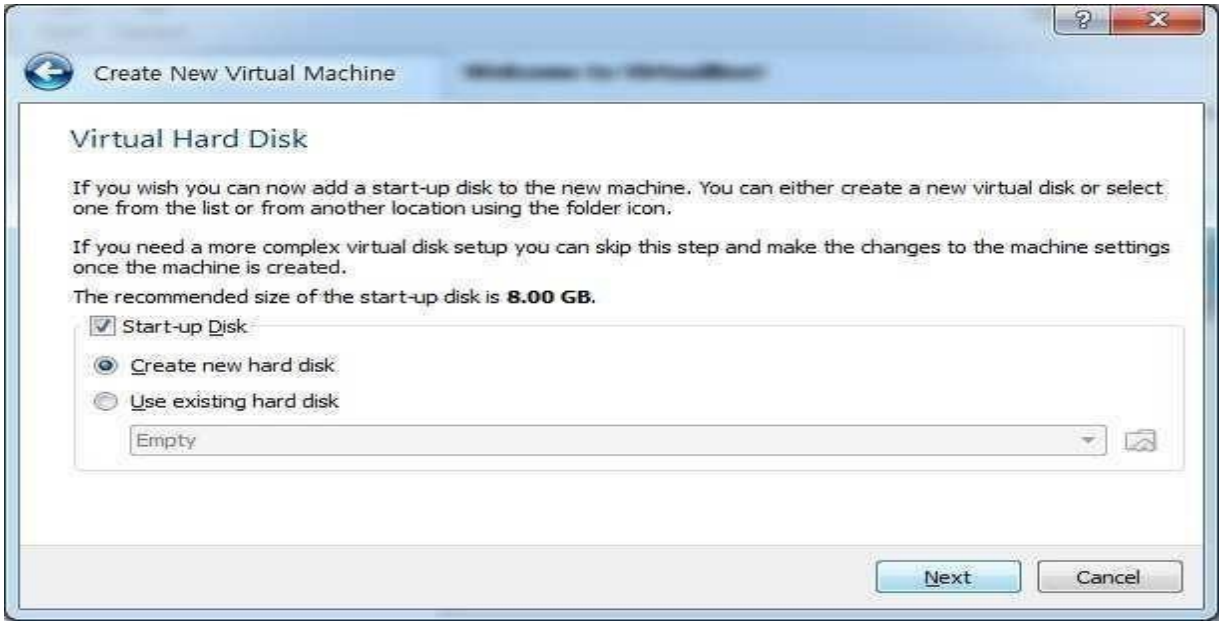
Click“Continue” on the pop-up window
TypeVM name,select“Linux” for the OS and choose“Ubuntu” for the version.



Choose the amount of memory to allocate (I suggest choosing between 5I2 MB to I024 MB) Click Continue or Next



Choose create a new virtual hard disk
Click Continue or Next

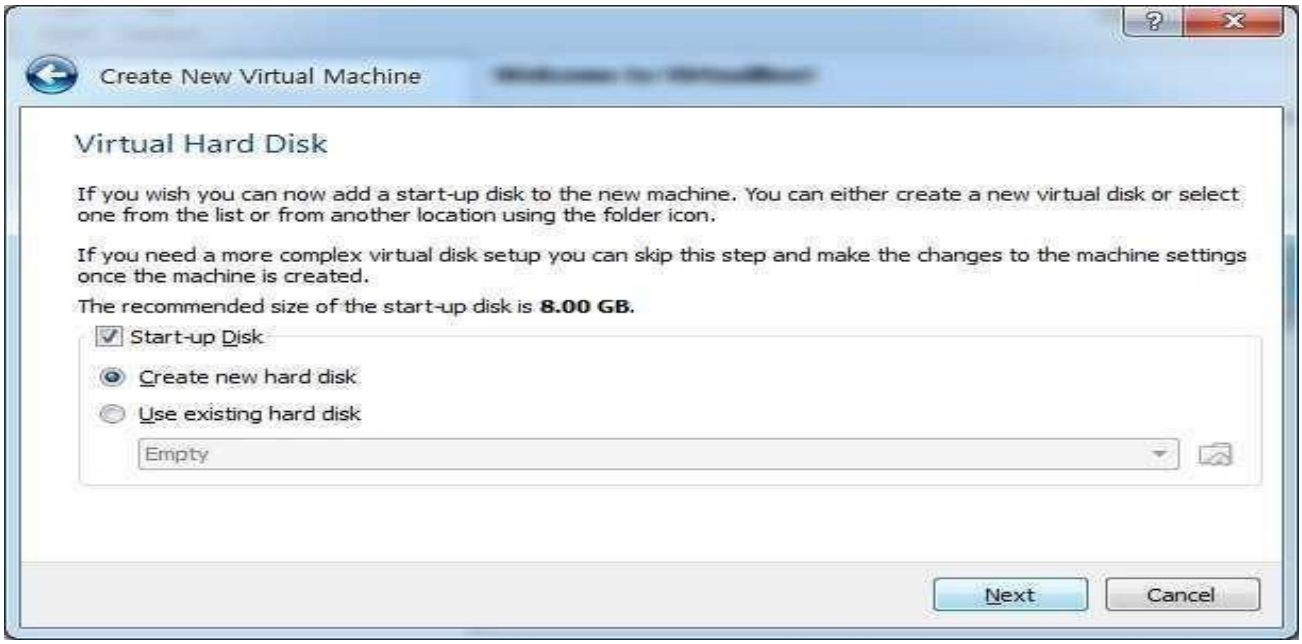


Choose VDI (VirtualBox Disk Image)

Click Continue or Next

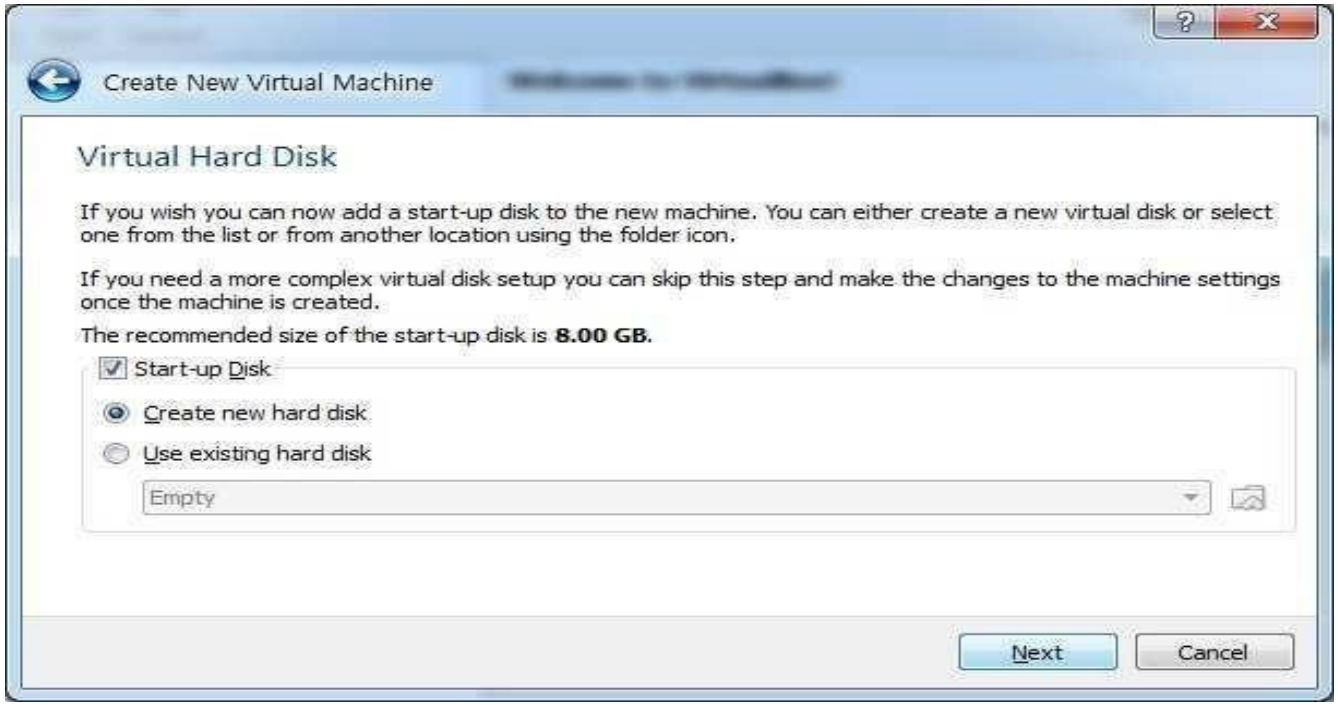
Choose “Dynamically Allocated” click continue.

This way, the size of your Virtual Hard Disk will grow as you use.

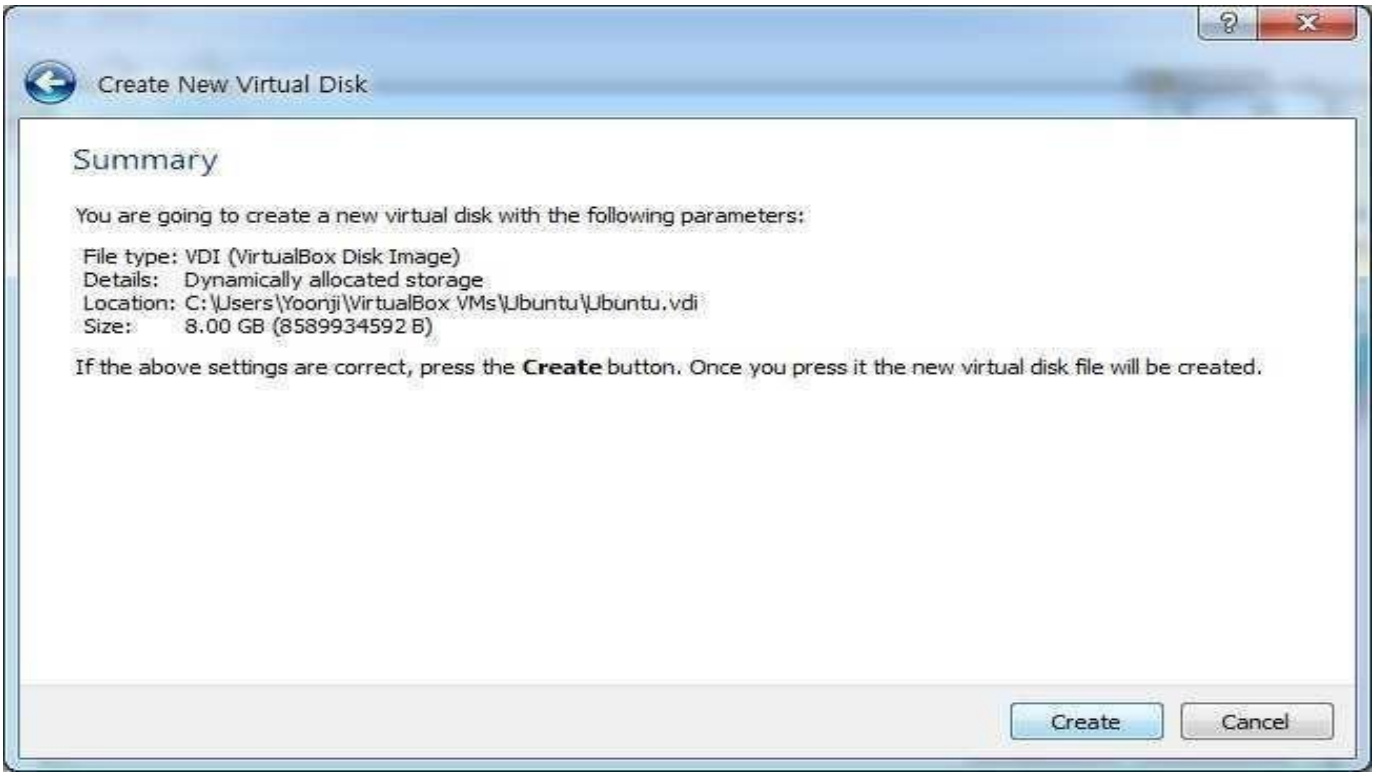


Click the folder icon and choose the ubuntu iso file you downloaded.

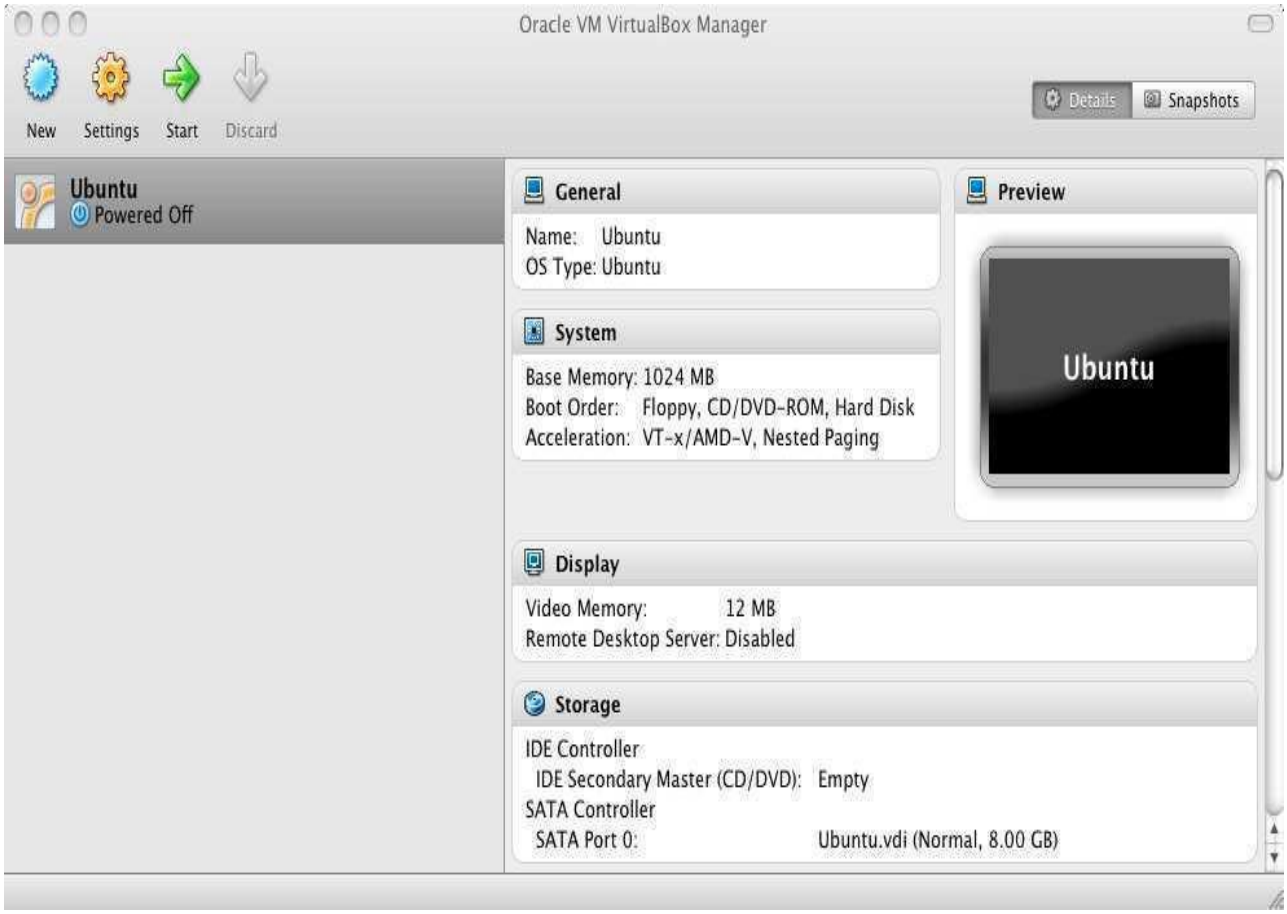
Select the size of the Virtual Disk (I recommend choosing 8 GB) and click continue



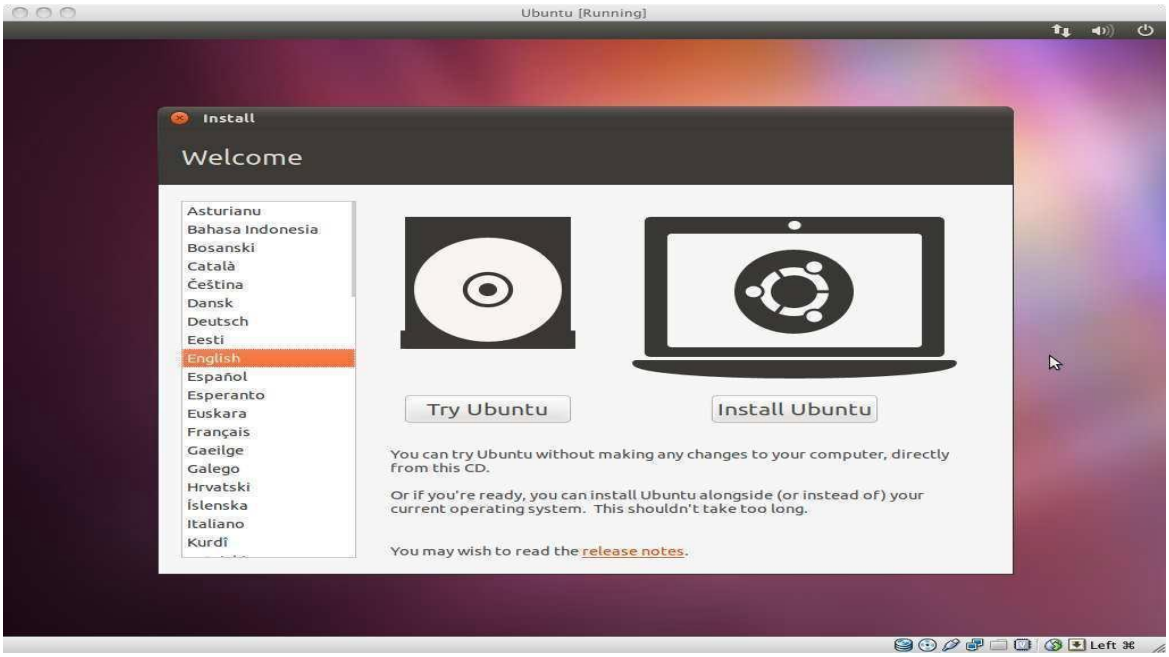
Click Create



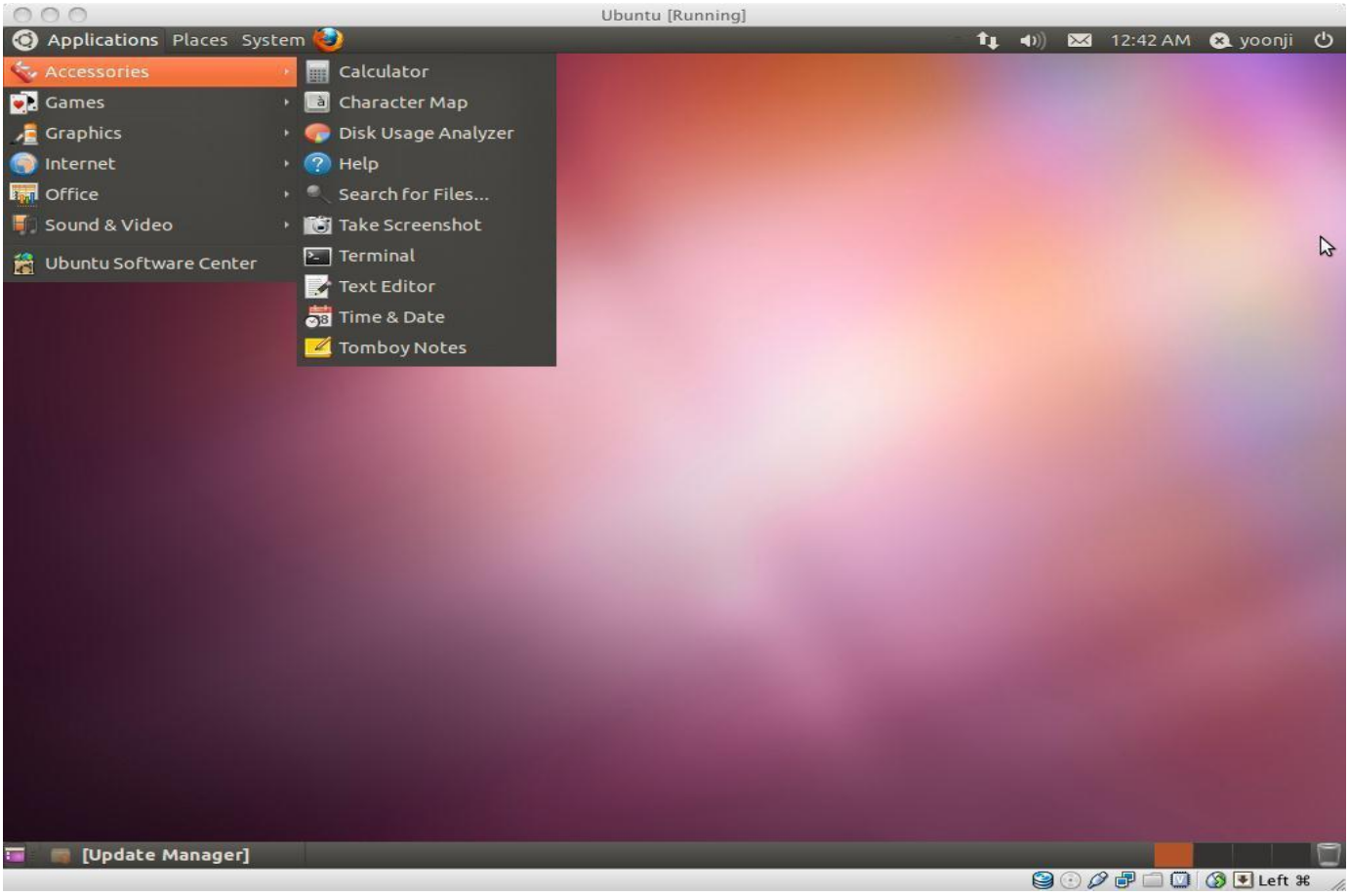
Choose Ubuntu from left column and click Start



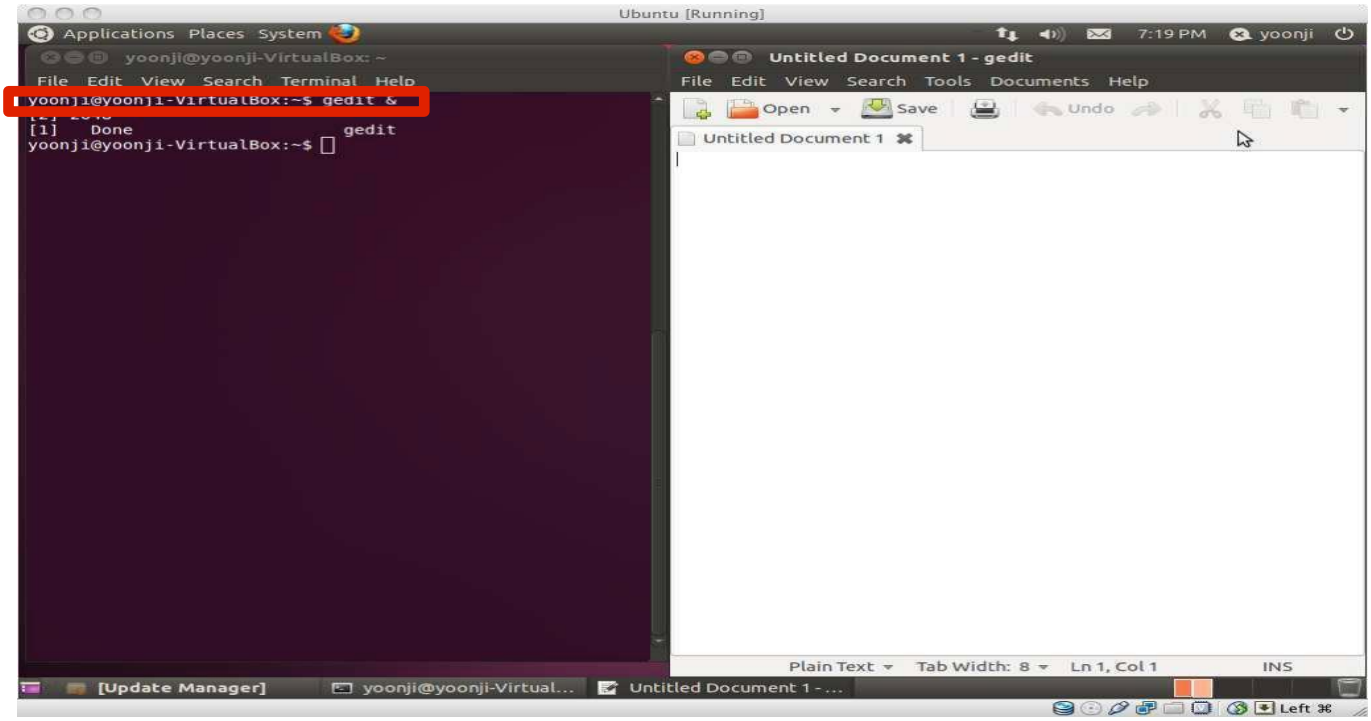
Click Install Ubuntu



Open Terminal (Applications-Accessories-Terminal)



Open gedit by typing “gedit &” on terminal



Step 4:

Write a sample program likeWelcome.cpp

```
#include<iostream.h>

using namespace std;

int main()

{

cout<<"Hello world";

return 0;

}
```

Step 5:

First we need to compile and link our program. Assuming the source code is saved in a file welcome.cpp, we can do that using GNU C++ compiler g++, for example g++ -Wall -o welcome welcome.cpp

Result:

Thus, the C compiler has been successfully installed and executed a sample program.

Ex. No:03 Date:	GOOGLE APP ENGINE

Aim:

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

Procedure:

Use **Eclipse** to create a **Google App Engine** (GAE) **Java** project (hello worldexample), run it locally and deploy it to Google App Engine account.

Tools used:

JDK 1.6

Eclipse 3.7 + Google Plugin for Eclipse

Google App Engine Java SDK 1.6.3.1

Note

GAE supports Java 1.5 and 1.6.

P.S Assume JDK1.6 and Eclipse 3.7 are installed.

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

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

Figure – New Web Application Project

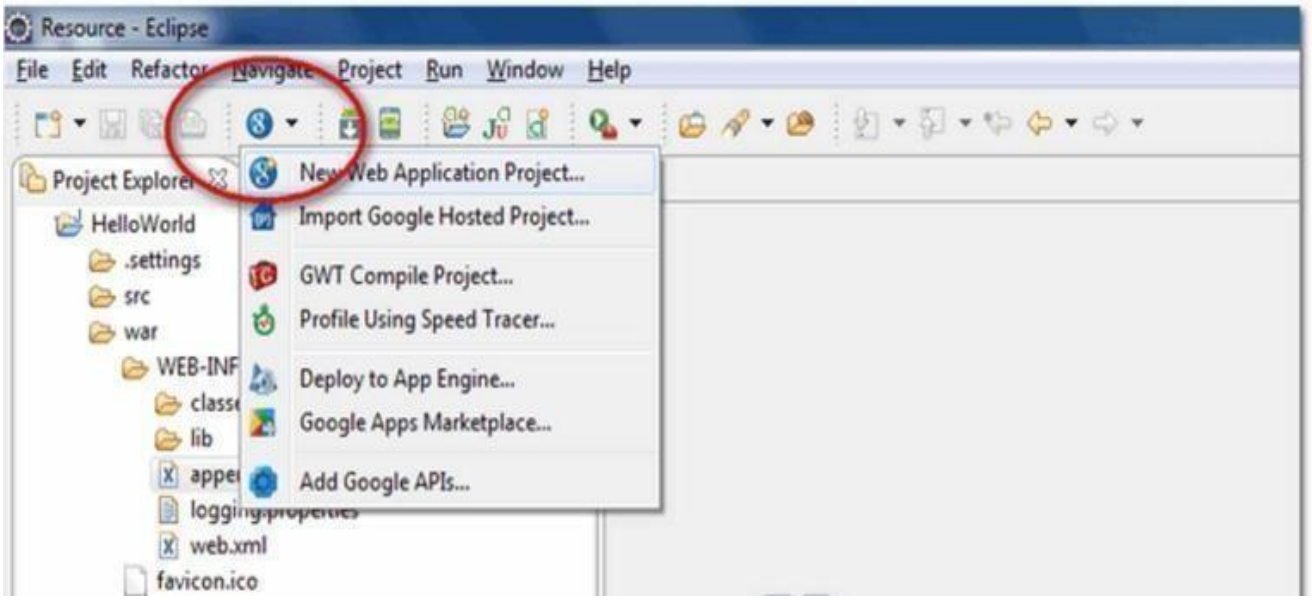
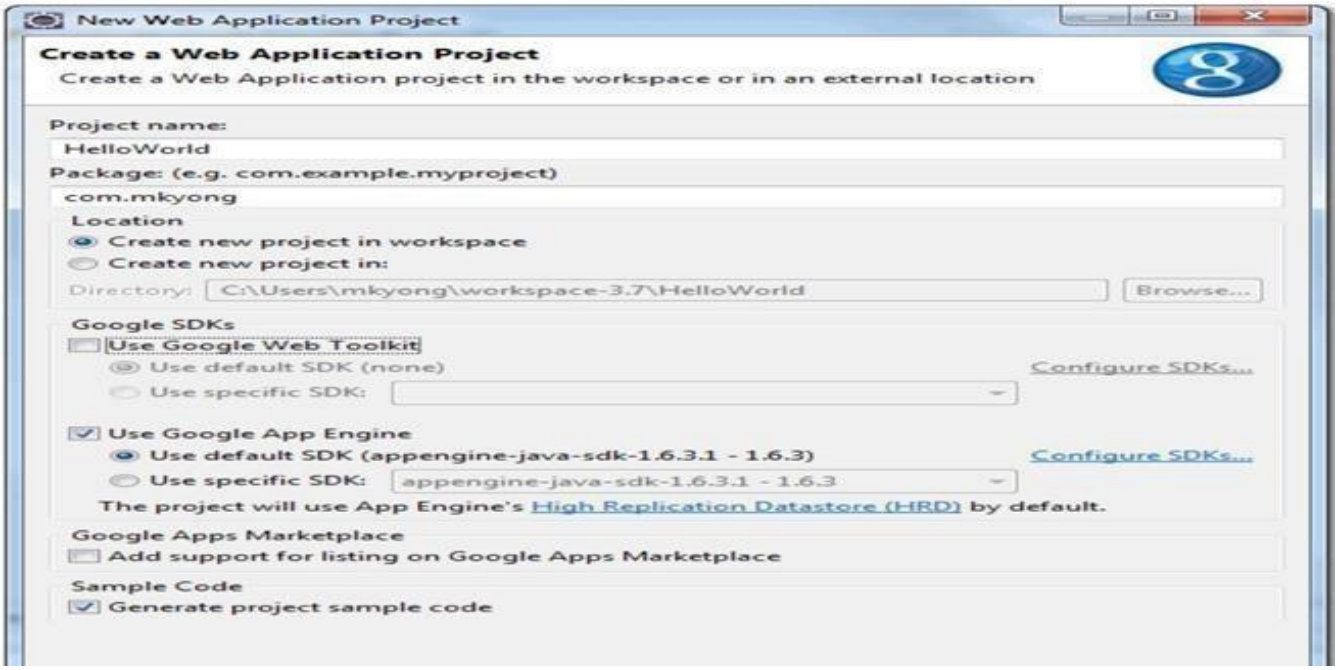


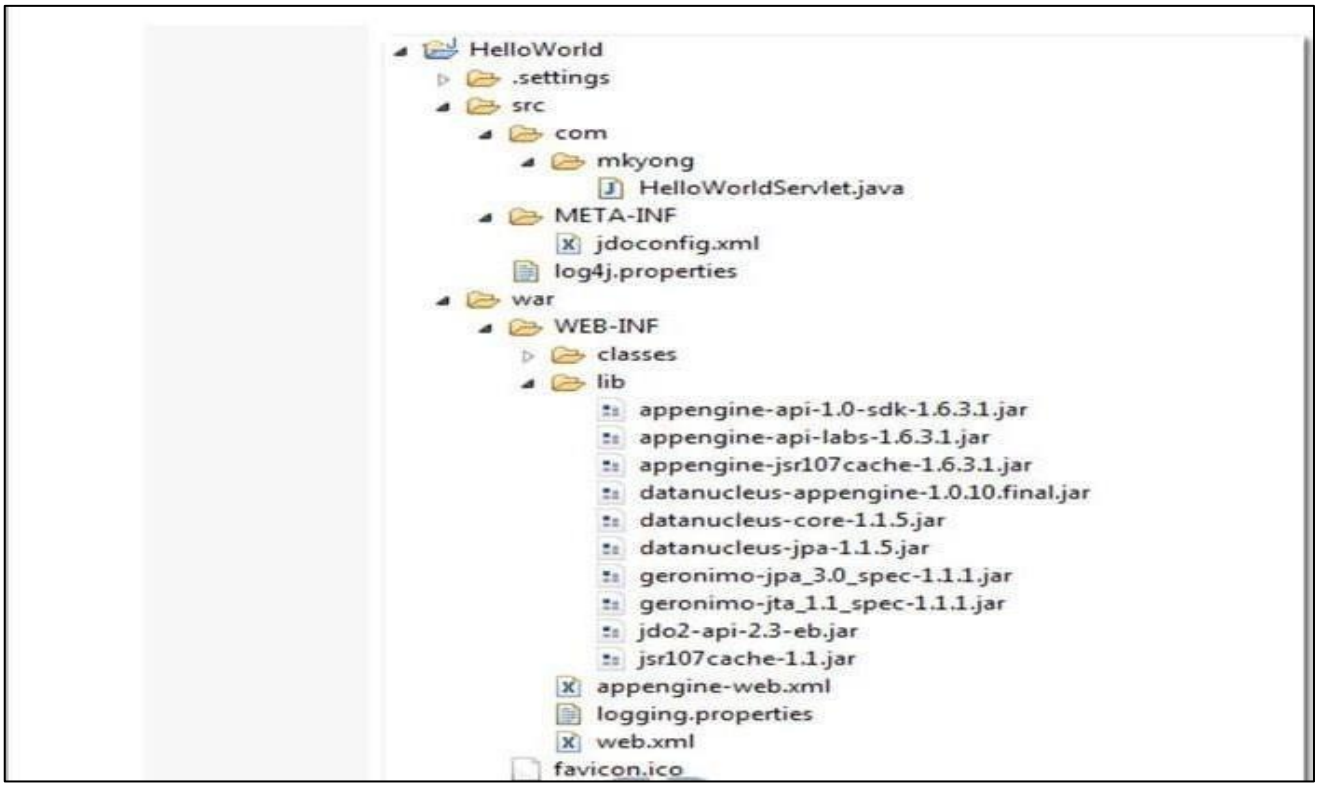
Figure – Deselect the "Google Web ToolKit", and link your GAE Java SDK via the "configure SDK" link.



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

3. Hello World

Review the generated project directory.



Nothing special, a standard Java web project structure.

HelloWorld/ src/

...Java source code... META-INF/

...other configuration... war/

...JSPs, images, data files... WEB-INF/

...app configuration... lib/

...JARs for libraries... classes/

...compiled classes...

Copy

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

File : appengine-web.xml

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

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

</appengine-web-app>
```

Copy

1. Run it local

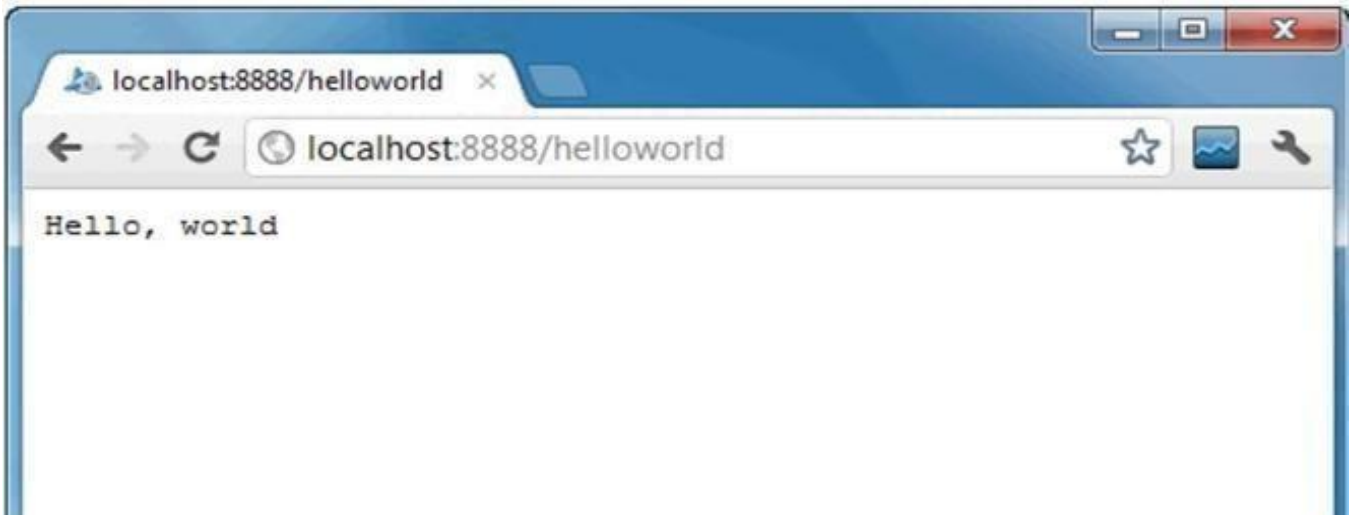
Right click on the project and run as "**Web Application**".

Eclipse console :

```
//...
INFO: The server is running at http://localhost:8888/
30 Mac 2012 11:13:01 PM
com.google.appengine.tools.development.DevAppServerImplstart 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



1. Deploy to Google App Engine

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

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

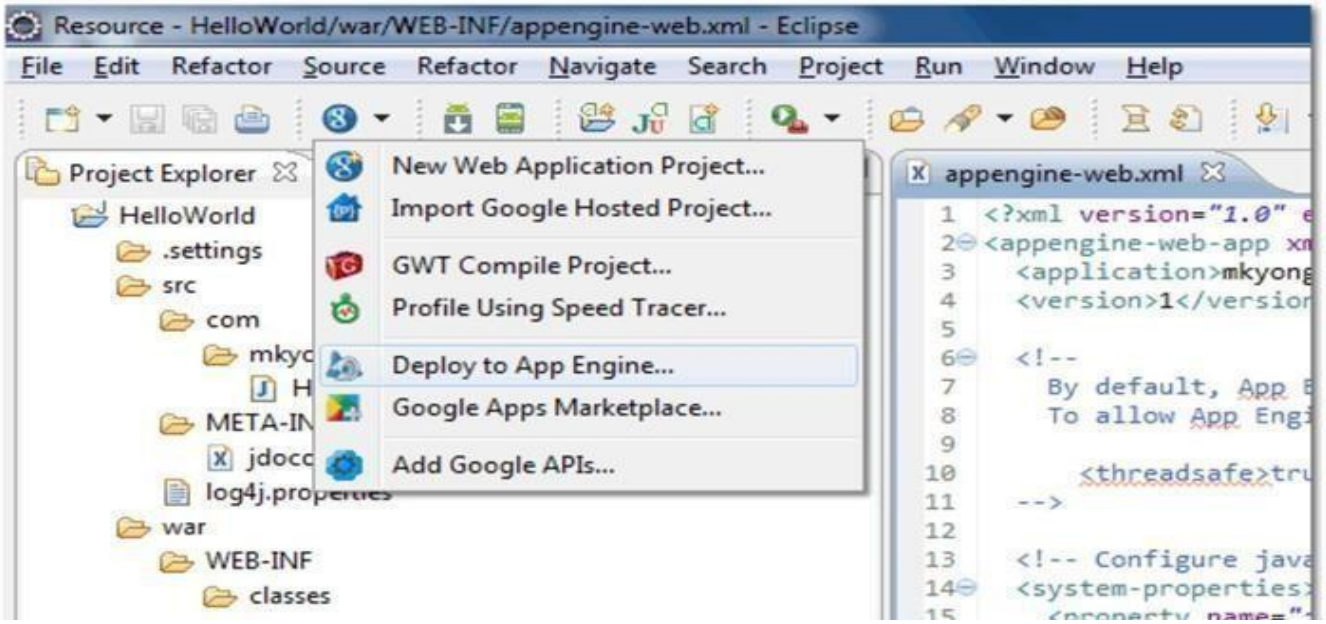
File : appengine-web.xml

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

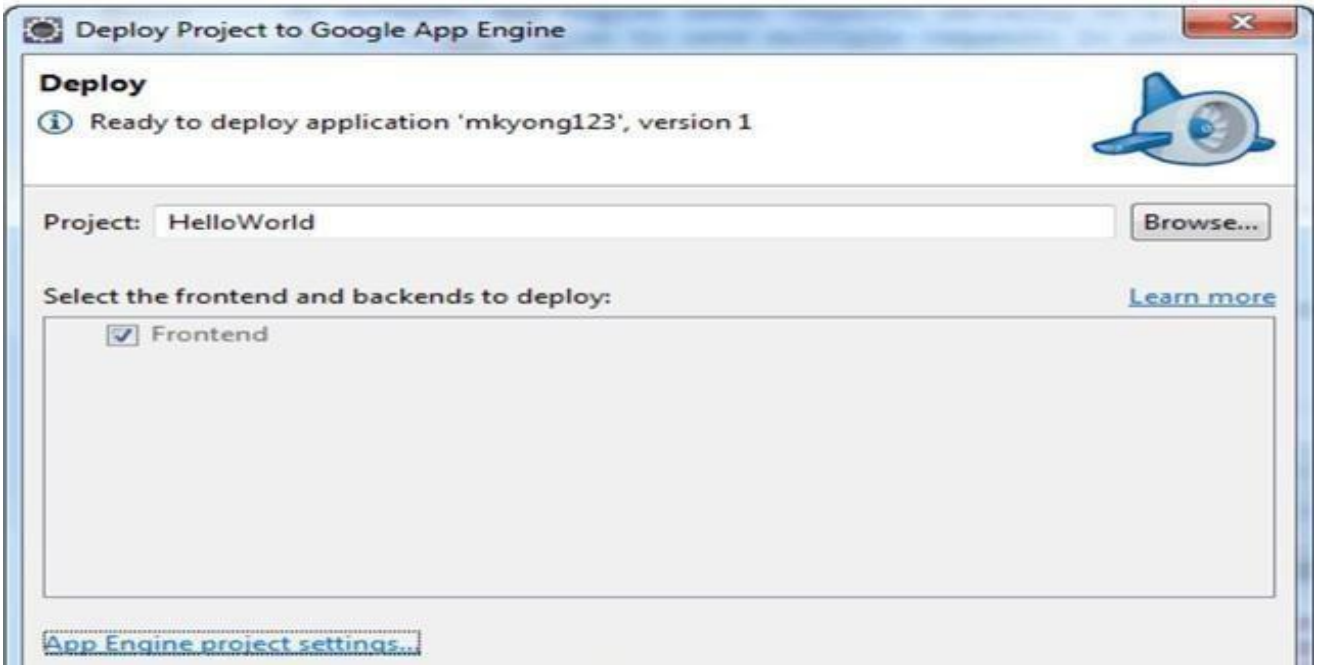
Copy

To deploy, see following steps:

Click on GAE deploy button on the toolbar.



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



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

Result:

Thus, the GAE is installed and executed the hello world application.

Ex. No: 04 Date:	GAE LAUNCHER

Aim:

To use GAE launcher to launch the web applications.

Procedure:

You can use Google App Engine to host a static website. Static web pages can contain client- side technologies such as HTML, CSS, and JavaScript. Hosting your static site on App Engine can cost less than using a traditional hosting provider, as App Engine provides a free tier.

Sites hosted on App Engine are hosted on the REGION_ID.r.appspot.com subdomain, such as [my-project-id].uc.r.appspot.com. After you deploy your site, you can map your own domain name to your App Engine-hosted website.

Before you begin

Before you can host your website on Google App Engine:

Create a new Cloud Console project or retrieve the project ID of an existing project to use: Go to the Project page

Tip: You can retrieve a list of your existing project IDs with the gcloud command line tool.

2. Install and then initialize the Google Cloud SDK: Download the SDK

Creating a website to host on Google App Engine Basic structure for the project

This guide uses the following structure for the project:

- app.yaml: Configure the settings of your App Engine application.
- www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
- css/: Directory to store stylesheets.
- style.css: Basic stylesheet that formats the look and feel of your site.
- images/: Optional directory to store images.
- index.html: An HTML file that displays content for your website.
- js/: Optional directory to store JavaScript files.
- Other asset directories.

Creating the app.yaml file

The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files. In the following steps, you will add handlers that will load www/index.html when someone visits your website, and all static files will be stored in and called from the www directory.

Create the app.yaml file in your application's root directory:

1. Create a directory that has the same name as your project ID. You can find your project ID in the Console.

2. In directory that you just created, create a file named app.yaml.

3. Edit the app.yaml file and add the following code to the file: runtime: python27

api_version: 1

threadsafe:

true handlers:

- url: /

static_files: www/index.html upload: www/index.html

- url: /(.*) static_files: www/\1 upload: www/(.*)

More reference information about the app.yaml file can be found in the app.yaml reference documentation.

Creating the index.html file

Create an HTML file that will be served when someone navigates to the root page of your website. Store this file in your www directory.

```
<html>
<head>
<title>Hello, world!</title>
<link rel="stylesheet"
type="text/css"
href="/css/style.css"> </head>
<body>
<h1>Hello, world!</h1>
<p>
This is a simple static HTML file that will be served from Google
App Engine.
</p>
</body>
</html>
```

Deploying your application to App Engine

When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the app.yaml file is located:

gcloud app deploy Optional flags:

- Include the --project flag to specify an alternate Cloud Console project ID to what you initialized as the default in the gcloud tool. Example: --project [YOUR_PROJECT_ID]

- Include the -v flag to specify a version ID, otherwise one is generated for you. Example: - v [YOUR_VERSION_ID]

To learn more about deploying your app from the command line, see Deploying a Python 2 App.

Viewing your application

To launch your browser and view the app at https://PROJECT_ID.REGION_ID.r.appspot.com, run the following command:

gcloud app browse

Result:

Thus, the GAE launcher used to launch the web applications.

Ex. No: 05 Date:	CLOUDSIM

Aim:

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

Pre--Requisites:

- JAVA installation and Configuring in Environmental Variable
- Eclipse – Java for Developer
- Apache Commons Math jar file
- CloudSim 3.0.3 libraries

Procedure:

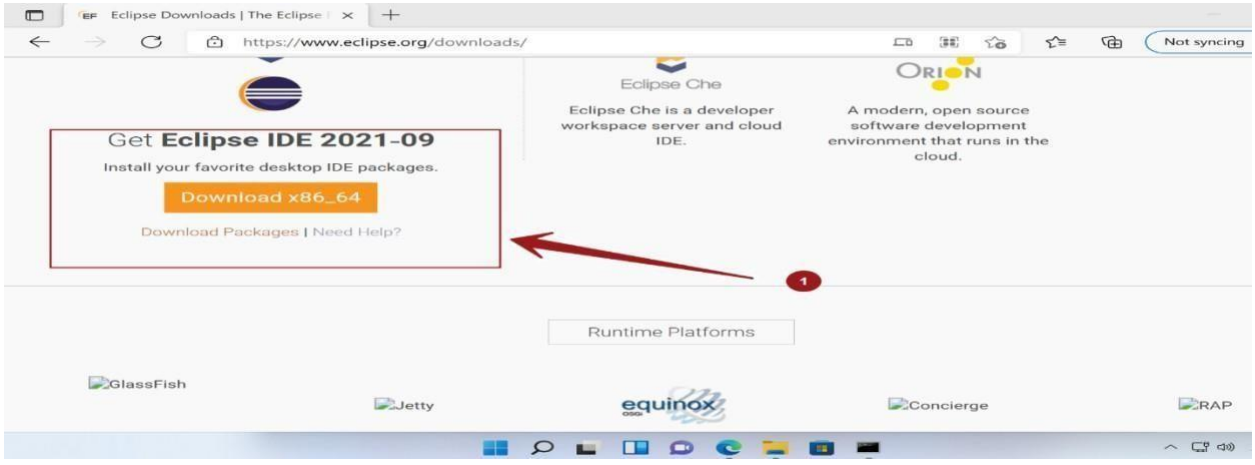
Install Eclipse IDE on Windows

Downloading an Eclipse IDE is very easy just open your browser, search for Eclipse IDE and open the link from eclipse.org, click on Download. It will download an Eclipse installer file.

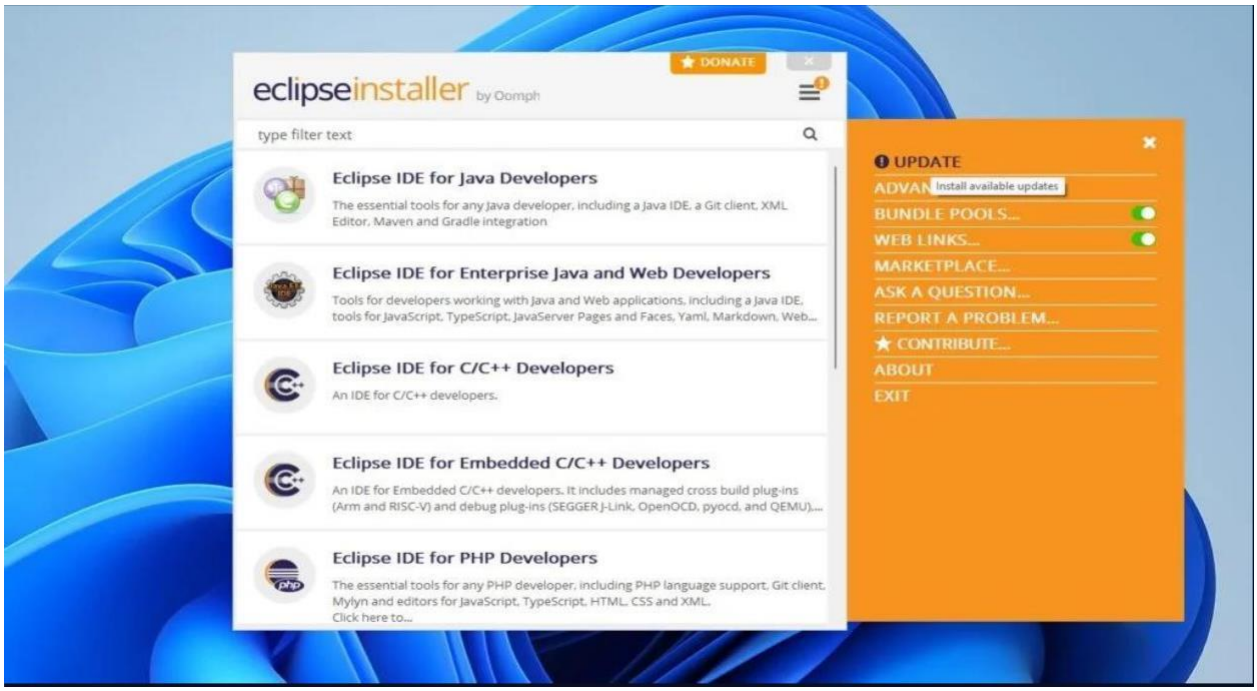
After the download has been completed, follow the following steps to install it.

Step 1: After the download has been completed open the eclipse installer

file. https://www.eclipse.org/downloads/



Step 2: Now if you find any notification regarding the update. You need to update the eclipse installer file before installing "**Eclipse IDE for Java Developers**". For that simply click on the Hamburg menu, click on Update and accept the license agreement.



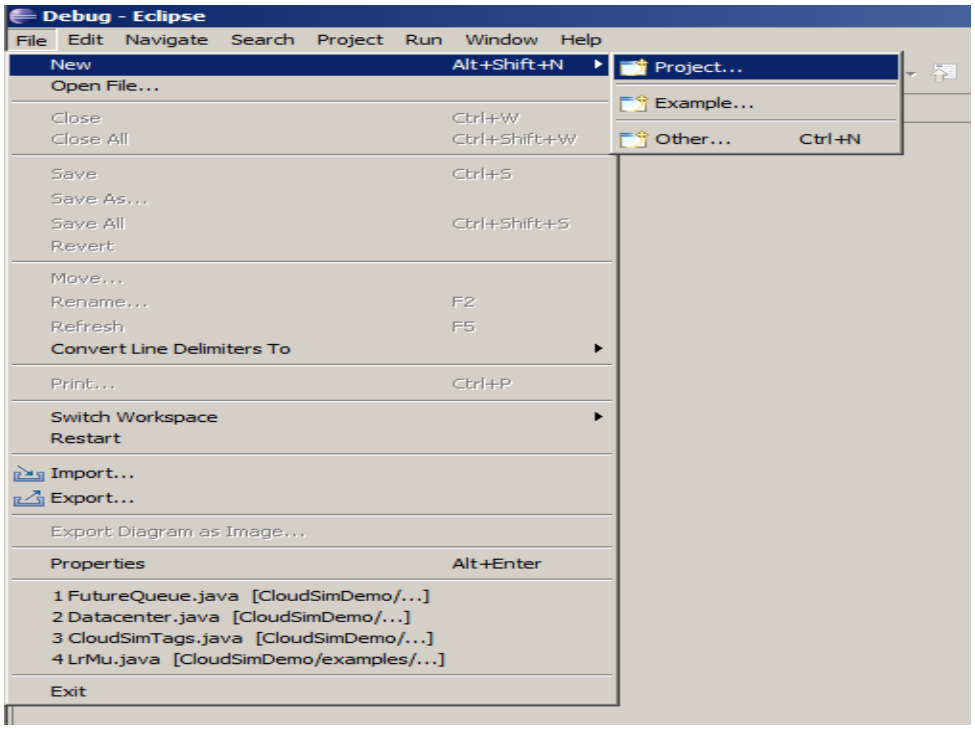
Step 3: If you don't find any update notification then you can click on "**Eclipse IDE for Java Developers**" and select the "**Java Installation Folder**" and the location where you want to install Eclipse IDE.

After that, click on "**INSTALL**" and accept the license agreement in order to install Eclipse IDE for Java Developers. (*An active internet connection is needed to install Eclipse IDE*).

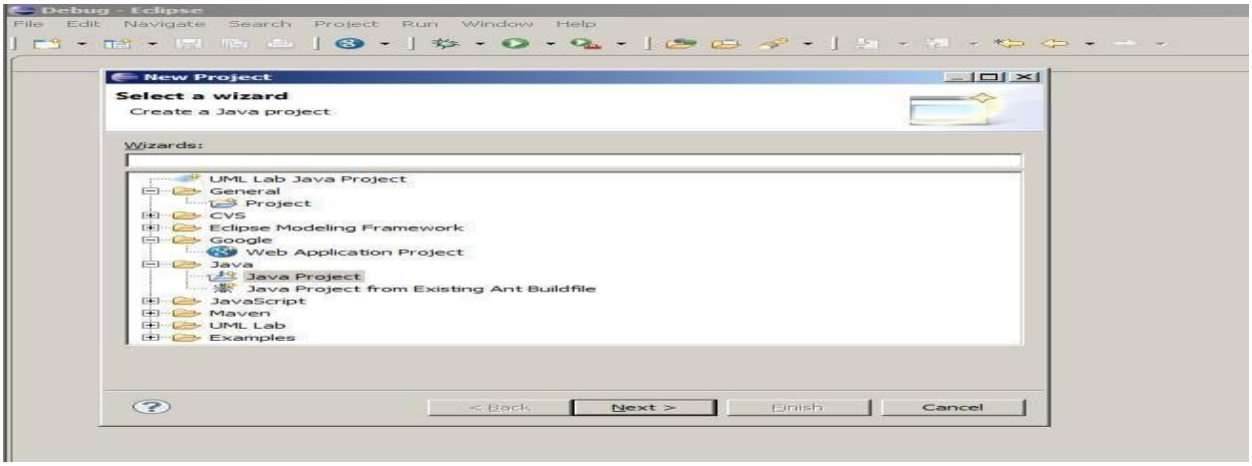
Step 4: After the installation is complete, click on the "LAUNCH" button to open Eclipse IDE.

CloudSim Setup using Eclipse

- Open Eclipse IDE, now within Eclipse window navigate the menu: ***File -> New -> Project***, to open the new project wizard



- A ‘New Project‘ wizard should open. There are a number of options displayed and you have to find & select the ‘Java Project‘ option, once done click ‘Next‘

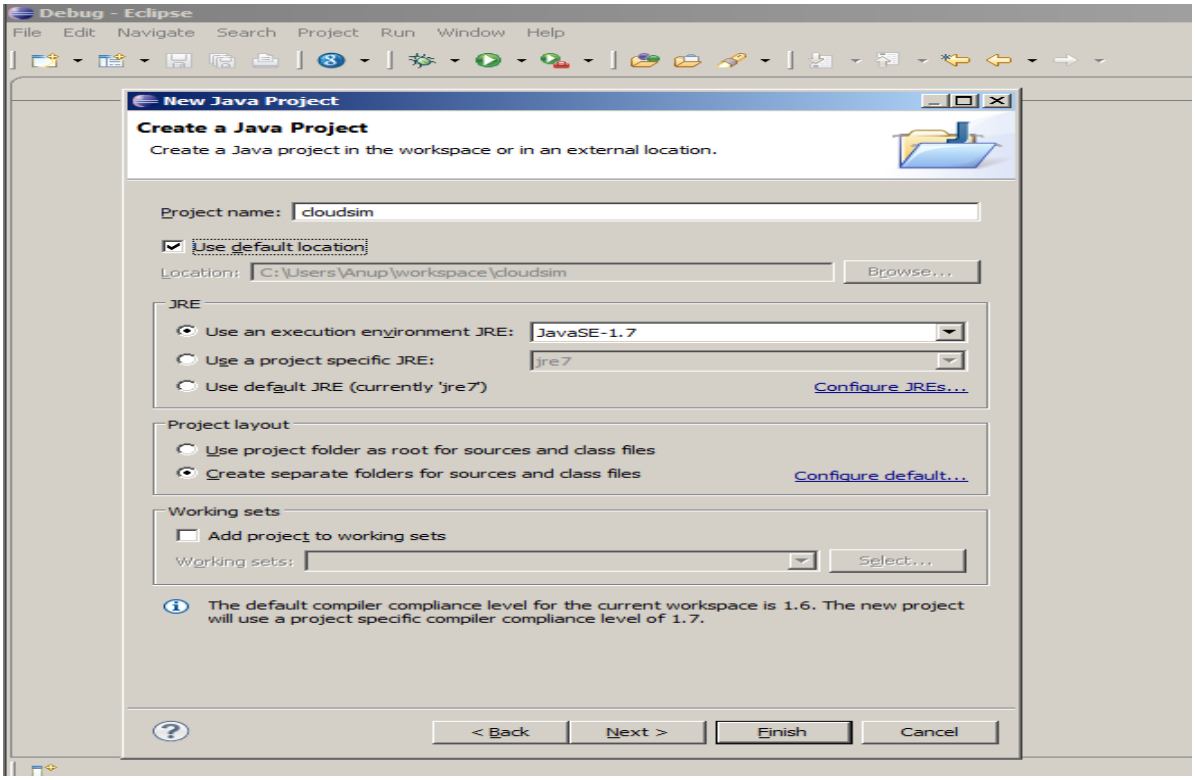


Download CloudSim 3.0.3 in below location

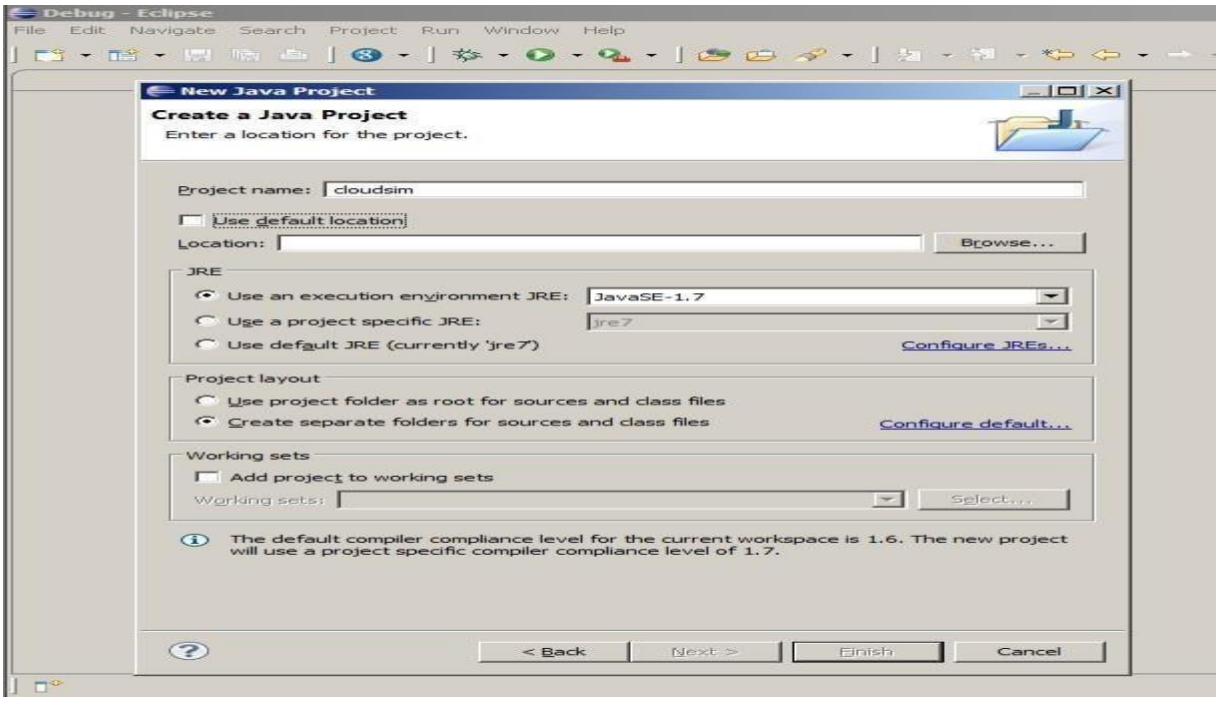
https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-3.0.3

Now a detailed new project window will open, here you will provide the project name and the path of CloudSim project source code, which will be done as follows:

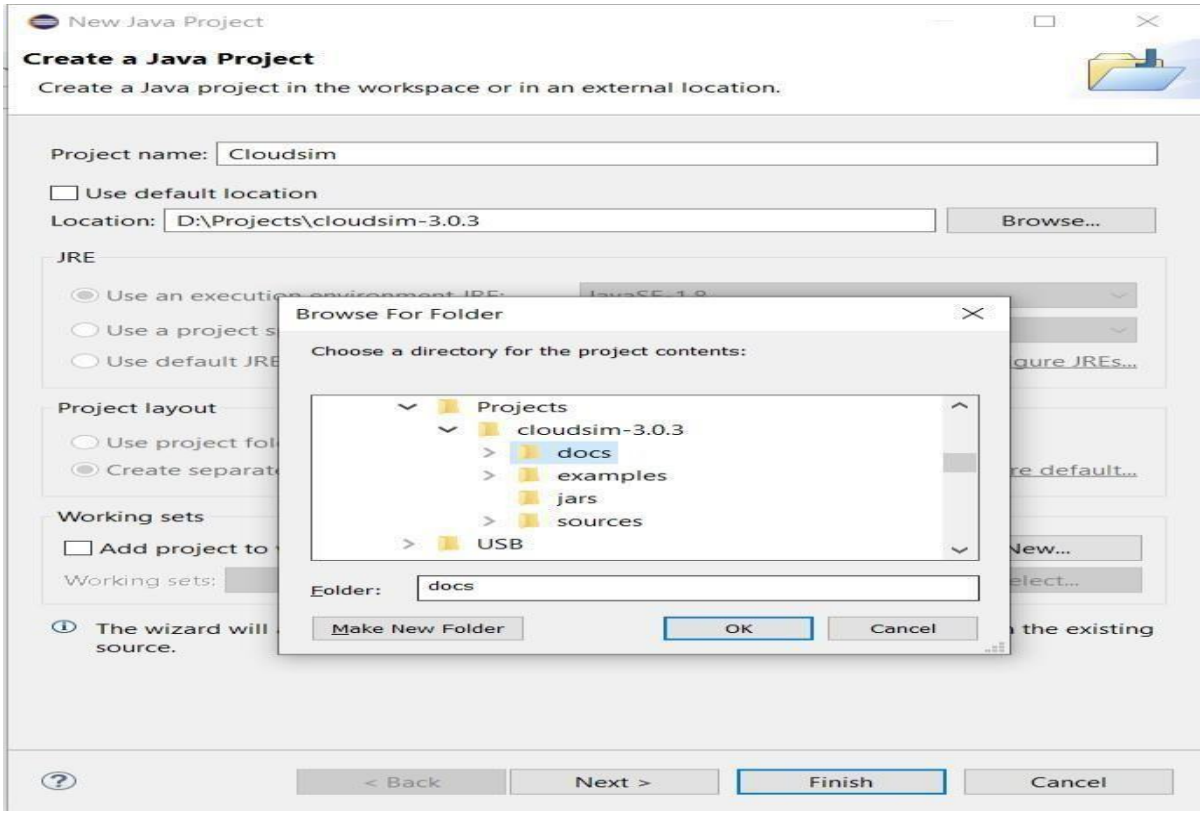
- Project Name: CloudSim.



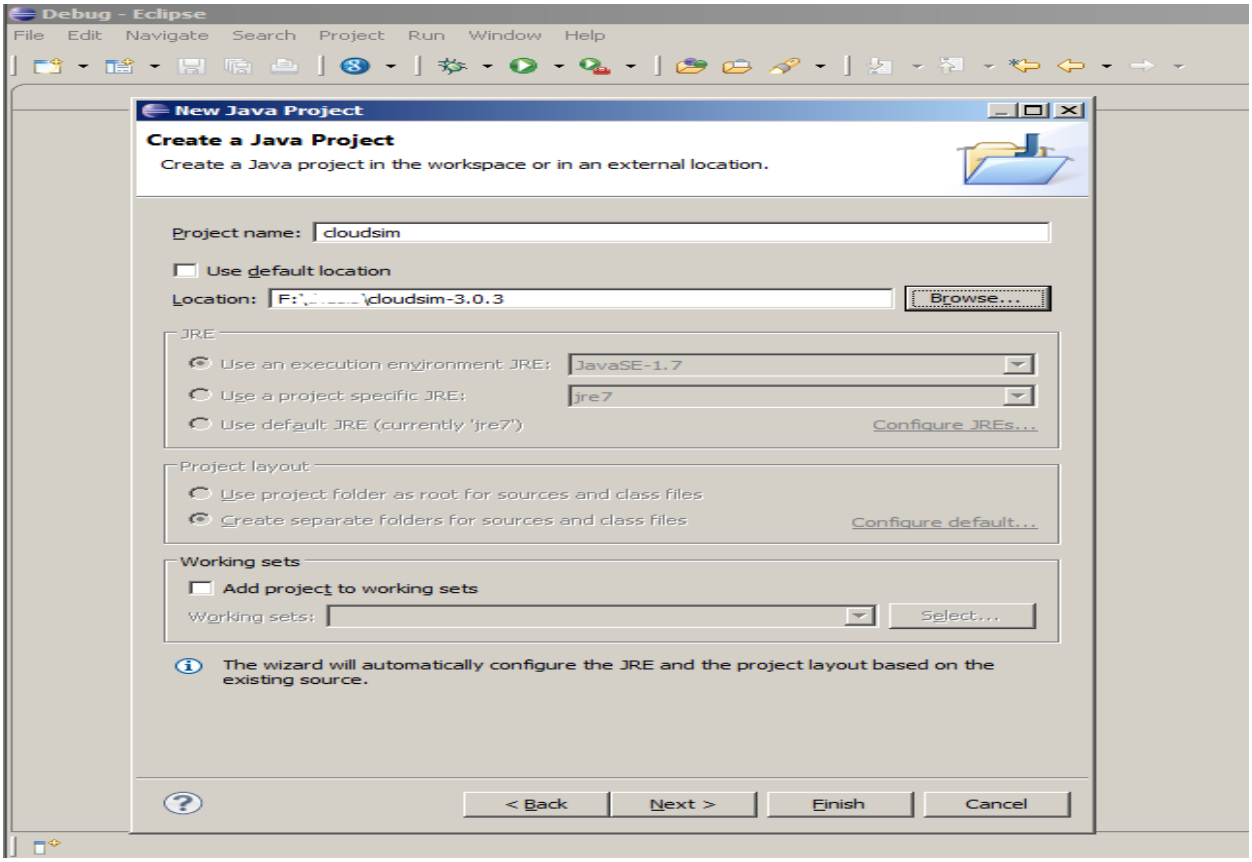
Unselect the ‘Use default location’ option and then click on ‘Browse’ to open the path where you have unzipped the Cloudsim project and finally click Next to set project settings.



Make sure you navigate the path till you can see the bin, docs, examples etc folder in the navigation plane.

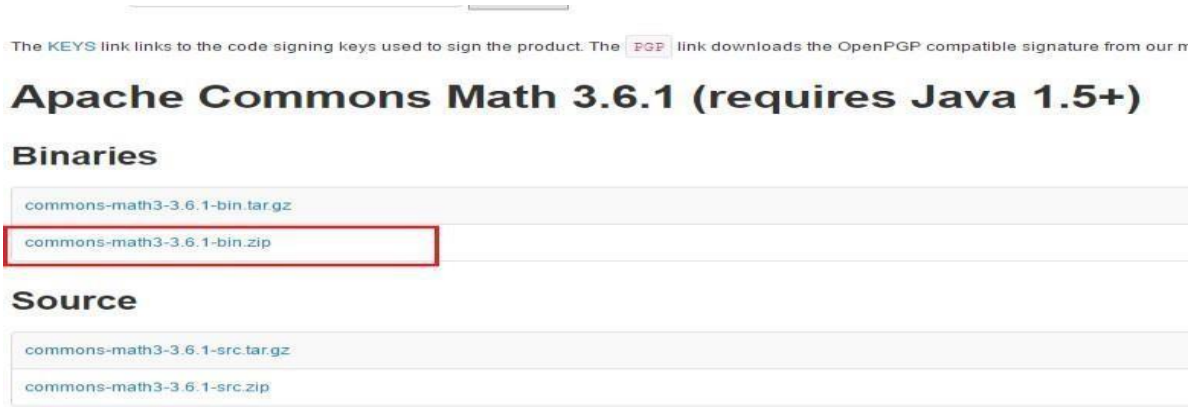


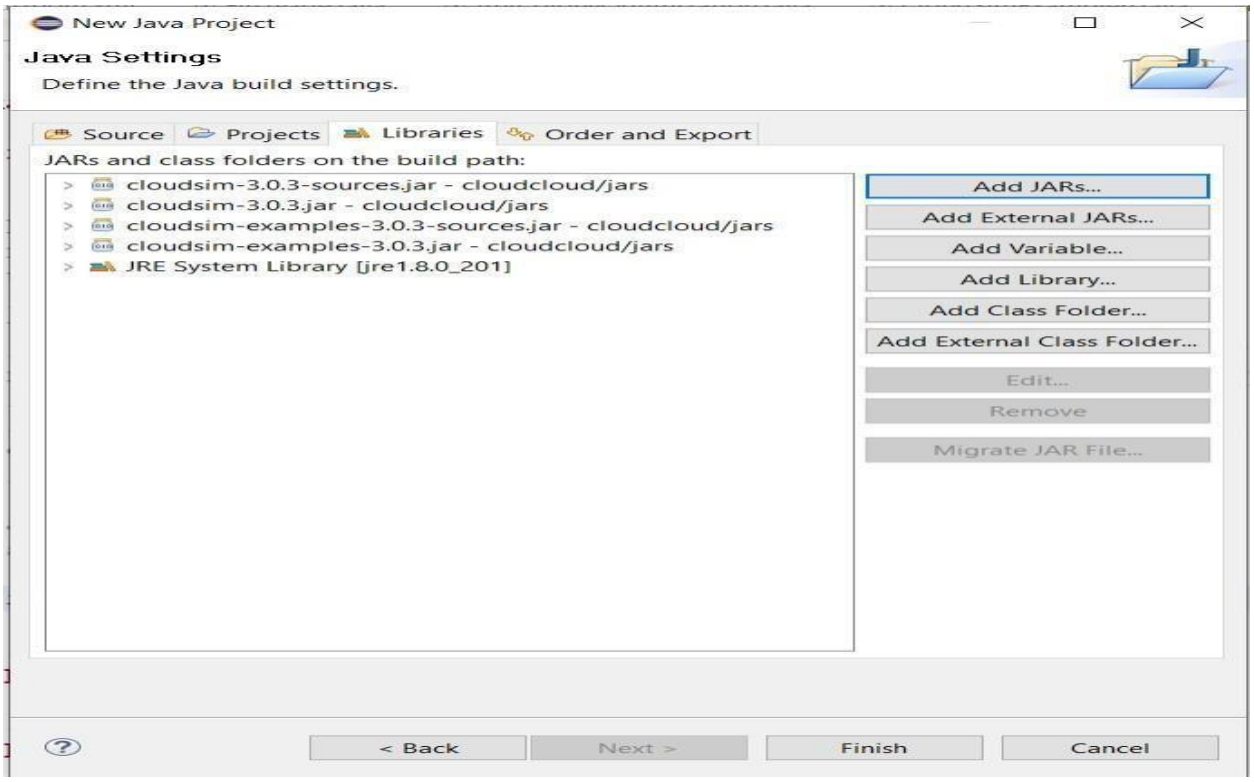
Once done finally, click ‘Next’ to go to the next step i.e. setting up of project settings



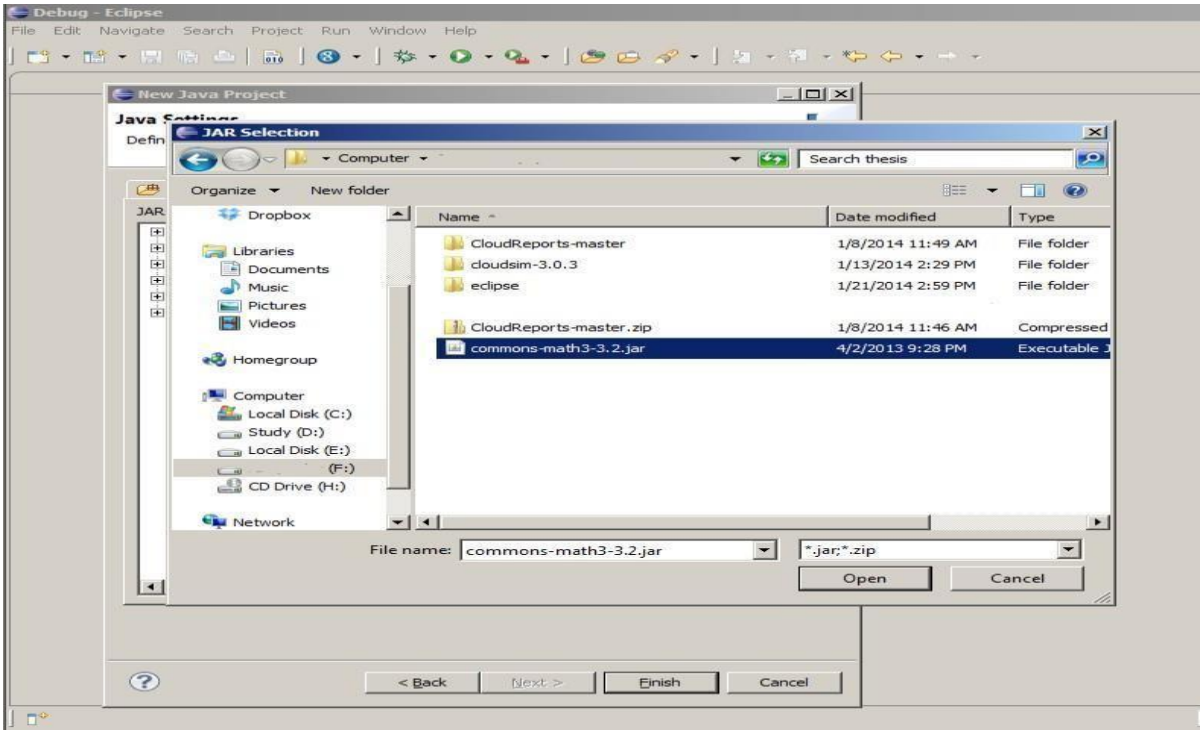
Now open ‘Libraries’ tab and if you do not find **commons-math3-3.x.jar** (here ‘x’ means the minor version release of the library which could be 2 or greater) in the list then simply click on **‘Add External Jar’** (commons-math3-3.x.jar will be included in the project from this step)

Step 5: To successfully configure CloudSim Simulation toolkit we have to download a file named **'commons-math3-3.6.1-bin.zip'**, this file can be downloaded from commons math website **'http://commons.apache.org/proper/commons-math/download_math.cgi'** and extract 'commons- math3-3.6.1.jar'.

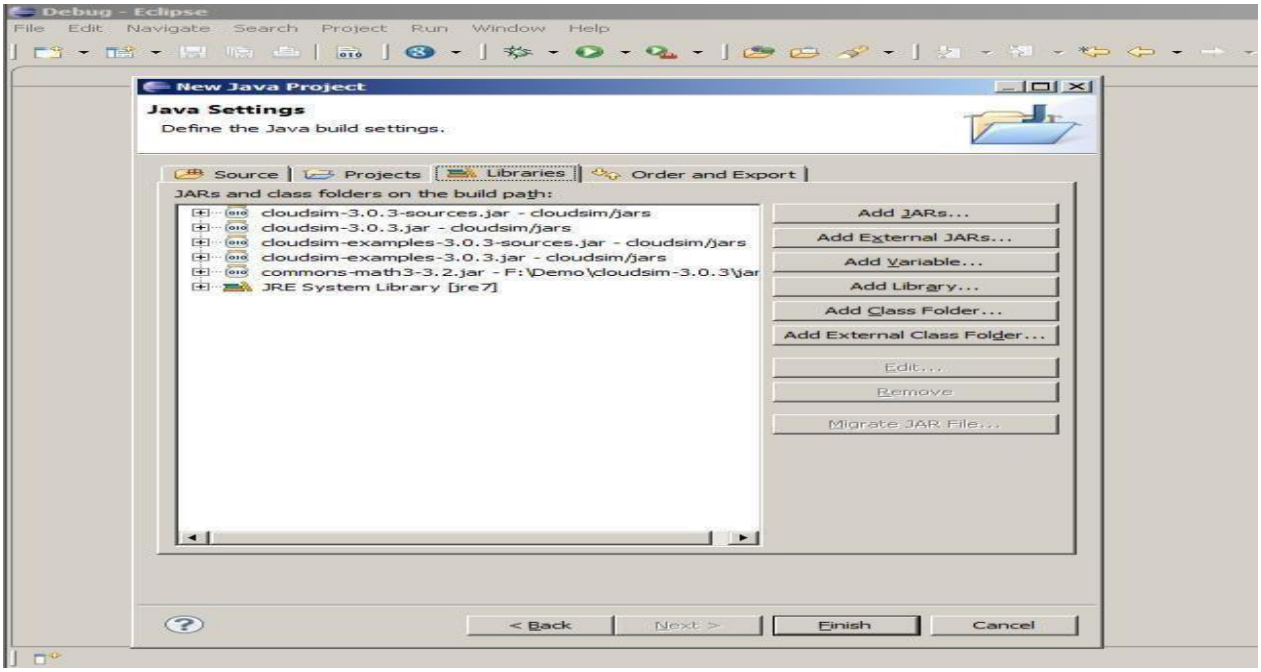




Once you have clicked on ‘Add External JAR’s‘ Open the path where you have unzipped the commons- math binaries and select ‘Commons-math3-3.x.jar’ and click on open.

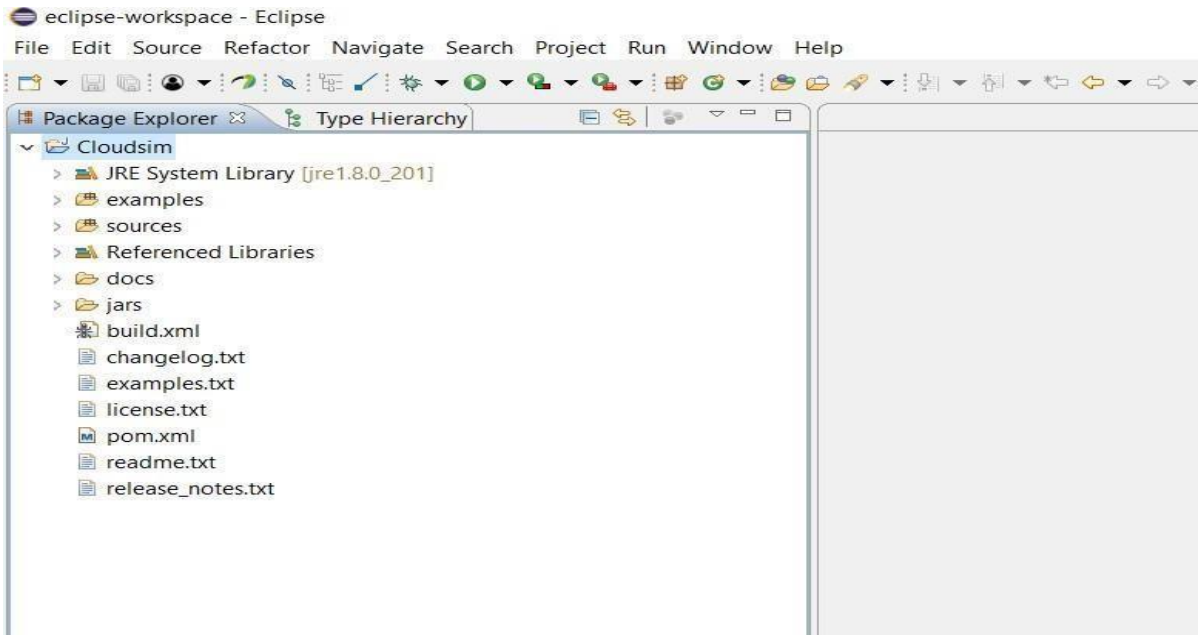


Ensure external jar that you opened in the previous step is displayed in the list and then click on **'Finish'** (your system may take 2-3 minutes to configure the project)

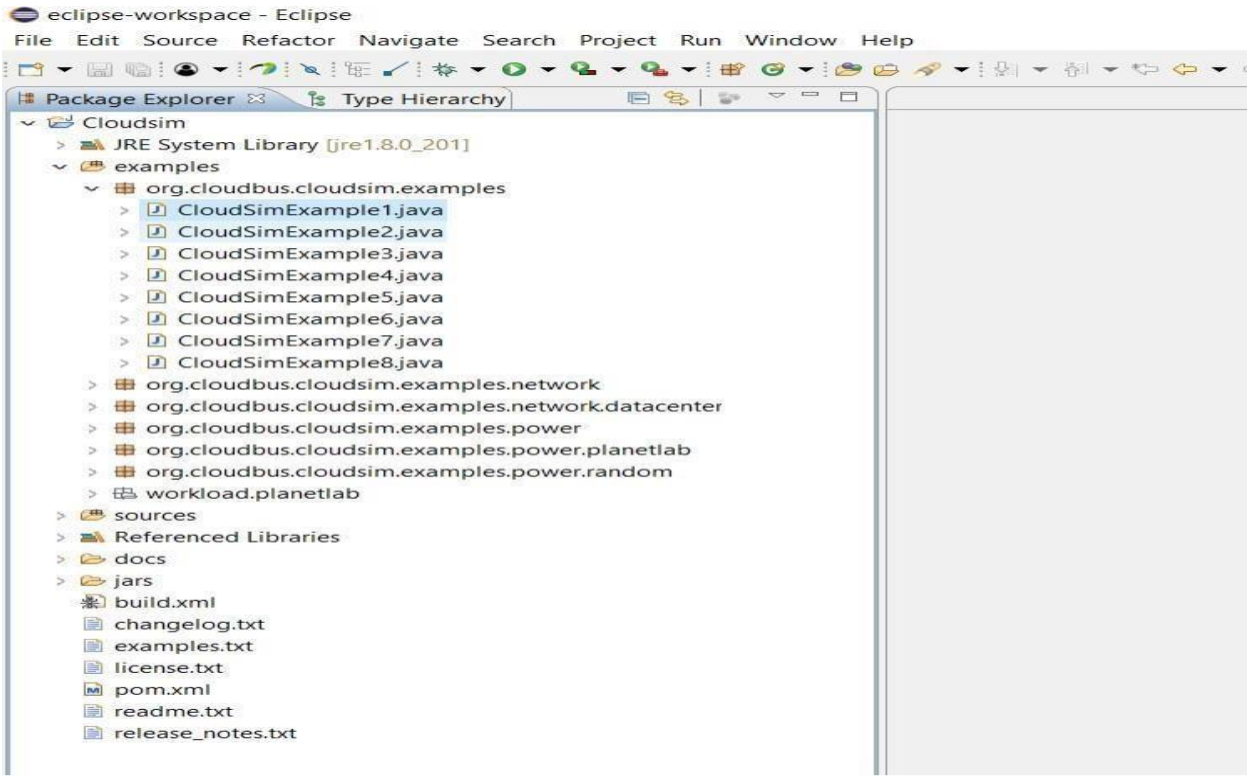


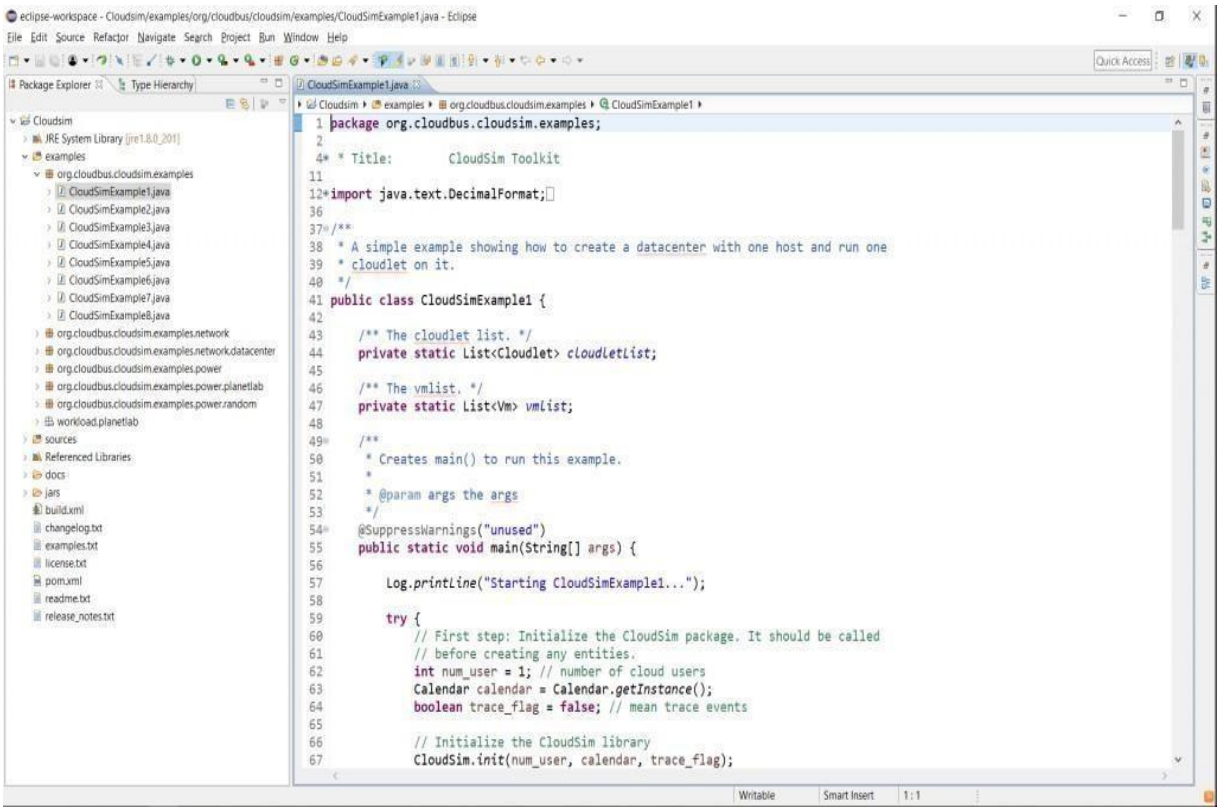
Once the project is configured you can open the **'Project Explorer'** and start exploring the Cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured Cloudsim project, which may take some time depending on the configuration of the computer system.

Following is the final screen which you will see after Cloudsim is configured.

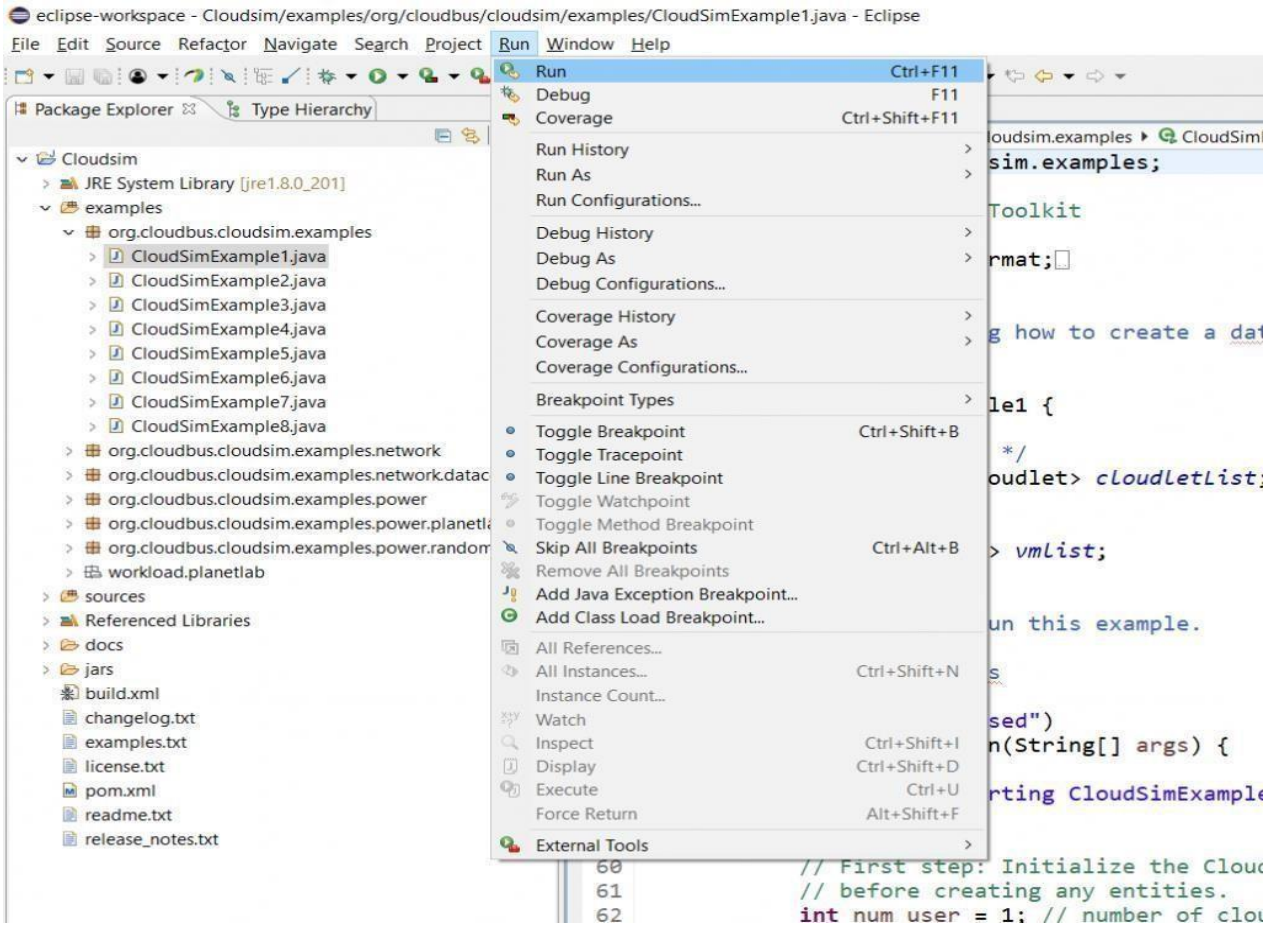


Now just to check you within the '**Project Explorer**', you should navigate to the '**examples**' folder, then expand the package '***org.cloudbus.cloudsim.examples***' and double click to open the '***CloudsimExample1.java***'.

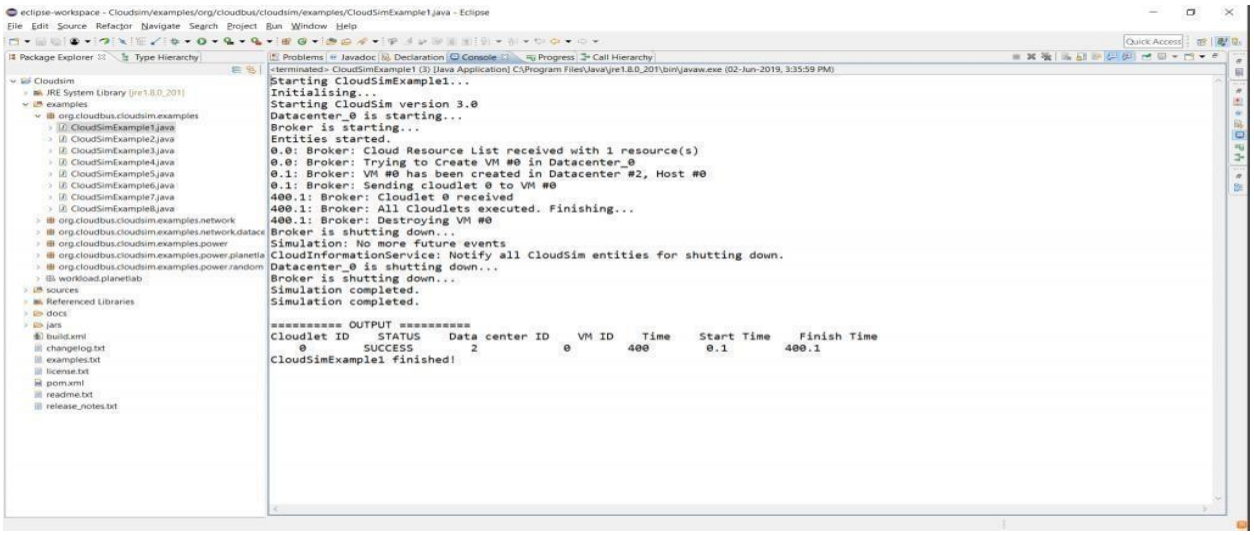




Now navigate to the Eclipse menu '***Run -> Run***' or directly use a keyboard shortcut ***'Ctrl + F11'*** to execute the '***CloudsimExample1.java***'.



If it is successfully executed it should be displaying the following type to output in the console window of the Eclipse IDE.



If Yes then Congratulations!! Now you have successfully setup/Installed the CloudSim Simulation.

Result:

Thus, the cloud scenario using CloudSim and the scheduling algorithm in CloudSim are executed.

Ex. No: 06 Date:	VIRTUAL MACHINE OPERATION

Aim:

To transfer the files from one virtual machine to another virtual machine.

Procedure:

You can move files between virtual machines in several ways:

- You can copy files using network utilities as you would between physical computers on your network. To do this between two virtual machine:

o Both virtual machines must be configured to allow access to your network. Any of the networking methods (host-only, bridged and NAT) are appropriate.

o With host-only networking, you copy files from the virtual machines to the host and vice-versa, since host-only networking only allows the virtual machines see your host computer.

o With bridged networking or NAT enabled, you can copy files across your network between the virtual machines.

- You can create a shared drive, either a virtual disk or a raw partition, and mount the drive in each of the virtual machines.

How to Enable File sharing in VirtualBox.

Step 1. Install Guest Additions on the Guest machine.

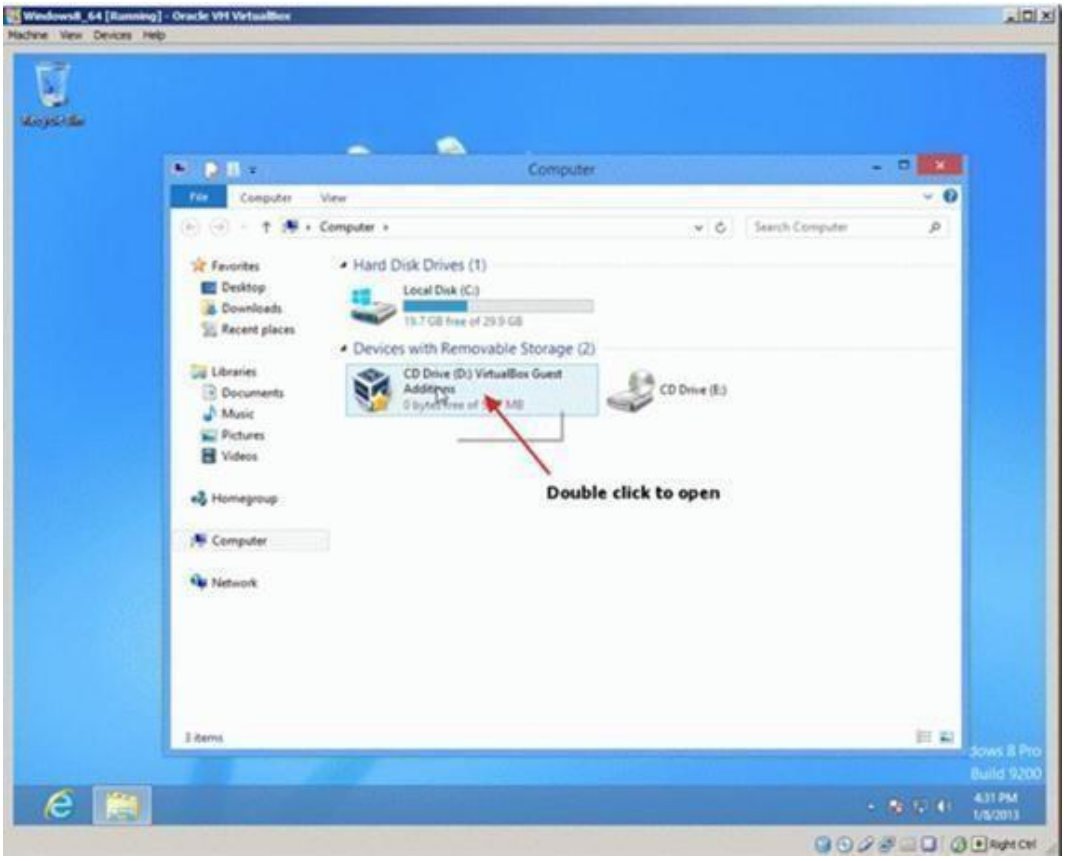
Step 2. Configure File Sharing on VirtualBox.

Step 1. Install Guest Additions on the Guest machine.

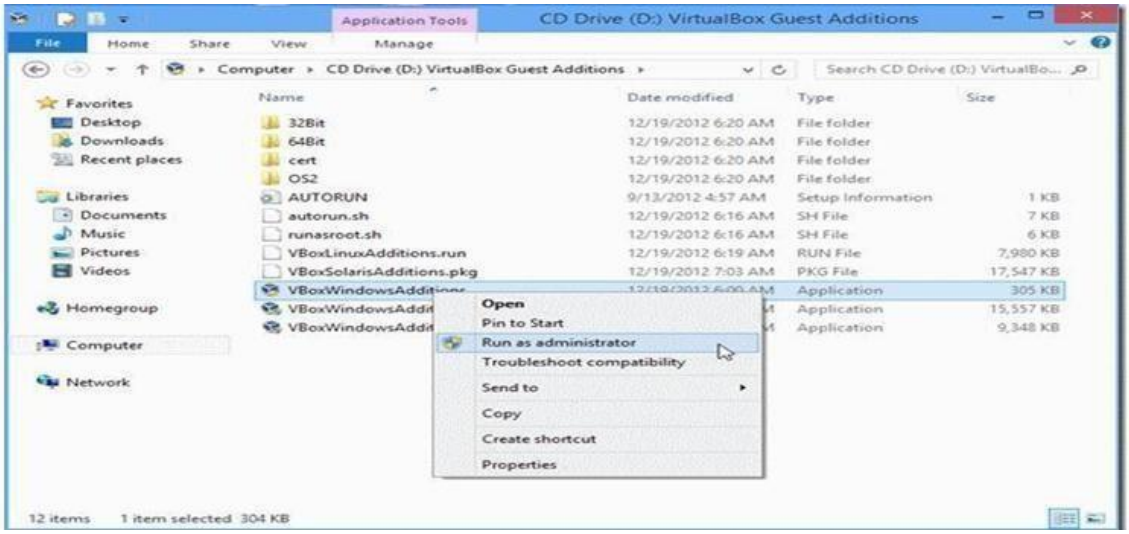
1. Start the Virtuabox Guest Machine (OS).
2. From Oracle's VM VirtualBox main menu, select Devices > Install Guest Additions *

Open Windows Explorer.

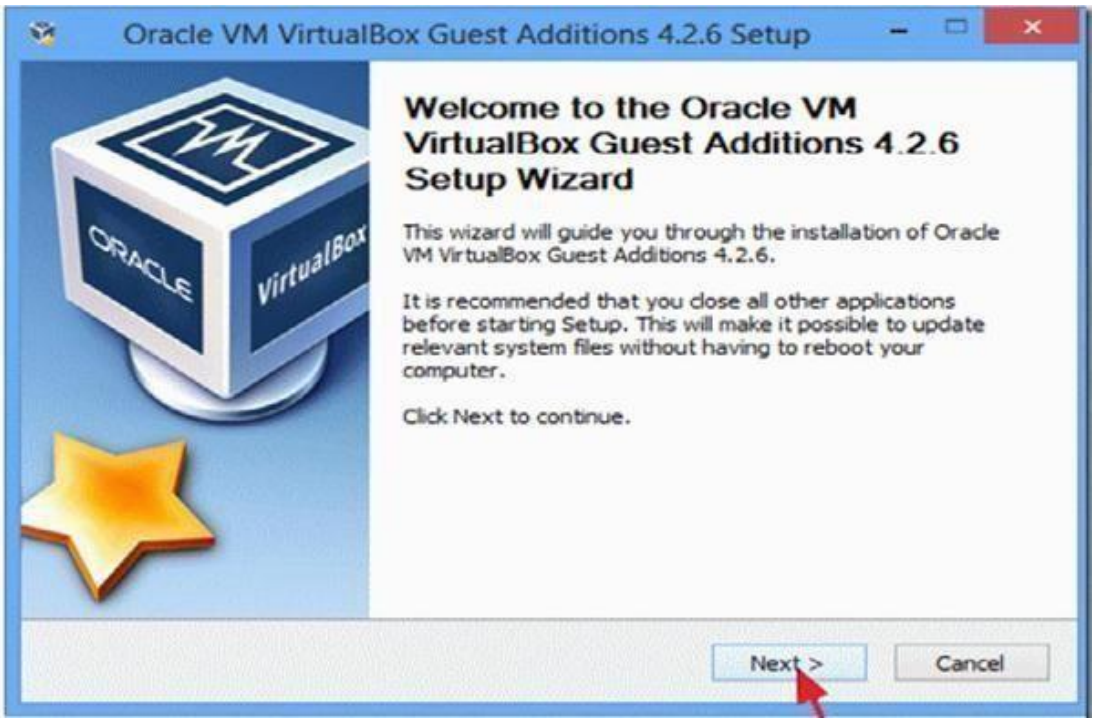
Double click at the "CD Drive (X:) VirtualBox Guest additions" to explore its contents.



Right click at "VBoxWindowsAdditions" application and from the pop-up menu, choose "Run as administrator"



Press **Next** and then follow the on screen instructions to complete the Guest Additions installation.

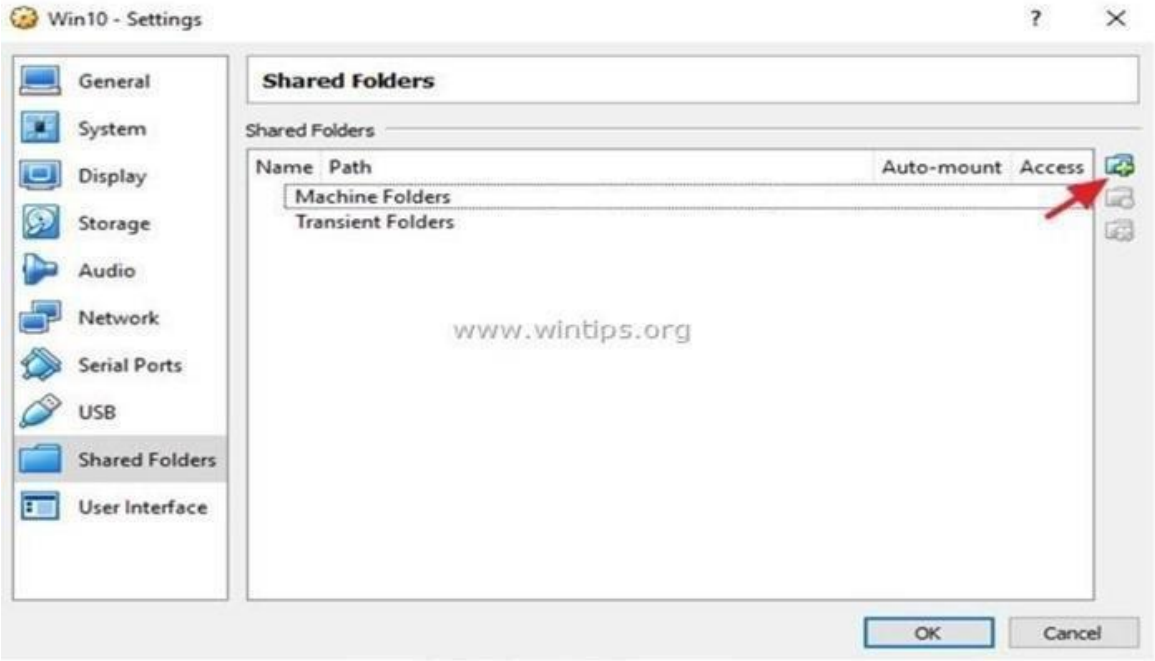


When the setup is completed, choose Finish and restart the Virtuabox guest machine

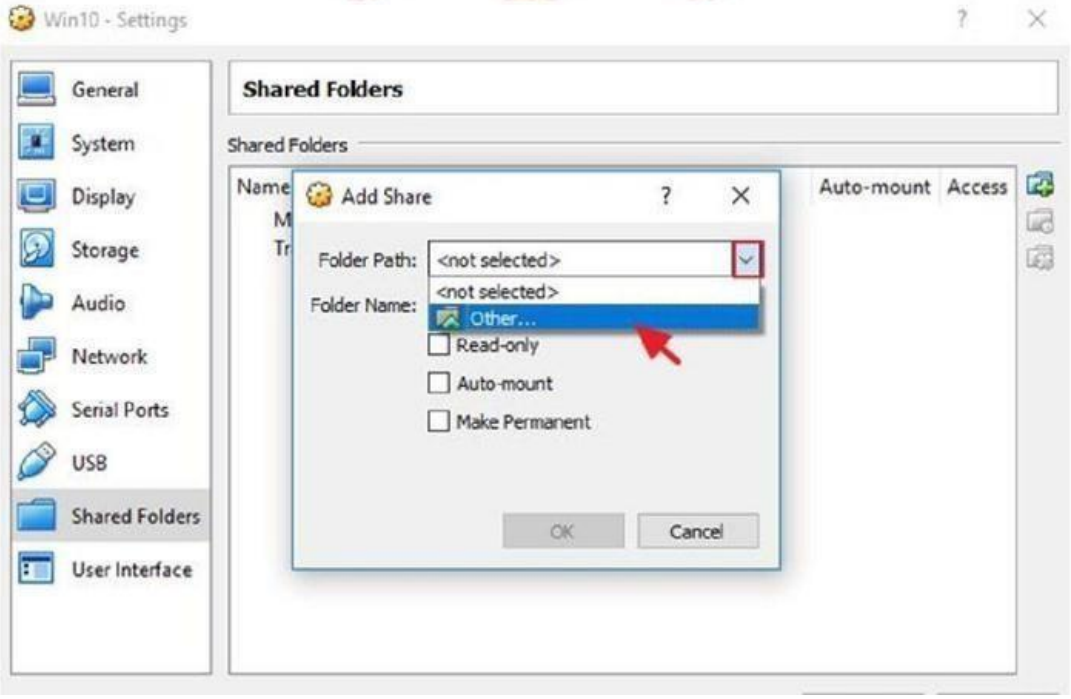
Step 2. Setup File Sharing on VirtualBox Guest Machine.

1. From VirtualBox menu click Devices and choose Shared Folders -> Shared Folder Settings.

2. Click the Add new shared folder icon.

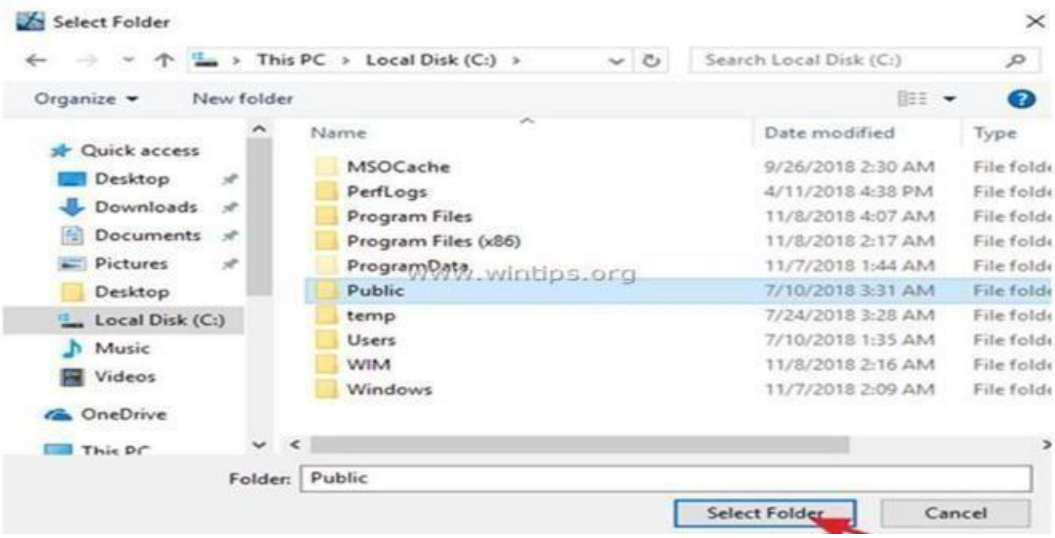


3. Click the drop-down arrow and select other.



4. Locate and highlight (from the Host OS) the folder that you want to share betweenthe VirtualBox Guest machine and the Host and click Select Folder. *

* Note: To make your life easier, create a new folder for the file sharing, on the Host OS and give it with a recognizable name. (e.g. "Public")



Now, in the 'Add Share' options, type a name (if you want) at the 'Folder Name box,click the **Auto Mount** and the **Make Permanent** checkboxes and click **OK** twice toclose the Shared Folder Settings.

5. You 're done! To access the shared folder from the Guest OS, open Windows Explorer and under the 'Network locations' you should see a new network drive that corresponds to the shared folder on the Host OS.

Result:

Thus, the virtual machine files are moved to another VM.

Ex. No: 07 Date:	HADOOP SINGLE NODE CLUSTER

Aim:
To Set up the one node Hadoop Cluster and execute a word count program.

Procedure:
1) Installing Java:

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware. Hadoop needs Java 6 or above to work.

Step 1: Download Jdk tar.gz file for linux-62 bit, extract it into "/usr/local"
boss@solaiv[]# cd /opt
boss@solaiv[]# sudo tar xvpzf /home/itadmin/Downloads/jdk-8u5-linux-x64.tar.gz boss@solaiv[]# cd /opt/jdk1.8.0_05

Step 2:
Open the "/etc/profile" file and Add the following line as per the version seta environment for Java
Use the root user to save the /etc/proflie or use gedit instead of vi .
The 'profile' file contains commands that ought to be run for login shells

boss@solaiv[]# sudovi /etc/profile #--insert JAVA_HOME

JAVA_HOME=/opt/jdk1.8.0_05
#--in PATH variable just append at the end of the line PATH=$PATH:$JAVA _HOME/bin
#--Append JAVA_HOME at end of the export statement export PATH
JAVA_HOME save the file using by pressing "Esc" key followed by :wq!

Step 3: Source the /etc/profile boss@solaiv[]# source /etc/profile Step 4: Update the java alternatives
By default OS will have a open jdk. Check by "java -version". You will be prompt "openJDK"

If you also have openjdk installed then you'll need to update the java alternatives:

If your system has more than one version of Java, configure which one your system causes by entering the following command in a terminal window

By default OS will have a open jdk. Check by "java -version". You will be prompt "Java HotSpot(TM) 64-Bit Server"

boss@solaiv[]# update-alternatives --install "/usr/bin/java" java "/opt/jdk1.8.0_05/bin/java" 1

boss@solaiv[]# update- alternatives --config java --type selection number: boss@solaiv[]# java -version
2)configure ssh

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this short tutorial). For our single-node setup of Hadoop, we

therefore need to configure SSH access to localhost

The need to create a Password-less SSH Key generation based authentication is so

that the master node can then login to slave nodes (and the secondary node) to start/stop them easily without any delays for authentication

If you skip this step, then have to provide password

Generate an SSH key for the user. Then Enable password-less SSH access to yo sudo apt-get install openssh-server
--You will be asked to enter password, root@solaiv[]# sshlocalhost root@solaiv[]# ssh-keygenroot@solaiv[]# ssh-copy-id -i localhost
--After above 2 steps, You will be connected without password, root@solaiv[]# sshlocalhost

root@solaiv[]# exit

3)Hadoop installation

Now Download Hadoop from the official Apache, preferably a stable release version of Hadoop 2.7.x and extract the contents of the Hadoop package to a location of your choice.

We chose location as "/opt/"

Step 1: Download the tar.gz file of latest version Hadoop(hadoop-2.7.x) from the official site .
Step 2: Extract(untar) the downloaded file from this commands to /opt/bigdata

root@solaiv[]# cd /opt
root@solaiv[/opt]# sudo tar xvpzf /home/itadmin/Downloads/hadoop-2.7.0.tar.gz root@solaiv[/opt]# cd hadoop-2.7.0/

Like java, update Hadop environment variable in /etc/profile

boss@solaiv[]# sudovi /etc/profile

#--insert HADOOP_PREFIX HADOOP_PREFIX=/opt/hadoop-2.7.0

#--in PATH variable just append at the end of the line PATH=$PATH:$HADOOP_PREFIX/bin

#--Append HADOOP_PREFIX at end of the export statement export PATH JAVA_HOME HADOOP_PREFIX

save the file using by pressing "Esc" key followed by :wq!
Step 3: Source the /etc/profile boss@solaiv[]# source /etc/profile Verify Hadoop installation boss@solaiv[]# cd $HADOOP_PREFIX boss@solaiv[]# bin/hadoop version Modify the Hadoop Configuration Files

Add the following properties in the various hadoop configuration files which is available under $HADOOP_PREFIX/etc/hadoop/
core-site.xml, hdfs-site.xml, mapred-site.xml & yarn-site.xml

Update Java, hadoop path to the Hadoop environment file boss@solaiv[]# cd $HADOOP_PREFIX/etc/hadoop

boss@solaiv[]# vi hadoop-env.sh
Paste following line at beginning of the file export JAVA_HOME=/usr/local/jdk1.8.0_05 export HADOOP_PREFIX=/opt/hadoop-2.7.0 Modify the core-site.xml
boss@solaiv[]# cd $HADOOP_PREFIX/etc/hadoopboss@solaiv[]# vi core- site.xml Paste following between <configuration> tags <configuration>

<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
Modify the hdfs-site.xml boss@solaiv[]# vi hdfs-site.xml
Paste following between <configuration> tags
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

YARN configuration - Single Node modify the mapred-site.xml boss@solaiv[]# cpmapred-site.xml.template mapred-site.xml boss@solaiv[]# vi mapred-site.xml
Paste following between <configuration> tags
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
Modiy yarn-site.xml boss@solaiv[]# vi yarn-site.xml
Paste following between <configuration> tags
<configuration>
<property><name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value></property>
</configuration>
Formatting the HDFS file-system via the NameNode

The first step to starting up your Hadoop installation is formatting the Hadoop files system which is implemented on top of the local file system of our "cluster" which includes only our local machine. We need to do this the first time you set up a Hadoop cluster.

Do not format a running Hadoop file system as you will lose all the data currently in the cluster (in HDFS)

root@solaiv[]# cd $HADOOP_PREFIX root@solaiv[]# bin/hadoopnamenode - format Start NameNode daemon and DataNode daemon: (port 50070) root@solaiv[]# sbin/start-dfs.sh

To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps Start ResourceManager daemon and NodeManager daemon: (port 8088) root@solaiv[]# sbin/start-yarn.sh
To stop the running process root@solaiv[]# sbin/stop-dfs.sh
To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps Start ResourceManager daemon and NodeManager daemon: (port 8088) root@solaiv[]# sbin/stop-yarn.sh

Procedure:

Update APT

test@cs-88:~$ sudo apt-get update

E: Unable to lock directory /var/lib/apt/lists/

Check back ground Process:

test@cs-88:~$ ps -ef |grep apt

root 2292 2285 0 11:11 ? 00:00:00 /bin/sh /etc/cron.daily/apt

root 2612 2292 0 11:30 ? 00:00:00 apt-get -qq -y update

root 2615 2612 0 11:30 ? 00:00:00 /usr/lib/apt/methods/http

root 2616 2612 0 11:30 ? 00:00:00 /usr/lib/apt/methods/http

root 2617 2612 0 11:30 ? 00:00:00 /usr/lib/apt/methods/http

root 2619 2612 0 11:30 ? 00:00:00 /usr/lib/apt/methods/gpgv

root 2627 2612 0 11:30 ? 00:00:01 /usr/lib/apt/methods/bzip2

test 2829 2813 0 11:36 pts/0 00:00:00 grep --color=auto apt

Kill Backgroud Process :

test@cs-88:~$ sudo kill -9 2292 2612 2615 2616 2617 2619 2627 2829

Updaet apt :

test@cs-88:~$ sudo apt-get update

git installation :

root@cs-88:~# sudo apt-get install git

Clone :

root@cs-88:~# git clone https://git.openstack.org/openstack-dev/devstack

root@cs-88:~# ls devstack

root@cs-88:~# cd devstack

root@cs-88:~/devstack# nano local.conf [[local|localre]]

HOST_IP=192.168.4.88

// FLOATING_RANGE=192.168.1.224/27 FIXED_RANGE=10.11.12.0/24 FIXED_NETWORK_SIZE=256 FLAT_RANGE=eth0 ADMIN_PASSWORD=linux DATABASE_PASSWORD=linux RABBIT_PASSWORD=linux SERVICE-

TOKEN=linux

Save Nono file:

control+x

Hadoop Installation;

stack@cs-88:~/Downloads$ sudo scp -r * /opt/ stack@cs-88:~/Downloads$ ls /opt/

test@cs-88:/opt$ ls

hadoop-2.7.0.tar.gz Hadoop Pseudo-Node.pdf HDFSCommands.pdf jdk-8u60-linux- x64

test@cs-88:/opt$ ls total 383412

drwxr-xr-x 2 root root 4096 May 6 15:41 ./

drwxr-xr-x 23 root root 4096 May 6 16:34 ../

-rw-r--r-- 1 root root 210343364 May 6 15:41 hadoop-2.7.0.tar.gz

-rw-r--r-- 1 root root 159315 May 6 15:41 Hadoop Pseudo-Node.pdf

-rw-r--r-- 1 root root 43496 May 6 15:41 HDFSCommands.pdf

-rw-r--r-- 1 root root 181238643 May 6 15:41 jdk-8u60-linux-x64.gz

-rw-r--r-- 1 root root 402723 May 6 15:41 mrsampledata(1).tar.gz

-rw-r--r-- 1 root root 402723 May 6 15:41 mrsampledata.tar.gz Change root user to test user:

test@cs-88:/opt$ sudo chown -Rh test:test /opt/ test@cs-88:/opt$ ll /* display list file with permission total 383412

drwxr-xr-x 2 test test 4096 May 6 15:41 ./

drwxr-xr-x 23 root root 4096 May 6 16:34 ../

-rw-r--r-- 1 test test 210343364 May 6 15:41 hadoop-2.7.0.tar.gz

-rw-r--r-- 1 test test 159315 May 6 15:41 Hadoop Pseudo-Node.pdf

-rw-r--r-- 1 test test 43496 May 6 15:41 HDFSCommands.pdf

-rw-r--r-- 1 test test 181238643 May 6 15:41 jdk-8u60-linux-x64.gz

-rw-r--r-- 1 test test 402723 May 6 15:41 mrsampledata(1).tar.gz

-rw-r--r-- 1 test test 402723 May 6 15:41 mrsampledata.tar.gz test@cs-88:/opt$

Unzip JAVA

test@cs-88:/opt$ tar -zxvf jdk-8u60-linux-x64.gz

test@cs-88:/opt$ cd jdk1.8.0_60 test@cs-88:/opt/jdk1.8.0_60$ pwd

/opt/jdk1.8.0_60 test@cs-88:/

Set profiloe for JAVA

test@cs-88:/opt/jdk1.8.0_60$ sudo nano /etc/profile

JAVA_HOME=/opt/jdk1.8.0_60 HADOOP_PREFIX=/opt/hadoop-2.7.0

PATH=$PATH:$JAVA_HOME/bin PATH=$PATH:$HADOOP_PREFIX/bin

export PATH JAVA_HOME HADOOP_PREFIX

Save: Control +x Press y

Press Enterkey

test@cs-88:/opt/jdk1.8.0_60$ cd .. test@cs-88:/opt$ pwd

/opt

test@cs-88:/opt$ Unzip hadoop file

test@cs-88:/opt$tar -zxvf hadoop-2.7.0.tar.gz test@cs-88:/opt$ source /etc/profile

Java Version

test@cs-88:/opt$ java -version

test@cs-88:/$ source /etc/profile test@cs-88:/$ java -version

java version "1.8.0_60"

Java(TM) SE Runtime Environment (build 1.8.0_60-b27)

Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)

SSH keygeneration

test@cs-88:/opt$ ssh-keygen Generating public/private rsa key pair.

Enter file in which to save the key (/home/test/.ssh/id_rsa): Created directory '/home/test/.ssh'.

Enter passphrase (empty for no passphrase): Enter same passphrase again:

Your identification has been saved in /home/test/.ssh/id_rsa.

Your public key has been saved in /home/test/.ssh/id_rsa.pub. The key fingerprint is:
c6:f4:33:42:4d:87:fb:3a:72:29:e9:5b:ce:ee:e9:e4 test@cs-88

The key's randomart image is:

+-- [RSA 2048] --- +

```
|         ...       |
|        o..        |
|       o ..        |
|      + ..         |
|      S +.         |
|      . . o.       |
|       .oo         |
|         +*=.      |
|       .oOE.       |
+…….+
```

test@cs-88:/opt$

configure ssh

test@cs-88:/opt$ ssh-copy-id -i localhost

\/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

test@cs-88:/opt$ sudo apt-get install openssh-server test@cs-88:/opt$ ssh-copy-id -i localhost

The authenticity of host 'localhost (127.0.0.1)' can't be established.

ECDSA key fingerprint is 67:12:a1:69:99:ea:b7:b7:96:b1:f5:4a:29:b5:d0:29. Are you sure you want to continue connecting (yes/no)? y

Please type 'yes' or 'no': yes

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that

are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

test@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'localhost'" and check to make sure that only the key(s) you wanted were added.

test@cs-88:/opt$ Configure Hadoop

Verify Hadoop installation

test@cs-88:/opt$ cd $HADOOP_PREFIX test@cs-88:/opt/hadoop-2.7.0$ pwd

/opt/hadoop-2.7.0

test@cs-88:/opt/hadoop-2.7.0$

test@cs-88:/opt/hadoop-2.7.0$ ls

bin include libexec

NOTICE.txt sbin etc lib

LICENSE.txt README.txt share

test@cs-88:/opt/hadoop-2.7.0$ bin/hadoop version Hadoop 2.7.0

Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r d4c8d4d4d203c934e8074b31289a28724c0842cf Compiled by jenkins on 2015-04-10T18:40Z

Compiled with protoc 2.5.0

From source with checksum a9e90912c37a35c3195d23951fd18f

This command was run using /opt/hadoop-2.7.0/share/hadoop/common/hadoop-common- 2.7.0.jar

test@cs-88:/opt/hadoop-2.7.0$

Update Java, hadoop path to the Hadoop environment file

test@cs-88:/opt/hadoop-2.7.0$ cd $HADOOP_PREFIX/etc/hadoop test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ pwd

/opt/hadoop-2.7.0/etc/hadoop

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano hadoop-env.sh type at last export JAVA_HOME=/opt/jdk1.8.0_60 export HADOOP_PREFIX=/opt/hadoop-2.7.0

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

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

</property>

</configuration>

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ cp mapred-site.xml.template mapred-site.xml test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano mapred-site.xml <configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

yet another resourcce negociated : yarn

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ nano yarn- site.xml <configuration>

<property>

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

</property>

</configuration>

test@cs-88:/opt/hadoop-2.7.0/etc/hadoop$ cd $HADOOP_PREFIX test@cs-88:/opt/hadoop-2.7.0$ bin/hadoop namenode -format DEPRECATED: Use of this script to execute hdfs command is deprecated. Instead use the hdfs command for it.

16/05/07 09:24:13 INFO namenode.NameNode: STARTUP_MSG:

/** images with txid >= 0

16/05/07 09:24:14 INFO util.ExitUtil: Exiting with status 0

16/05/07 09:24:14 INFO namenode.NameNode: SHUTDOWN_MSG:

/** SHUTDOWN_MSG: Shutting down NameNode at cs-88/127.0.1.1

**/

test@cs-88:/opt/hadoop-2.7.0$

test@cs-88:/opt/hadoop-2.7.0$ sbin/start-dfs.sh Starting namenodes on [localhost]

localhost: starting namenode, logging to /opt/hadoop-2.7.0/logs/hadoop-test- namenode-cs-88.out

localhost: starting datanode, logging to /opt/hadoop-2.7.0/logs/hadoop-test-datanode- cs-88.out

Starting secondary namenodes [0.0.0.0]

The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.

ECDSA key fingerprint is 67:12:a1:69:99:ea:b7:b7:96:b1:f5:4a:29:b5:d0:29. Are you sure you want to continue connecting (yes/no)? y

Please type 'yes' or 'no': yes

0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts. 0.0.0.0: starting secondarynamenode, logging to /opt/hadoop-2.7.0/logs/hadoop-test- secondarynamenode-cs-88.out

test@cs-88:/opt/hadoop-2.7.0$

jps- java machines proces status test@cs-88:/opt/hadoop-2.7.0$ jps 5667 DataNode

5508 NameNode

5863 SecondaryNameNode

5997 Jps

test@cs-88:/opt/hadoop-2.7.0$

test@cs-88:/opt/hadoop-2.7.0$ sbin/start- yarn.sh starting yarn daemons

starting resourcemanager, logging to /opt/hadoop-2.7.0/logs/yarn-test-resourcemanager- cs-88.out

localhost: starting nodemanager, logging to /opt/hadoop-2.7.0/logs/yarn-test- nodemanager-cs-88.out

test@cs-88:/opt/hadoop-2.7.0$

/* now getting 6 Services */

test@cs-88:/opt/hadoop-2.7.0$ jps 5667 DataNode

6084 ResourceManager

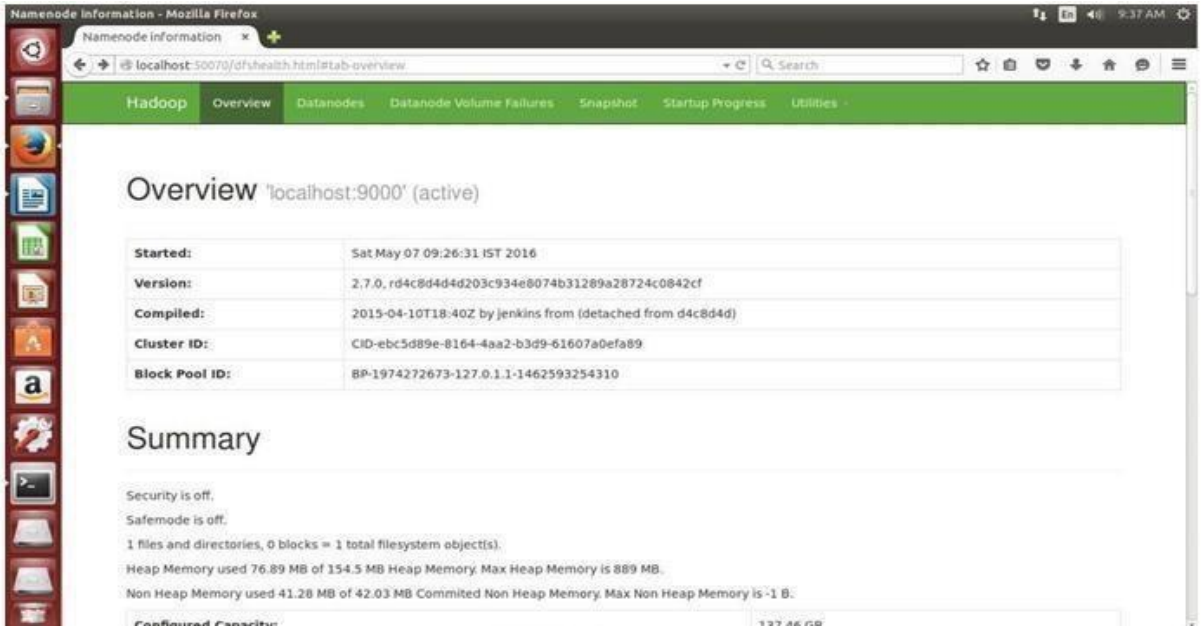
5508 NameNode

5863 SecondaryNameNode

6218 NodeManager

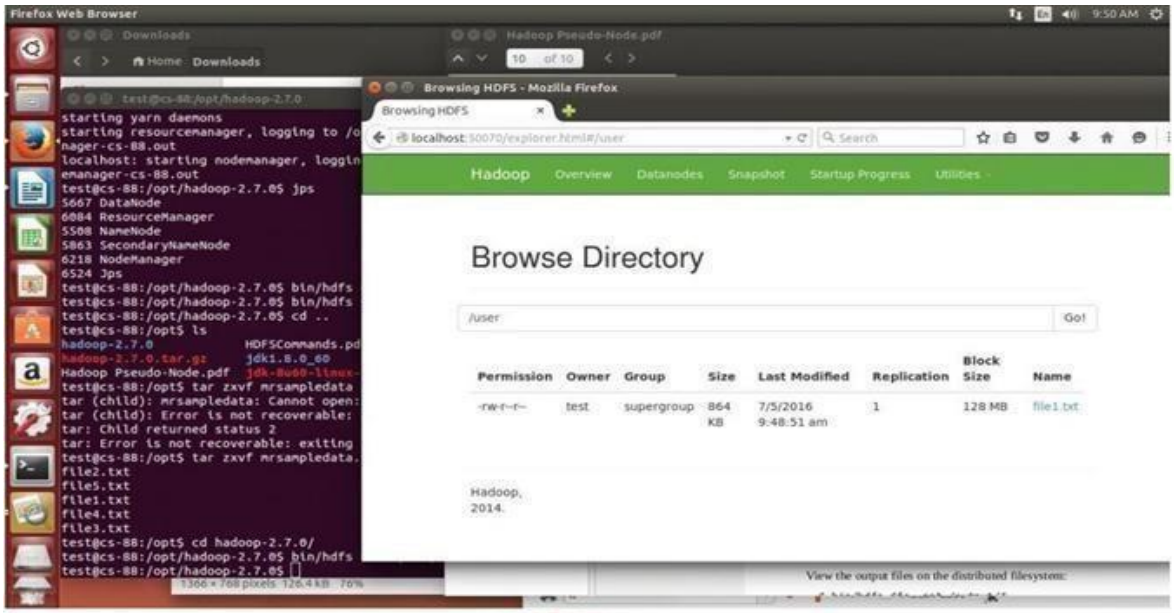
6524 Jps

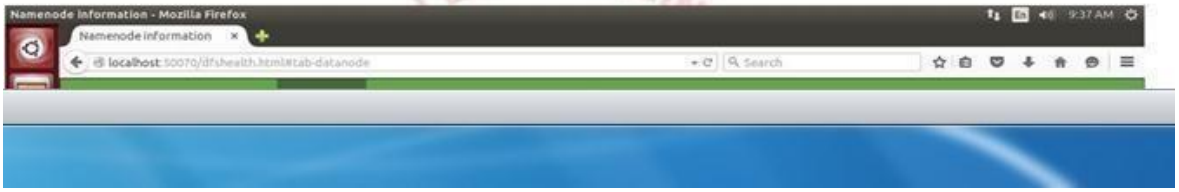
test@cs-88:/opt/hadoop-2.7.0$

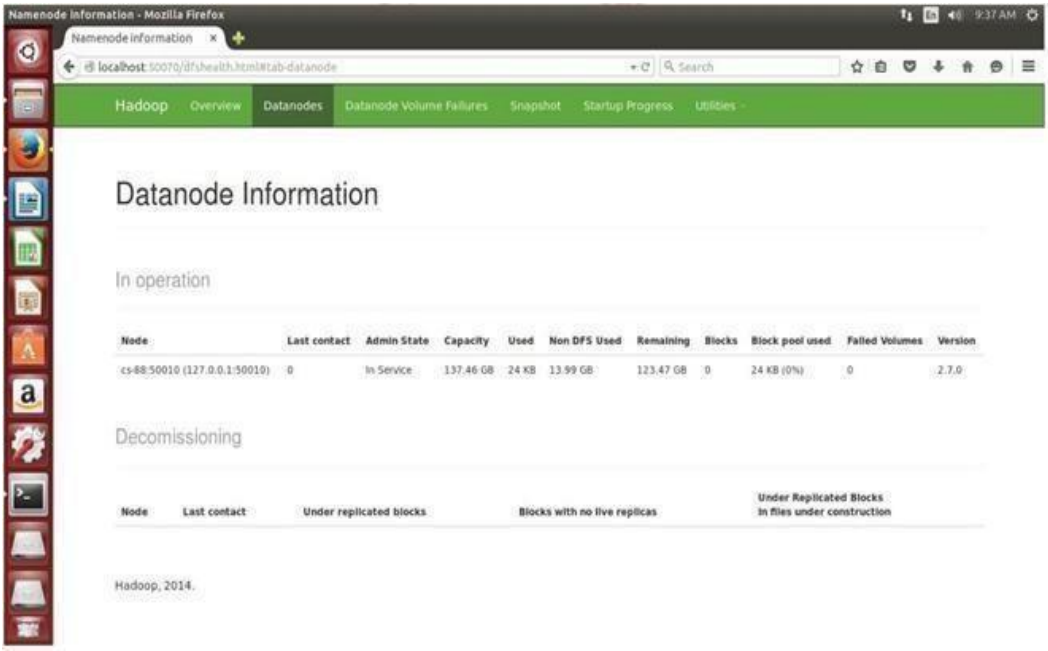


test@cs-88:/opt/hadoop-2.7.0$ bin/hdfs dfs -put /opt/file1.txt /user

test@cs-88:/opt/hadoop-2.7.0$



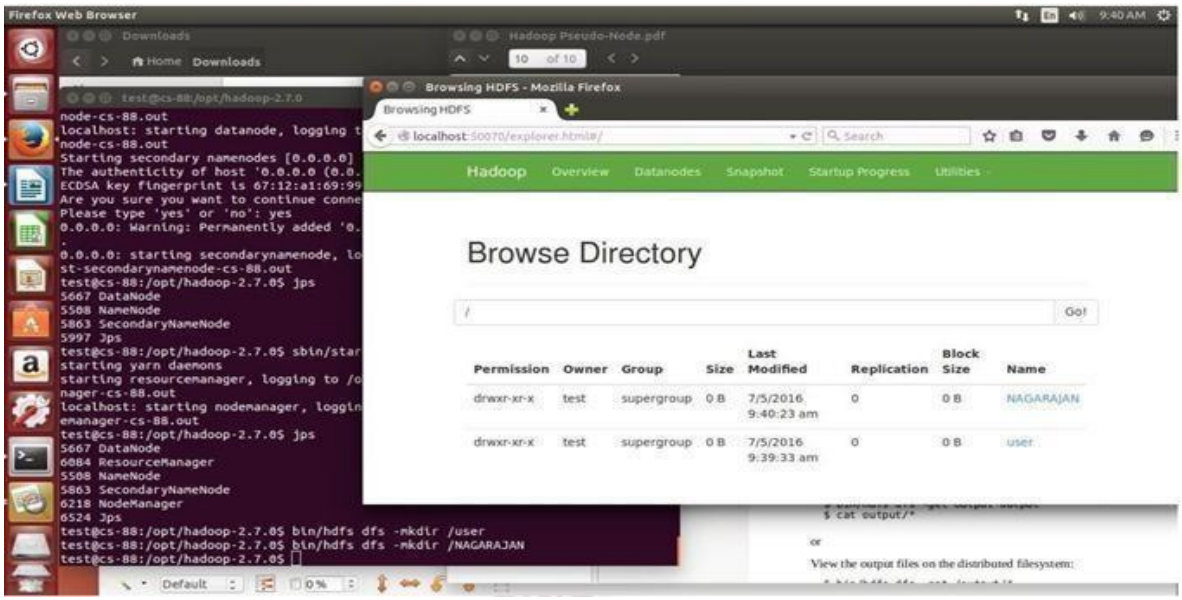




CREATING DIRECGTORY

test@cs-88:/opt/hadoop-2.7.0$ bin/hdfs dfs -mkdir /user

test@cs-88:/opt/hadoop-2.7.0$ bin/hdfs dfs -mkdir /exs



test@cs-88:/opt/hadoop-2.7.0$ cd .. test@cs-88:/opt$ ls

hadoop-2.7.0 HDFSCommands.pdf mrsampledata(1).tar.gz hadoop-2.7.0.tar.gz

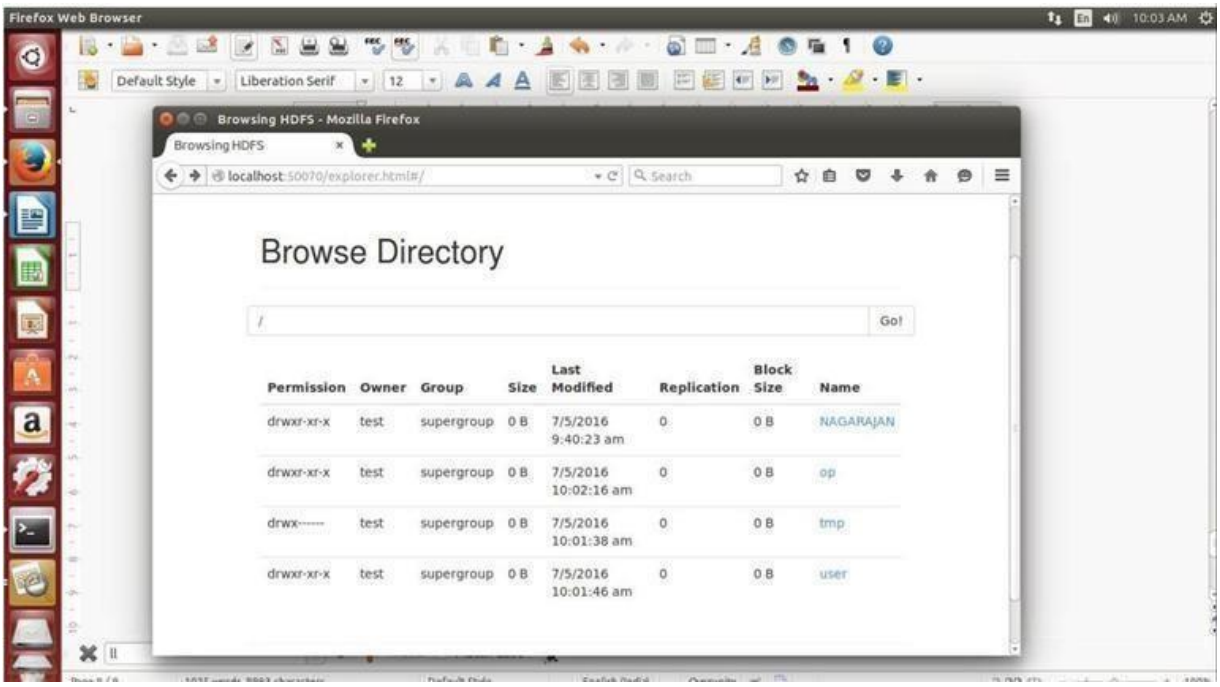
jdk1.8.0_60 mrsampledata.tar.gz Hadoop Pseudo-Node.pdf jdk-8u60-linux-x64.gz

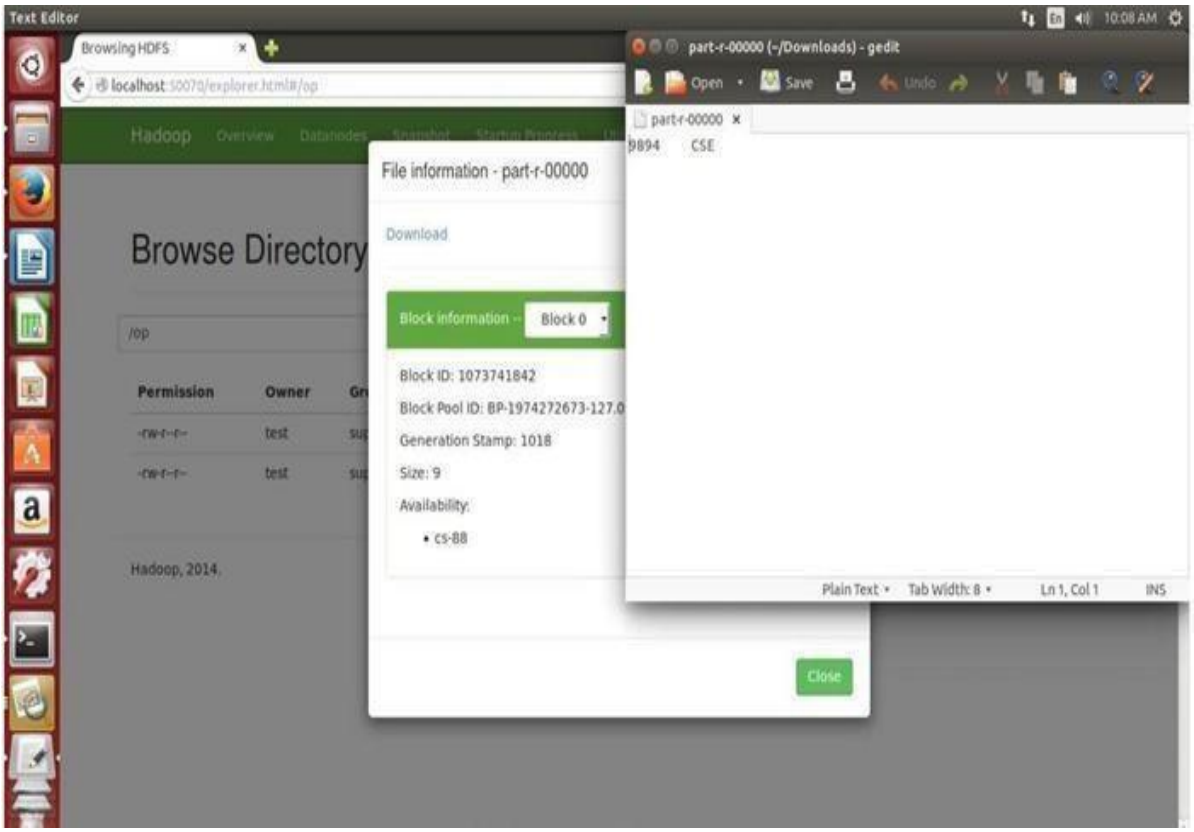
test@cs-88:/opt$ tar zxvf mrsampledata.tar.gz file2.txt

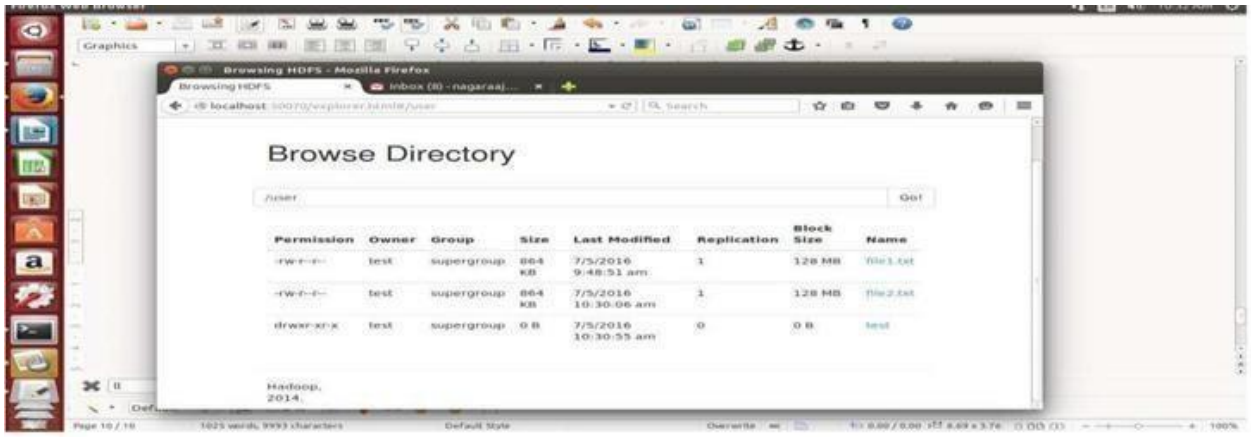
file5.txt file1.txt file4.txt file3.txt

test@cs-88:/opt$

test@cs-88:/opt/hadoop-2.7.0$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.0.jar grep /user/ /op '(CSE)'


Firefox Web Browser
Browsing HDFS - Mozilla Firefox
Browsing HDFS
localhost:50070/explorer.html#/
Browse Directory
/
Go!
Permission Owner Group Size Last Modified Replication Block Size Name
drwxr-xr-x test supergroup 0 B 7/5/2016 9:40:23 am 0 0 B NAGARAJAN
drwxr-xr-x test supergroup 0 B 7/5/2016 10:02:16 am 0 0 B op
drwx------ test supergroup 0 B 7/5/2016 10:01:38 am 0 0 B tmp
drwxr-xr-x test supergroup 0 B 7/5/2016 10:01:46 am 0 0 B user


Text Editor
Browsing HDFS
localhost:50070/explorer.html#/op
Hadoop
Browse Directory
/op
Permission Owner
-rw-r--r- test
-rw-r--r- test
Hadoop, 2014.
File information - part-r-00000
Download
Block information -- Block 0
Block ID: 1073741842
Block Pool ID: BP-1974272673-127.0
Generation Stamp: 1018
Size: 9
Availability:
• cs-88
Close
part-r-00000 (~/Downloads) - gedit
Open
Save
Undo
part-r-00000
9894 CSE
Plain Text
Tab Width: 8
Ln 1, Col 1
INS



Word count program to demonstrate the use of Map and Reduce tasks Procedure:

1. Format the path.
2. Start the dfs and check the no. of nodes running.
3. Start the yarn and check the no. of nodes running.
4. Open the browser and check whether the hadoop is installed correctly.
5. Add a file and check whether we can view the file.
6. Implement the grep command for the file added and see the result.
7. Implement the wordcount command for the file added and see the result.
8. After completing the process stop dfs and yarn properly.

Commands:

Install the hadoop cluster by using the commands

1.$sudochown –Rh gee.gee/opt/

2. $nano yarn –site.xml <configuration>
<property> <name>yarn.nodemanager.aux- services </name>
<value>mapreduce_shuffle</value> </property>

</configuration>

3. $cd $Hadoop _prefix
4. $bin / Hadoopnamenode_format
5. $s.bin /start_dfs.sh
6.jps

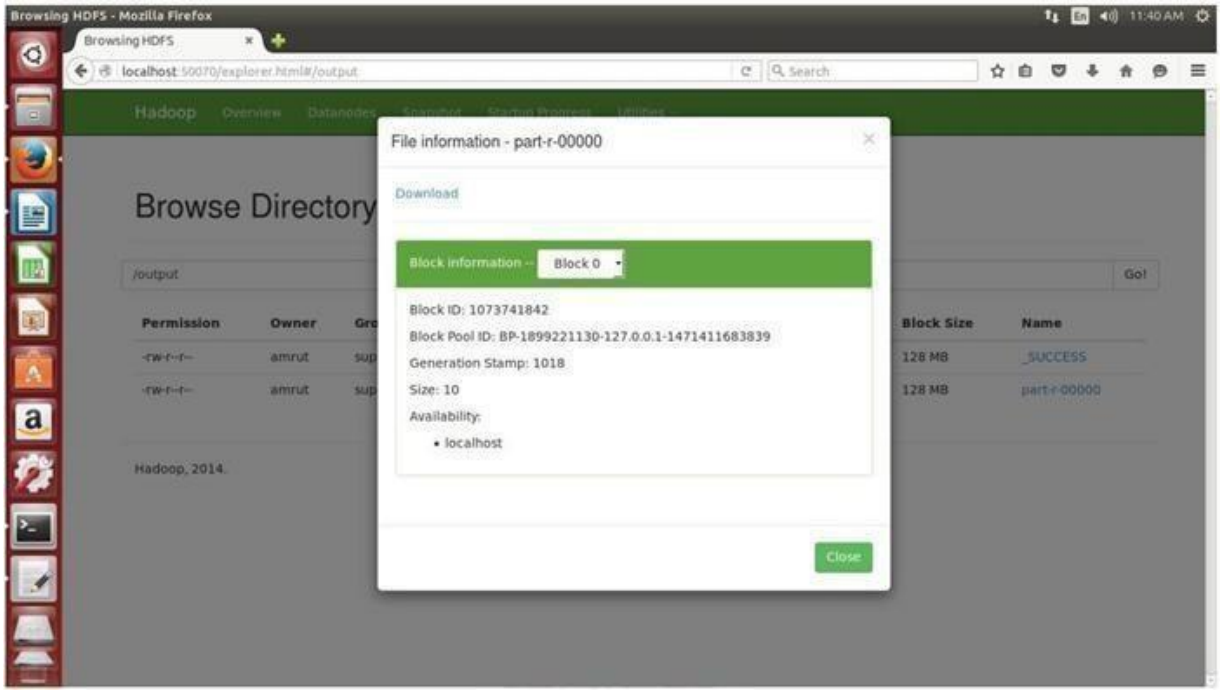
PROGRAM:

```
Package hadoop;
import java.util.* ;
import java.io.IOException;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.* ;
import org.apache.hadoop.io.* ;
import org.apache.hadoop.m apred.* ;
```

```
import org.apache.hadoop.util.* ;
public class ProcessUnits
{
public static class E_EMapper extends
MapReduceBaseimplements
Mapper<LongWritable Text,
Text, IntWritable>
{
public void m ap(LongWritable key,
Text value, OutputCollector<Text,
IntWritable> output, Reporter
reporter) throws IOException
{
String line = value.toString(); String
lasttoken = null; StringTokenizer s =
new
StringTokenizer(line,"\t"); String year
=                s.nextToken();
while(s.hasMoreTokens()) {

lasttoken=s.nextToken();
}
intavgprice =
Integer.parseInt(lasttoken);
output.collect(new Text(year), new
IntWritable(avgprice)); }
}
public static class E_EReduce extends
MapReduceBaseimplements
Reducer< Text, IntWritable, Text,
IntWritable> {
public void reduce( Text key, Iterator
<IntWritable> values,
OutputCollector<Text, IntWritable>
output, Reporter reporter) throws
IOException
{
int m axavg=30;
intval=Integer.MIN_V
ALUE;while
(values.hasNext())
{
if((val=values.next().get())>m axavg)
{
output.collect(key, new IntWritable(val));
```

```
}
}
}
}
public static void m ain(String args[])throws Exception
{
JobConfconf = new JobConf(Eleunits.class);
conf.setJobNam e("m ax_eletricityunits");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass(E_EMapper.class);
conf.setCombinerClass(E_EReduce.class);
conf.setReducerClass(E_EReduce.class);
conf.setInputForm at(TextInputFormat.class);
conf.setOutputForm at(TextOutputFormat.class);
FileInputFormat.setInputPaths(conf, new
Path(args[0]));
FileOutputFormat.setOutputPath(conf, new
Path(args[1]));JobClient.runJob(conf);
}
}
```

OUTPUT:



Result:

Thus, the map & reduce is successfully done using word count program using one node Hadoop cluster.

Ex. No: 8 Date:	CREATING AND EXECUTING YOUR FIRST CONTAINER USING DOCKER

Aim:

To introduce Docker containers through the process of creating and executing their first container.

Procedure:

The specific procedure was to:

1. Install Docker on participant machines.
2. Pull the hello-world image from Docker Hub.
3. Execute the hello-world container.
4. Explore basic Docker commands for container management.

1. Installation of Docker on Participant Machines

Step 1: Download Docker Desktop

- visit the official Docker website.
- Provide the link to download Docker Desktop suitable for their operating system (Windows/Mac/Linux).

Step 2: Installation Process

- Students should run the downloaded installer.
- Follow the on-screen instructions provided by the installer.
- For Windows and Mac, this typically involves accepting the license agreement, choosing installation options, and entering administrative credentials if prompted.

Step 3: Docker Setup

- Upon completion of the installation, Students should launch Docker Desktop.
- The setup may involve creating a Docker account or signing in, depending on the version.

Step 4: Verify Installation

- Open a terminal or Command Prompt.
- Run the command **docker --version** to confirm the successful installation of Docker.
- Additionally, running **docker run hello-world** can verify Docker's ability to pull and run containers.

Step 5: Troubleshooting (if necessary)

- Provide guidance on common installation issues students might encounter.
- Resources like Docker's official documentation or forums can be recommended for troubleshooting assistance.

2. Pull the hello-world image from Docker Hub.

- Open a terminal or Command Prompt.
- Execute the command: **docker pull hello-world**

```
PS C:\Users\NPRCET> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
```

3. **Execute the hello-world container**
   - Execute the hello-world container using the docker run command:

     docker run hello-world

```
PS C:\Users\NPRCET> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

4. **Interpret the Output**

   After running the docker run hello-world command, you should see output similar to the following**:**

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
...
```

5. **Understand the Output**
   - The output confirms that your Docker installation is functional.
   - It demonstrates that Docker successfully pulled the hello-world image and created a container based on it.

- The container executed a simple program that displayed the message.

6. **Verify Container Status (Optional)**
   - To verify the status of the container (which should now be stopped), you can use the **docker ps -a** command

```
PS C:\Users\NPRCET> docker ps -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS                      PORTS                NAMES
5eb921c35324   hello-world                    "/hello"                 5 minutes ago    Exited (0) 5 minutes ago                         pensive_mestorf
71cbcbe0f660   docker/getting-started:latest  "/docker-entrypoint.…"   2 weeks ago      Exited (255) 5 days ago     80/tcp               charming_hofstadter
735dcdbcc5f5   hello-world:latest             "/hello"                 2 weeks ago      Exited (0) 2 weeks ago                           rytyj
b9a94b35ad26   hello-world                    "/hello"                 2 weeks ago      Exited (0) 2 weeks ago                           pensive_cray
71d7b0e5453f   docker/getting-started         "/docker-entrypoint.…"   2 weeks ago      Exited (255) 5 days ago     0.0.0.0:80->80/tcp   determined_wilson
```

7. **Docker Commands**
   - **docker ps**

   Lists all currently running containers.
   - **docker ps -a**

   Lists all containers, including stopped ones.
   - **docker images**

   Lists all available images on your local machine.

```
PS C:\Users\NPRCET> docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
hello-world             latest    9c7a54a9a43c   6 months ago    13.3kB
docker/getting-started  latest    3e4394f6b72f   10 months ago   47MB
```

   - **docker stop**

   Stops a running container.
   - **docker start**

   Starts a stopped container.
   - **docker restart**

   Stops and then starts a container.
   - **docker rm**

   Removes one or more containers.
   - **docker rmi**

   Removes one or more images.

- **docker exec**

Runs a command inside a running container.

- **docker logs**

Retrieves the logs of a container.

- **docker inspect**

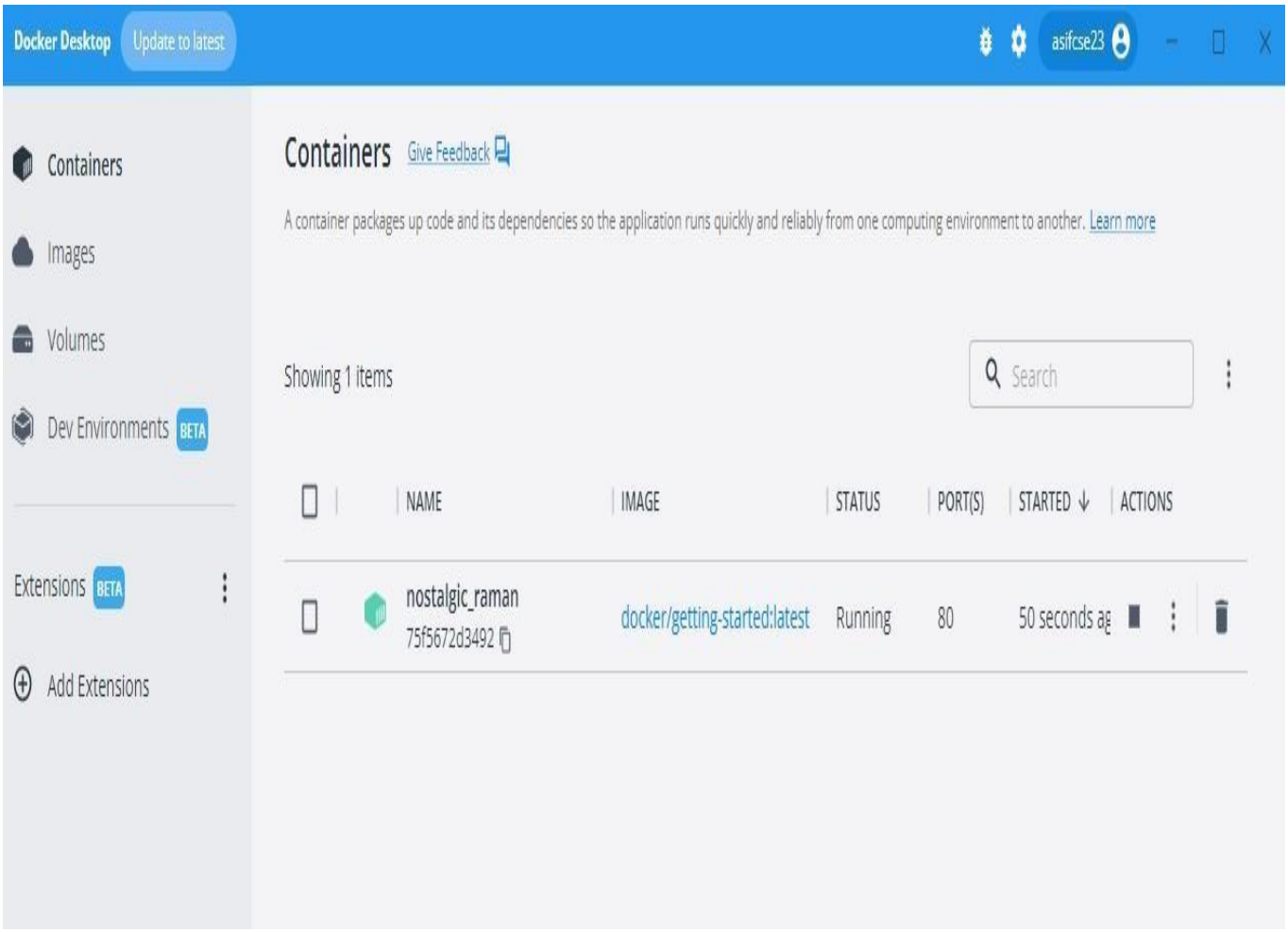
Displays detailed information on Docker objects (containers, images, networks, etc.).

- **docker build**

Builds an image from a Dockerfile.

- **docker-compose**

Manages multi-container Docker applications.



Result:

Thus, Docker containers through the process of creating and executing were completed.

Ex. No: 9 Date:	RUN A CONTAINER FROM DOCKER HUB

Aim:

To run a container from Docker Hub.

Procedure:

Step 1: Sign up for a Docker account

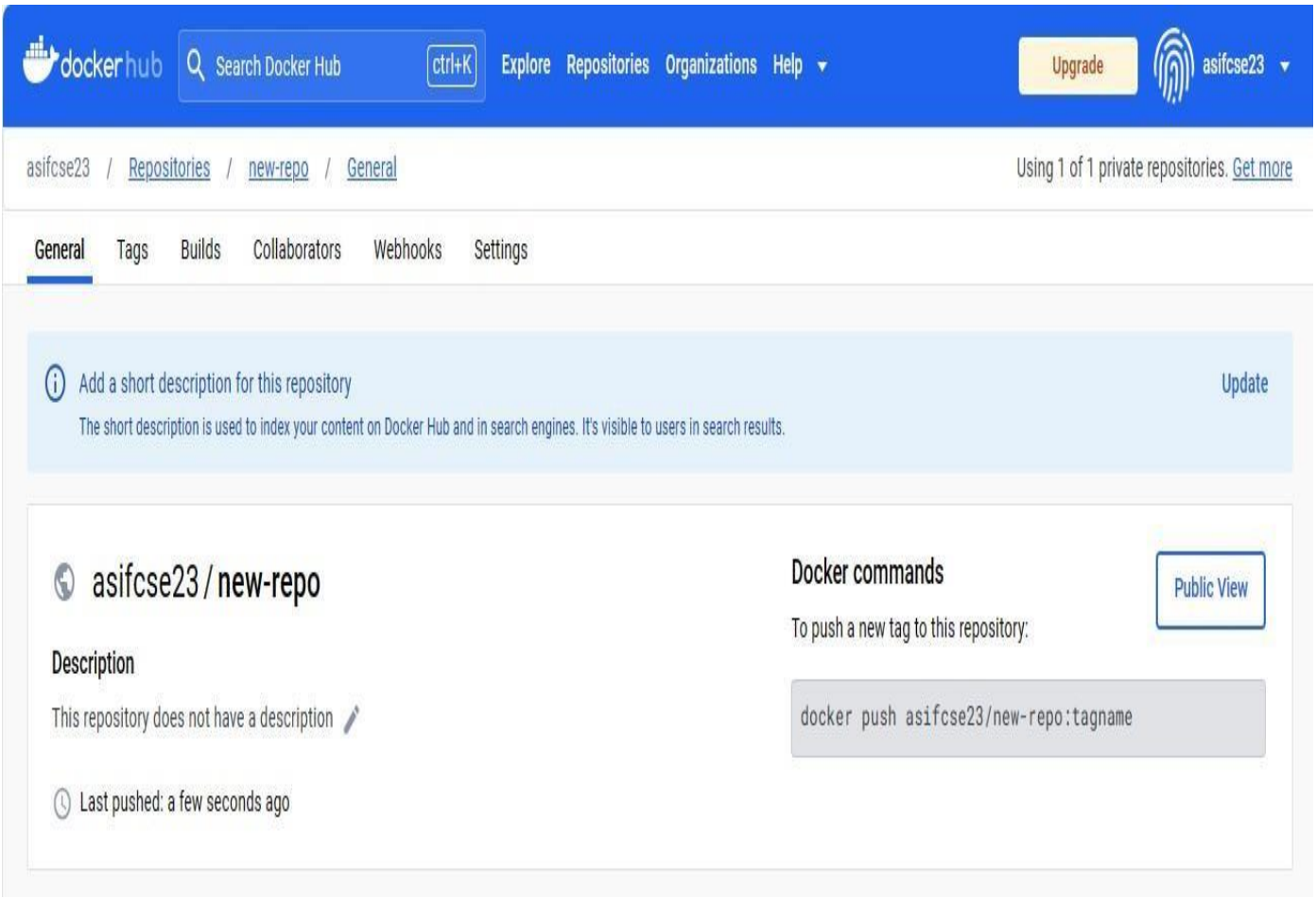
- Start by creating a Docker ID.
- A Docker ID grants you access to Docker Hub repositories and lets you explore available images from the community and verified publishers. You also need a Docker ID to share images on Docker Hub.

Step 2: Create your first repository

- To create a repository:
- Sign in to Docker Hub.
- Select Create a Repository on the Docker Hub welcome page.
- Name it <your-username>/my-private-repo.
- Set the visibility to Private.
- Select Create.
- You've created your first repository.

Step 3: Download and install Docker Desktop

- You need to download Docker Desktop to build, push, and pull container images.
- Download and install Docker Desktop.
- Sign in to Docker Desktop using the Docker ID you created in step one.



Step 4: Pull and run a container image from Docker Hub

- In your terminal, run **docker pull hello-world** to pull the image from Docker Hub. You should see output similar to:

```
PS C:\Users\NPRCET> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
```

- Run **docker run hello-world** to run the image locally. You should see output similar to:

```
PS C:\Users\NPRCET> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Step 5: Build and push a container image to Docker Hub from your computer

- Start by creating a Dockerfile to specify your application as shown below:

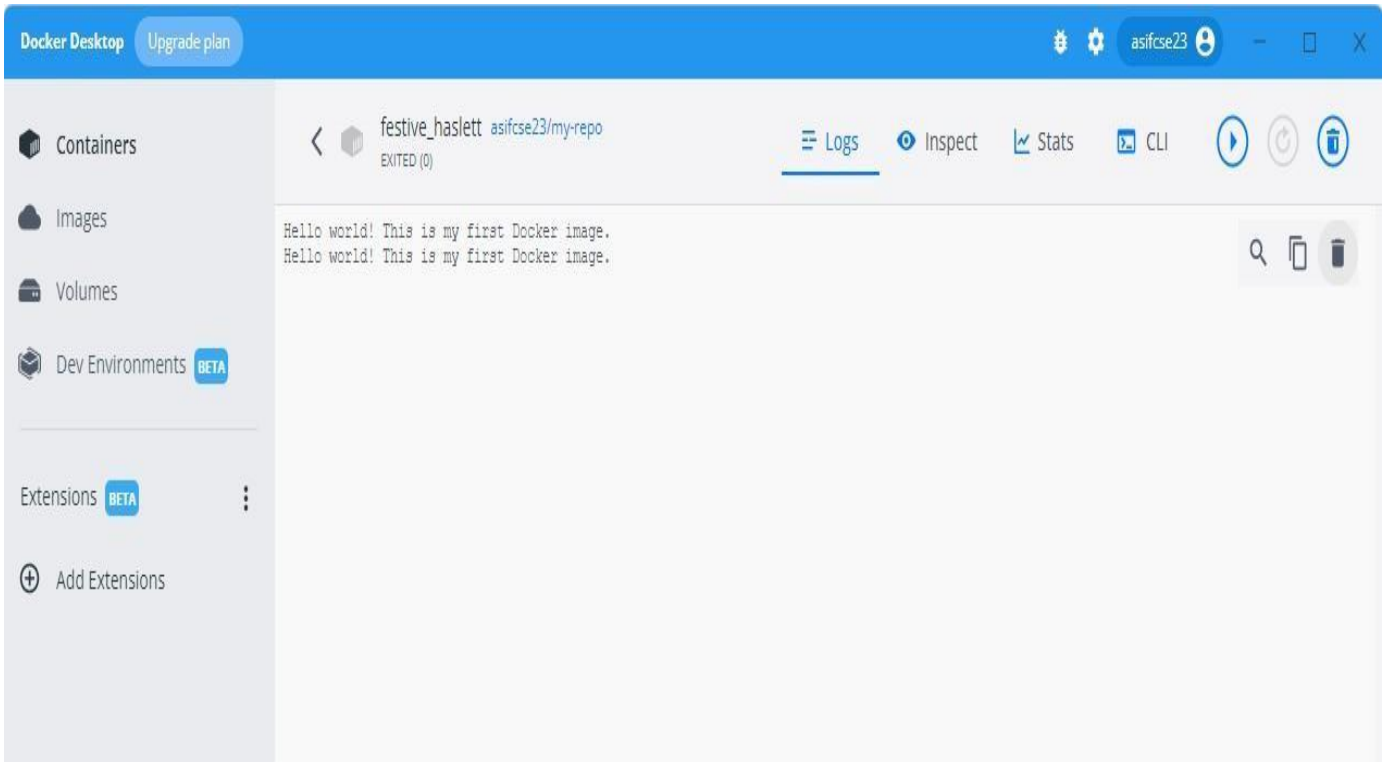
  ```
  # syntax=docker/dockerfile:1
  FROM busybox
  CMD echo "Hello world! This is my first Docker image."
  ```

- Run **docker build -t <your_username>/my-private-repo .** to build your Docker image.

```
PS C:\Users\CETLAB> docker build -t asifcse23/my-repo .
[+] Building 3.8s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 135B                                                                                0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => resolve image config for docker.io/docker/dockerfile:1                                                          2.0s
 => [auth] docker/dockerfile:pull token for registry-1.docker.io                                                    0.0s
 => CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50032edf31be0021   0.0s
 => [internal] load build definition from Dockerfile                                                                0.0s
 => [internal] load metadata for docker.io/library/busybox:latest                                                   1.5s
 => [auth] library/busybox:pull token for registry-1.docker.io                                                      0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => CACHED [1/1] FROM docker.io/library/busybox:latest@sha256:3fbc632167424a6d997e74f52b878d7cc478225cffac6bc977eedfe51c7f4e79   0.0s
 => exporting to image                                                                                              0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:085a2acac1c48ad82959c4ba5f40a774071927954 4dc3a3f4f31fdab7397ade7                       0.0s
 => => naming to docker.io/asifcse23/my-repo                                                                        0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

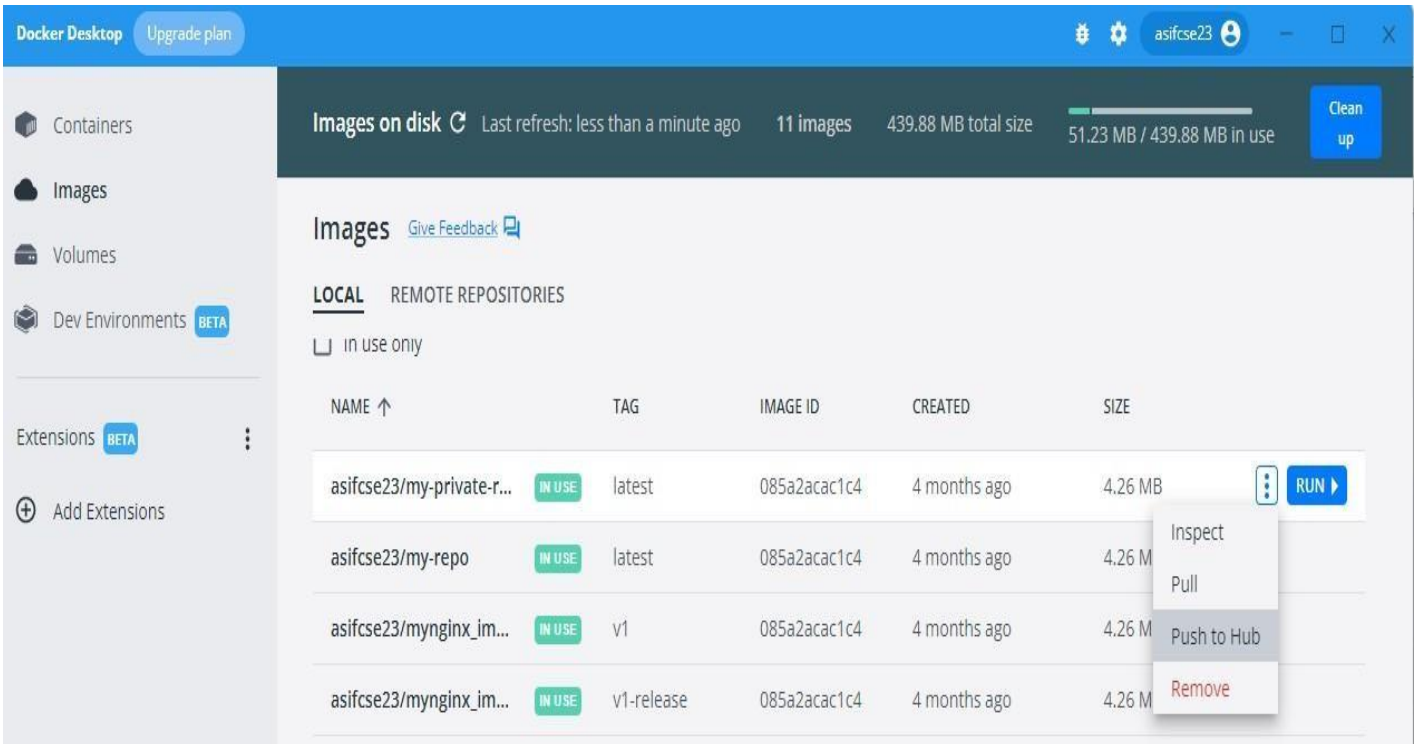
- Run **docker run <your_username>/my-private-repo** to test your Docker image locally.

```
PS C:\Users\CETLAB> docker run asifcse23/my-repo
Hello world! This is my first Docker image.
```

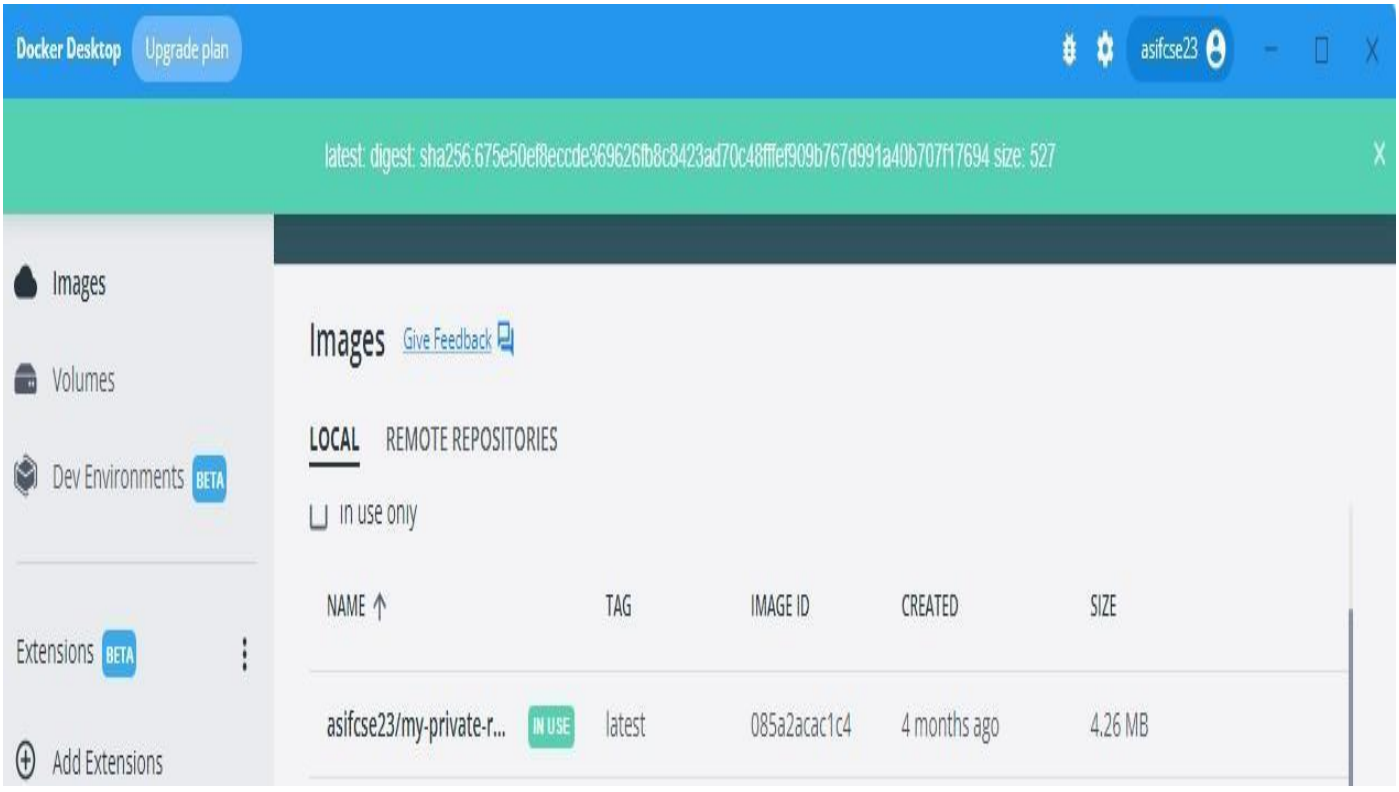


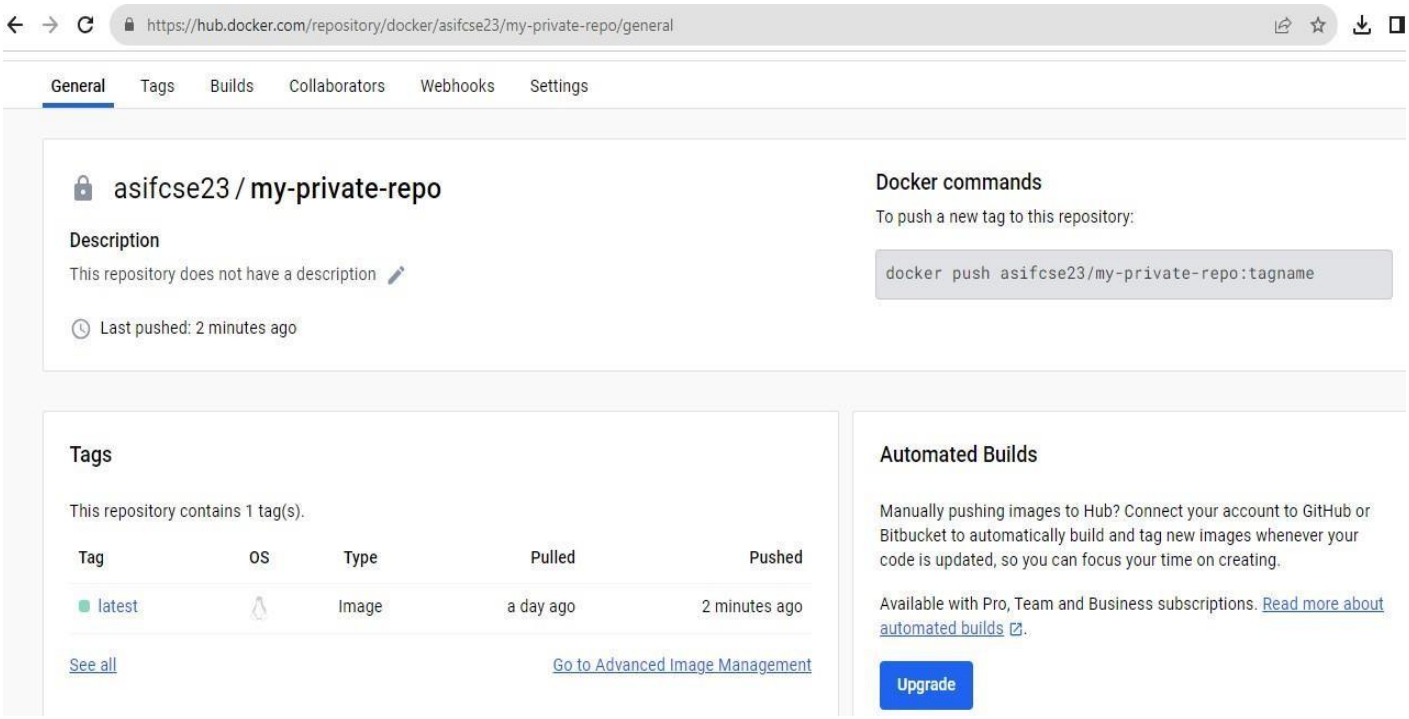
- Run **docker push <your_username>/my-private-repo** to push your Docker image to Docker Hub.

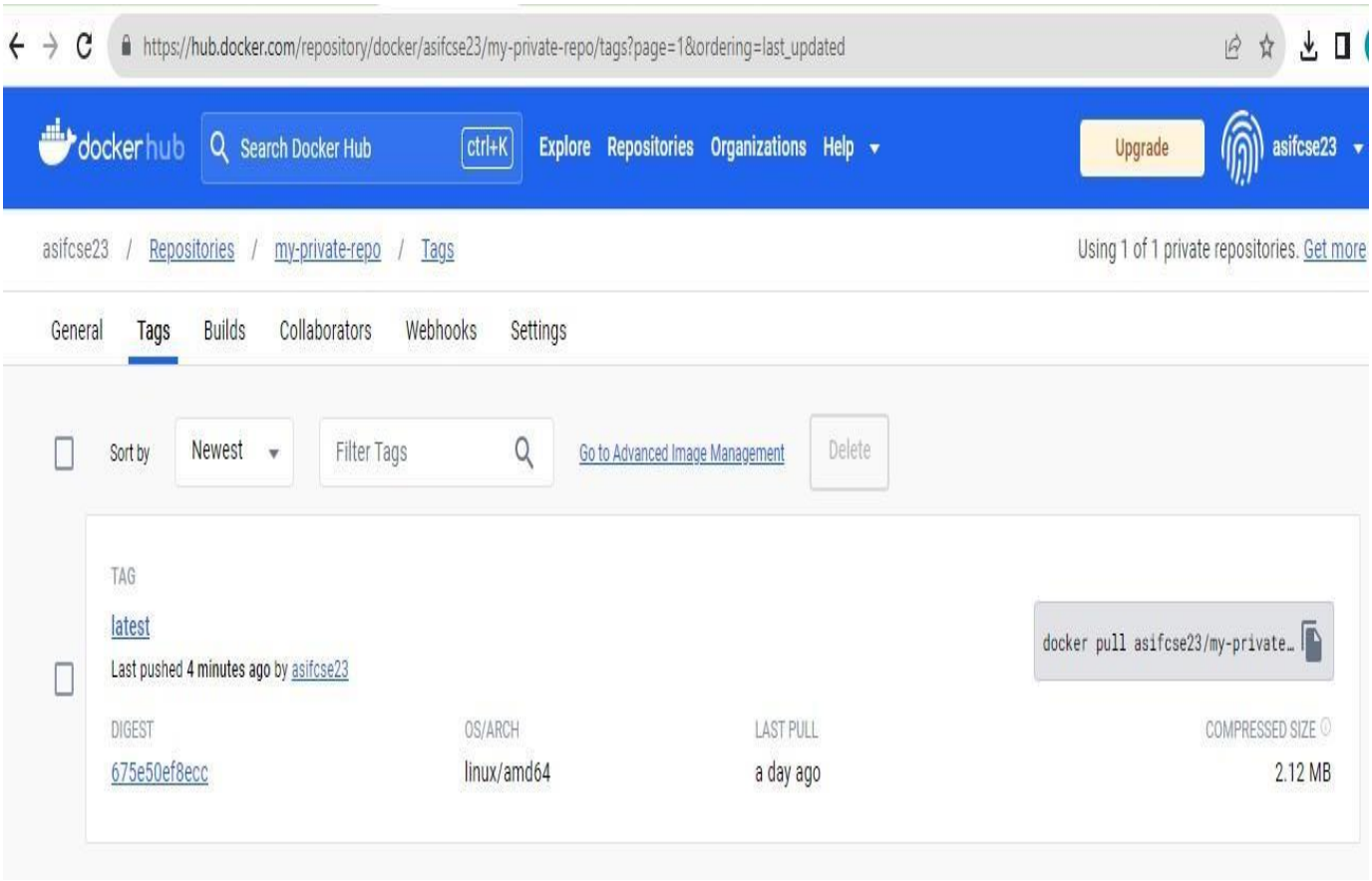
```
PS C:\Users\CETLAB> docker push asifcse23/my-repo
Using default tag: latest
The push refers to repository [docker.io/asifcse23/my-repo]
3d24ee258efc: Mounted from asifcse23/mynginx_image1
latest: digest: sha256:675e50ef8eccde369626fb8c8423ad70c48fffef909b767d991a40b707f17694 size: 527
```

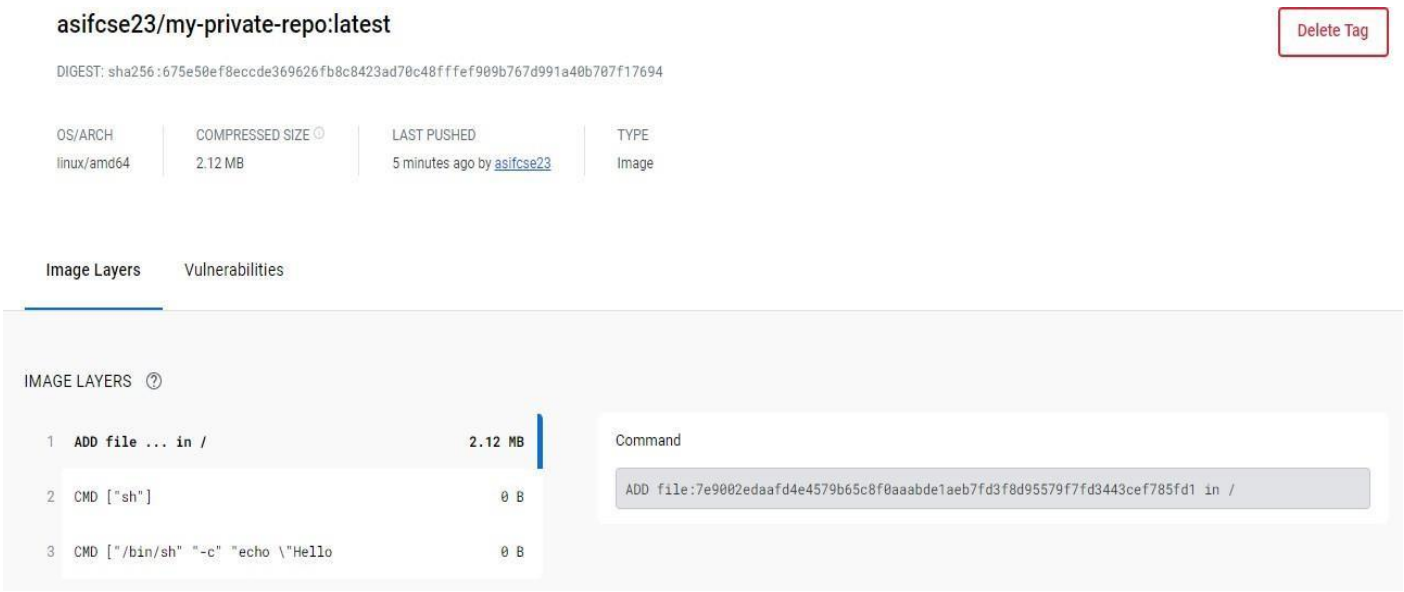


- Successfully pushed the latest image to the hub.




https://hub.docker.com/repository/docker/asifcse23/my-private-repo/general
General
Tags
Builds
Collaborators
Webhooks
Settings
asifcse23 / my-private-repo
Description
This repository does not have a description
Last pushed: 2 minutes ago
Docker commands
To push a new tag to this repository:
docker push asifcse23/my-private-repo:tagname
Tags
This repository contains 1 tag(s).
Tag
OS
Type
Pulled
Pushed
latest
Image
a day ago
2 minutes ago
See all
Go to Advanced Image Management
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds.
Upgrade


https://hub.docker.com/repository/docker/asifcse23/my-private-repo/tags?page=1&ordering=last_updated
docker hub
Search Docker Hub
ctrl+K
Explore
Repositories
Organizations
Help
Upgrade
asifcse23
asifcse23 / Repositories / my-private-repo / Tags
Using 1 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings
Sort by
Newest
Filter Tags
Go to Advanced Image Management
Delete
TAG
latest
Last pushed 4 minutes ago by asifcse23
docker pull asifcse23/my-private…
DIGEST
675e50ef8ecc
OS/ARCH
linux/amd64
LAST PULL
a day ago
COMPRESSED SIZE
2.12 MB



Result:

Thus, the run a container from Docker Hub were completed.

ADDITIONAL PROGRAMS

Ex. No: 10 Date:	CLOUD FOR HPC AND HTC

Aim:

To study the concept of Cloud HPC and HTC

High performance Computing

It is the use of parallel processing for running advanced application programs efficiently, relatives, and quickly. The term applies especially is a system that function above a teraflop (1012) (floating opm per second). The term High-performance computing is occasionally used as a synonym for supercomputing. Although technically a supercomputer is a system that performs at or near currently highest operational rate for computers. Some supercomputers work at more than a petaflop (1012) floating points opm per second. The most common HPC system all scientific engineers & academic institutions. Some Government agencies particularly military are also relying on APC for complex applications.

High-performance Computers :

Processors, memory, disks, and OS are elements of high-performance computers of interest to small & medium size businesses today are really clusters of computers. Each individual computer in a commonly configured small cluster has between one and four processors and today „s processors typically are from 2 to 4 crores, HPC people often referred to individual computers in a cluster as nodes. A cluster of interest to a small business could have as few as 4 nodes on 16 crores. Common cluster size in many businesses is between 16 & 64 crores or from 64 to 256 crores. The main reason to use this is that in its individual node can work together to solve a problem larger than any one computer can easily solve. These nodes are so connected that they can communicate with each other in order to produce some meaningful work.

There are two popular HPC‟s software i. e, Linux, and windows. Most of installations are in Linux because of its supercomputer but one can use it with his / her requirements.

High-throughput computing

The HTC community is also concerned with robustness and reliability of jobs over a long-time scale. That is, being able to create a reliable system from unreliable components. This research is similar to transaction processing, but at a much larger and distributed scale.

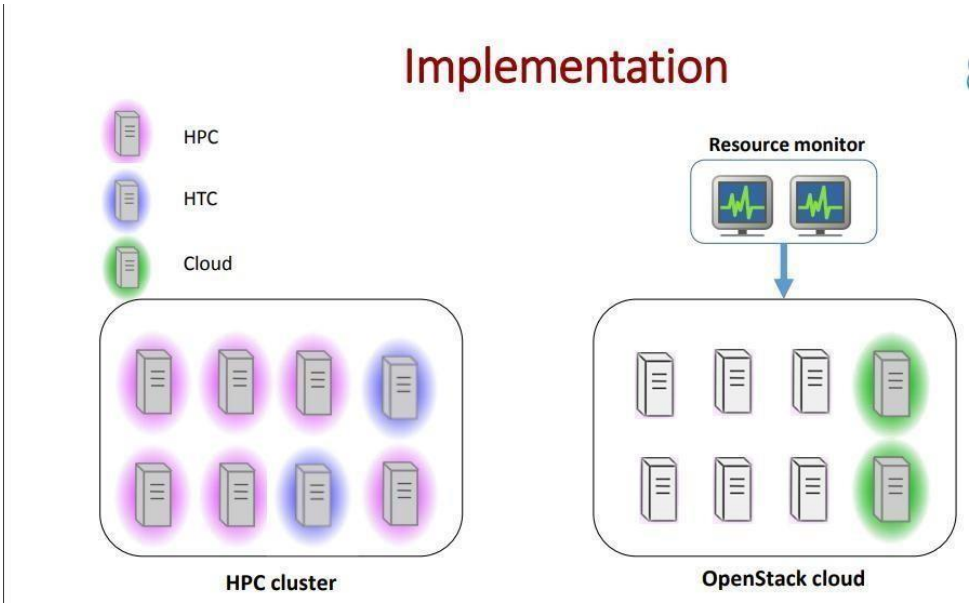
Some HTC systems, such as HTCondor and PBS, can run tasks on opportunistic resources. It is a difficult problem, however, to operate in this environment. On one hand the system needs to provide a reliable operating environment for the user's jobs, but at the same time the system must not compromise the integrity of the execute node and allow the owner to always have full control of their resources. Here are many differences between high-throughput computing, high-performance computing (HPC), and many-task computing (MTC).

HPC tasks are characterized as needing large amounts of computing power for short periods of time, whereas HTC tasks also require large amounts of computing, but for much longer times (months and years, rather than hours and days).[1] HPC environments are often measured in terms of FLOPS.

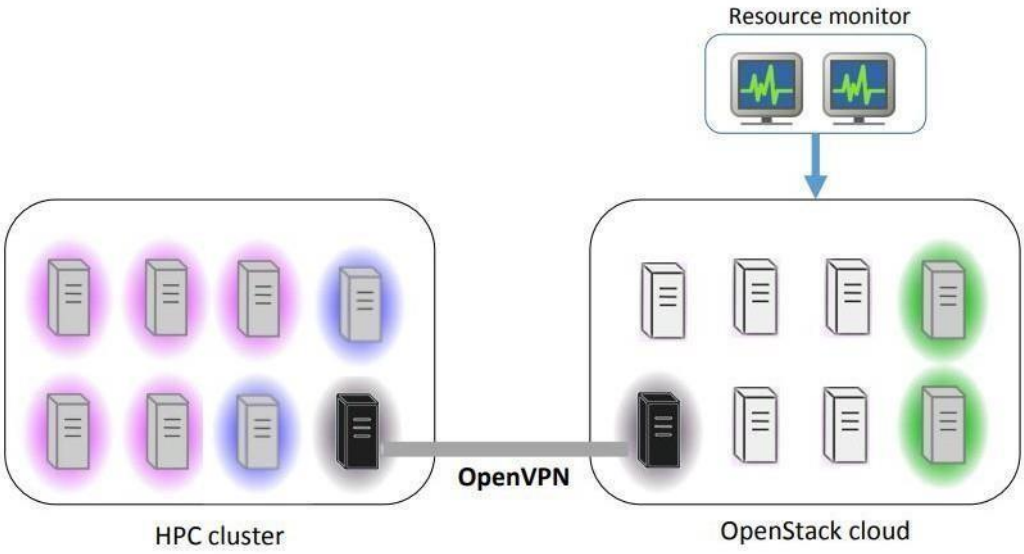
The HTC community, however, is not concerned about operations per second, but rather operations per month or per year. Therefore, the HTC field is more interested in how many jobs can be completed over a long period of time instead of how fast.

As an alternative definition, the European Grid Infrastructure defines HTC as "a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks",[2] while HPC systems tend to focus on tightly coupled parallel jobs, and as such they must execute within a particular site with low-latency interconnects. Conversely, HTC systems are independent, sequential jobs that can be individually scheduled on many different computing resources across multiple administrative boundaries. HTC systems achieve this using various grid computing technologies and techniques.

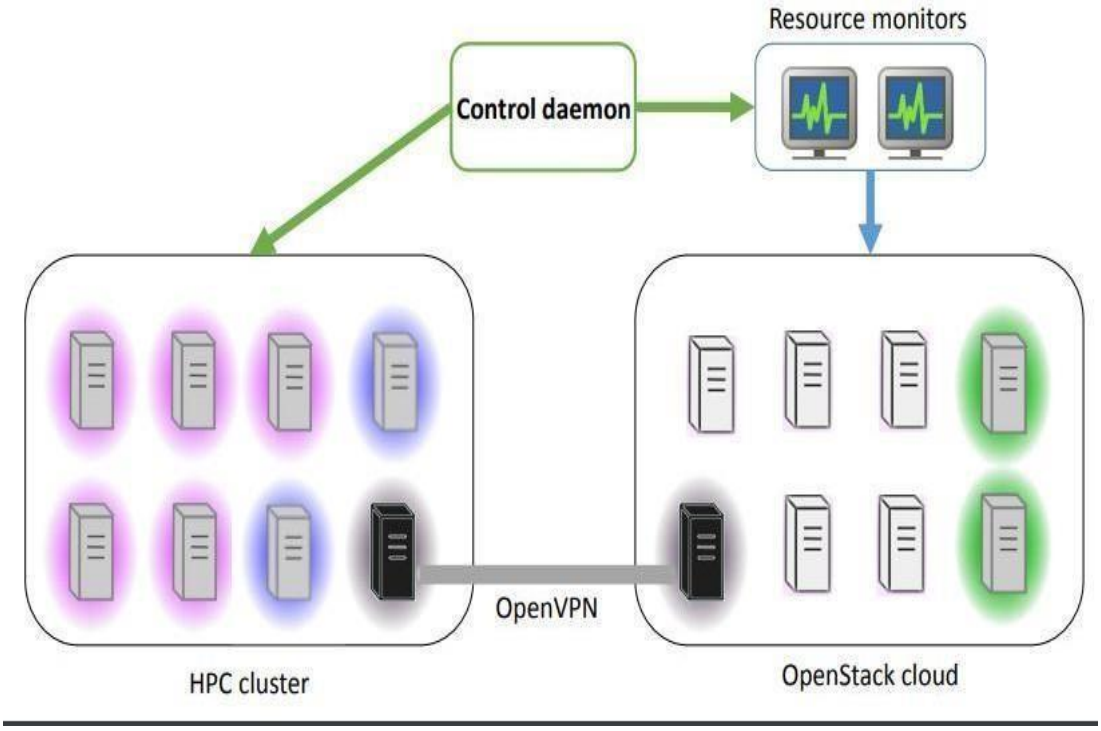
MTC aims to bridge the gap between HTC and HPC. MTC is reminiscent of HTC, but it differs in the emphasis of using many computing resources over short periods of time to accomplish many computational tasks (i.e. including both dependent and independent tasks), where the primary metrics are measured in seconds (e.g. FLOPS, tasks/s, MB/s I/O rates), as opposed to operations (e.g. jobs) per month. MTC denotes high-performance computations comprising multiple distinct activities, coupled via file system operations.

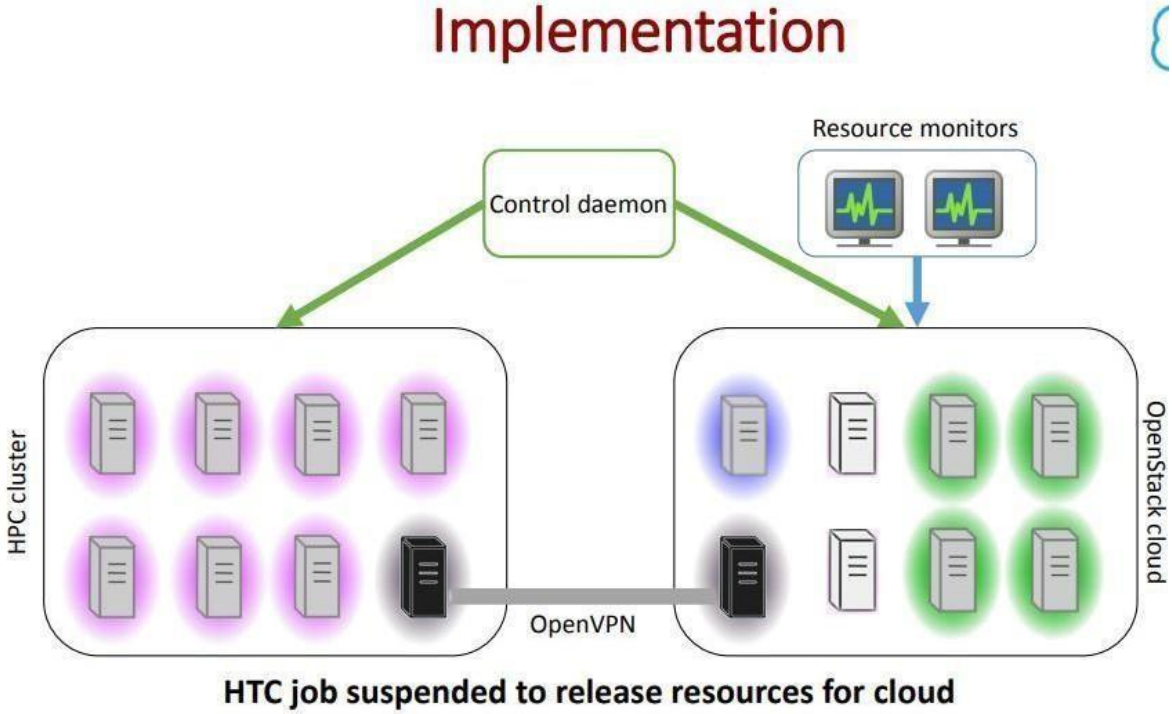


Implementation



Implementation





Modifications to Slurm Slurm – A workload manager for HPC cluster

• Manages the resource and job scheduling

• Marks a node DOWN and removes the jobs for an unreachable node

• Does the same for a suspended virtual node Modified Slurm to manage the suspended node and keep the job states intact

Result:

Thus, the Concept of HTC and HPC were studied successfully.

Ex. No :11 Date:	LAUNCHING OF VIRTUAL MACHINE USING TRY STACK

Aim:

To install the KVM and Open stack in Ubuntu 14.04 version and creation of virtual machine.

Mandatory prerequisite:

1. Linux 64 bit Operating System (The commands mentioned are for Ubuntu Linux Operating System latest version).

Installing KVM (Hypervisor for Virtualization)

1. Check if the Virtualization flag is enabled in BIOS Run the command in terminal

   egrep -c '(vmx|svm)' /proc/cpuinfo

   If the result is any value higher than 0, then virtualization is enabled.

   If the value is 0, then in BIOS enable Virtualization – Consult system administrator for this step.

2. To check if your OS is 64 bit,

   Run the command in terminal

   uname -m

   If the result is x86_64, it means that your Operating system is 64 bit Operating system.

3. Few KVM packages are availabe with Linux installation.

   To check this, run the command,

   ls /lib/modules/{press tab}/kernel/arch/x86/kvm

   nagarajan@JBL01:~$ ls /lib/modules/4.4.0-21-generic/kernel/arch/x86/kvm

   The three files which are installed in your system will be displayed

   kvm-amd.ko kvm-intel.ko kvm.ko

4. Install the KVM packages

   1. Switch to root (Administrator) user sudo -i

   export http_proxy=http://172.16.0.3:8080

   2. To install the packages,

   run the following commands,

   apt-get update

   apt-get install qemu-kvm apt-get install libvirt-bin apt-get install bridge-utils

   apt-get install virt-manager apt-get install qemu-system

5. To verify your installation, run the command

   virsh -c qemu:///system list

   it shows output

Id Name State

If Vms are running, then it shows name of VM. If VM is not runnign, the system shows blank output, whcih means your KVM installation is perfect.

6. Run the command

virsh --connect qemu:///system list --all

7. Working with KVM

run the command

virsh

version (this command displays version of software tools installed)

nodeinfo (this command displays your system information)

quit (come out of the system)

8. To test KVM installation - we can create Virtual machines but these machines are to be done in manual mode. Skipping this, Directly install Openstack.

Installation of Openstack

1. add new user named stack – This stack user is the adminstrator of the openstack services. To add new user – run the command as root user.

adduser stack

2. run the command

apt-get install sudo -y || install -y sudo

3. Be careful in running the command – please be careful with the syntax. If any error in thsi following command, the system will crash beacause of permission errors.

echo "stack ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers

4. Logout the system and login as stack user

5. Run the command (this installs git repo package)

Run in Root

export http_proxy=http://172.16.0.3:8080

sudo apt-get install git

6. Run the command (This clones updatesd version of dev-stack (which is binary auto- installer package of Openstack)

stack@JBL01:/$ export http_proxy=http://172.16.0.3:8080 stack@JBL01:/$ export https_proxy=http://172.16.0.3:8080 stack@JBL01:/$ git config --global http.proxy $http_proxy stack@JBL01:/$ git config --global https.proxy $http_proxy

git clone http://git.openstack.org/openstack-dev/devstack ls (this shows a folder named devstack)

cd devstack (enter into the folder)

7. create a file called local.conf. To do this run the command,

nano local.conf

stack@JBL01:/devstack$ sudo nano local.conf

8. In the file, make the following entry (Contact Your Network Adminstrator for doubts in these values)

[[local|localrc]] FLOATING_RANGE=192.168.1.224/27 FIXED_RANGE=10.11.11.0/24 FIXED_NETWORK_SIZE=256 FLAT_INTERFACE=eth0 ADMIN_PASSWORD=root DATABASE_PASSWORD=root RABBIT_PASSWORD=root SERVICE_PASSWORD=root SERVICE_TOCKEN=root

9. Save this file

stack@JBL01:/devstack$ sudo gedit stackrc

Save this file

Change File Permission: stack@JBL01:~$ chown stack * -R

10.Run the command (This installs Opentack)

./stack.sh

11. If any error occurs, then run the command for uninistallation

./unstack.sh

1. update the packages apt-get update
2. Then reinstall the package

./stack.sh

12. Open the browser, http://IP address of your machine, you will get the openstack portal.

13. If you restart the machine, then to again start open stack open terminal,

su stack

cd devstack run ./rejoin.sh

14. Again you can access openstack services in the browser, http://IP address of your machine,

VIRTUAL MACHINE CREATION

Launch an instance

1. Log in to the dashboard
2. Select the appropriate project from the drop down menu at the top left.
3. On the Project tab, open the Compute tab and click Instances category.

The dashboard shows the instances with its name, its private and floating IP addresses, size, status, task, power state, and so on.

4. Click Launch Instance.
5. In the Launch Instance dialog box, specify the following values:

Details tab

Availability Zone

By default, this value is set to the availability zone given by the cloud provider (for example, us-west or apac-south). For some cases, it could be nova.

Instance Name

Assign a name to the virtual machine.

Note

The name you assign here becomes the initial host name of the server. If the name is longer than 63 characters, the Compute service truncates it automatically to ensure dnsmasq works correctly. After the server is built, if you change the server name in the API or change the host name directly, the names are not updated in the dashboard.

Server names are not guaranteed to be unique when created so you could have two instances with the same host name.

Flavor

Specify the size of the instance to launch.

Note

The flavor is selected based on the size of the image selected for launching an instance. For example, while creating an image, if you have entered the value in the Minimum RAM (MB) field as 2048, then on selecting the image, the default flavor is m1.small. Instance Count

To launch multiple instances, enter a value greater than 1. The default is 1. Instance Boot Source

Your options are:

Boot from image

If you choose this option, a new field for Image Name displays. You can select the image from the list.

Boot from snapshot

If you choose this option, a new field for Instance Snapshot displays. You can select the snapshot from the list.

Boot from volume

If you choose this option, a new field for Volume displays. You can select the volume from the list.

Boot from image (creates a new volume)

With this option, you can boot from an image and create a volume by entering the Device Size and Device Name for your volume. Click the Delete Volume on Instance Delete option to delete the volume on deleting the instance.

Boot from volume snapshot (creates a new volume)

Using this option, you can boot from a volume snapshot and create a new volume by choosing Volume Snapshot from a list and adding a Device Name for your volume. Click the Delete Volume on Instance Delete option to delete the volume on deleting the instance.

Image Name

This field changes based on your previous selection. If you have chosen to launch an instance using an image, the Image Name field displays. Select the image name from the dropdown list. Instance Snapshot

This field changes based on your previous selection. If you have chosen to launch an instance using a snapshot, the Instance Snapshot field displays. Select the snapshot name from the dropdown list.

Volume

This field changes based on your previous selection. If you have chosen to launch an instance using a volume, the Volume field displays. Select the volume name from the dropdown list. If you want to delete the volume on instance delete, check the Delete Volume on Instance Delete option.

Access & Security tab Key Pair

Specify a key pair.

If the image uses a static root password or a static key set (neither is recommended), you do not need to provide a key pair to launch the instance. Security Groups

Activate the security groups that you want to assign to the instance.

Security groups are a kind of cloud firewall that define which incoming network traffic is forwarded to instances.

If you have not created any security groups, you can assign only the default security group to the instance.

Networking tab Selected Networks

To add a network to the instance, click the + in the Available Networks field.

Network Ports tab Ports

Activate the ports that you want to assign to the instance. Post-Creation tab

Customization Script Source

Specify a customization script that runs after your instance launches. Advanced Options tab

Disk Partition

Select the type of disk partition from the dropdown list:

Automatic

Entire disk is single partition and automatically resizes. Manual

Faster build times but requires manual partitioning.

6. Click Launch.

The instance starts on a compute node in the cloud.

Note

If you did not provide a key pair, security groups, or rules, users can access the instance only from inside the cloud through VNC. Even pinging the instance is not possible without an ICMP rule configured.

You can also launch an instance from the Images or Volumes category when you launch an instance from an image or a volume respectively.

When you launch an instance from an image, OpenStack creates a local copy of the image on the compute

node where the instance starts.

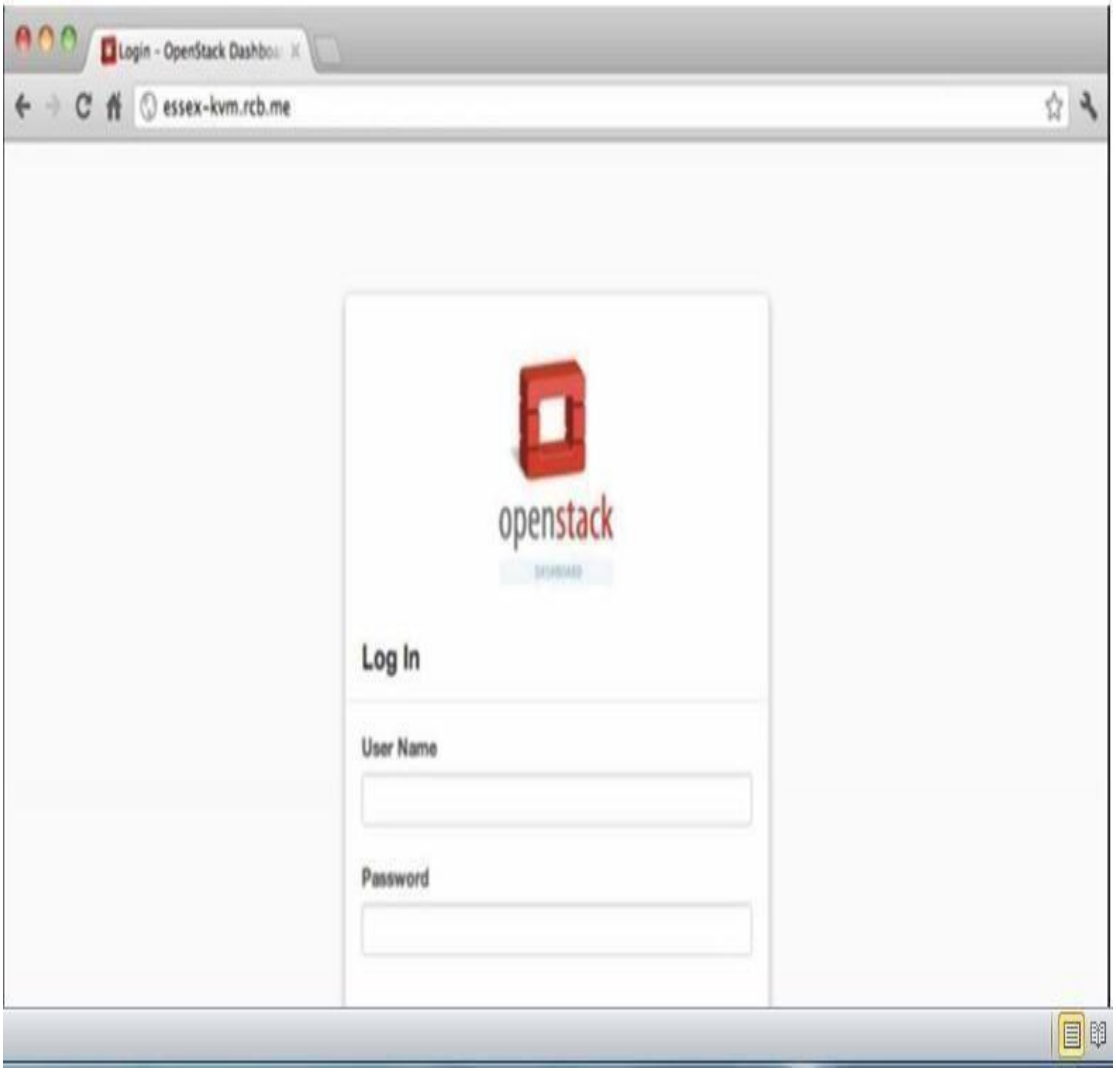
For details on creating images, see Creating images manually in the OpenStack Virtual Machine Image Guide.

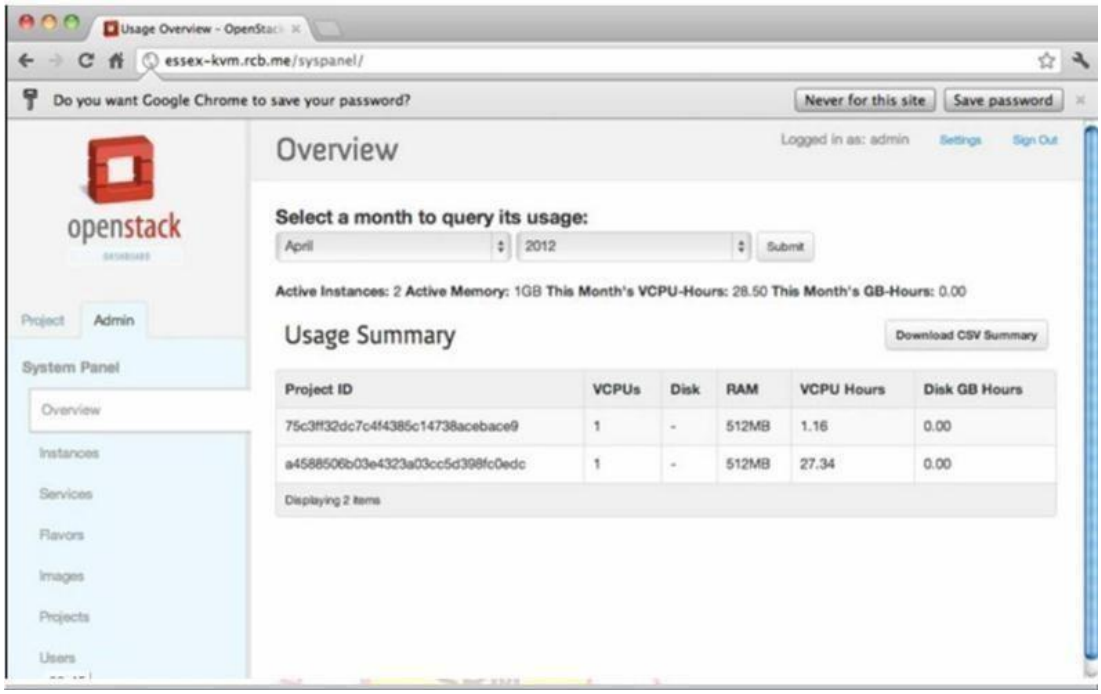
When you launch an instance from a volume, note the following steps:

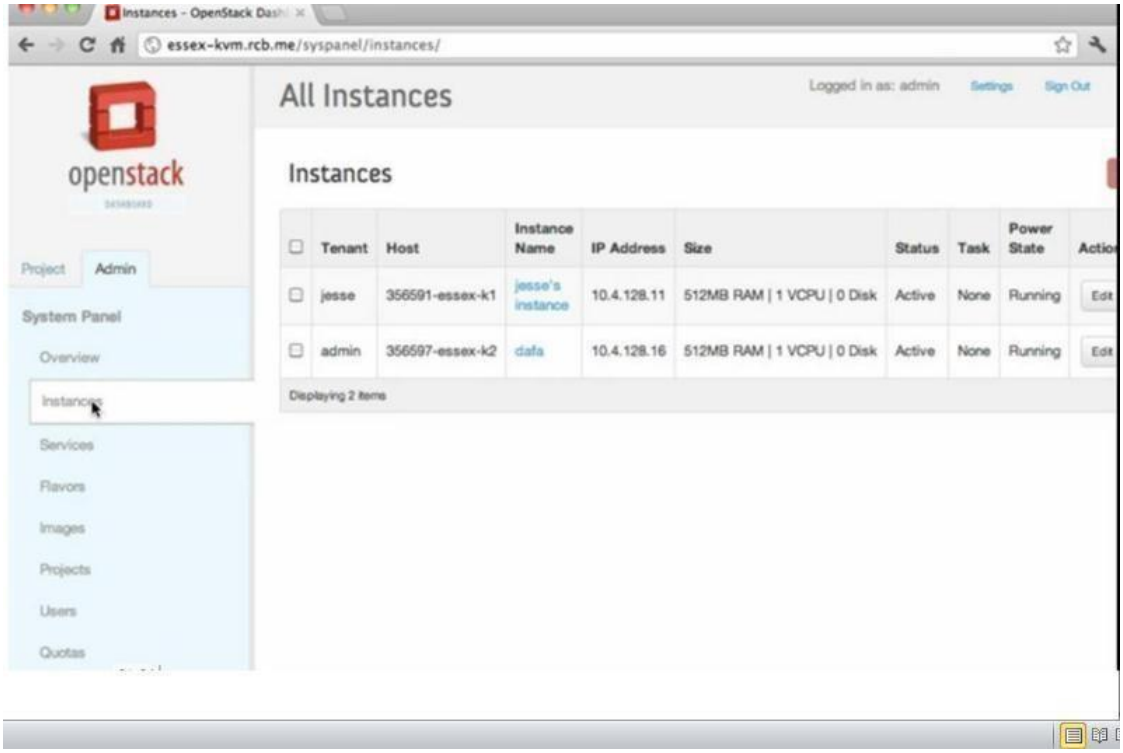
•To select the volume from which to launch, launch an instance from an arbitrary image on the volume. The arbitrary image that you select does not boot. Instead, it is replaced by the image on the volume that you choose in the next steps.

To boot a Xen image from a volume, the image you launch in must be the same type, fully virtualized or paravirtualized, as the one on the volume.

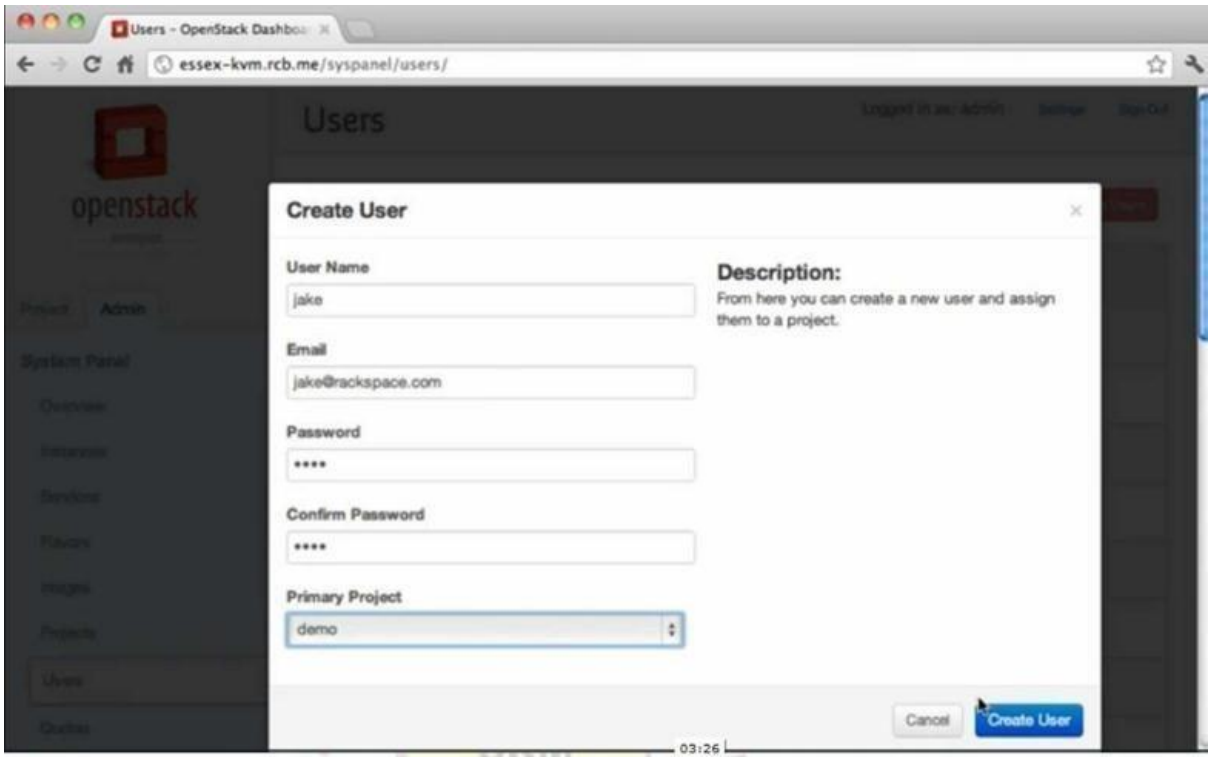
•Select the volume or volume snapshot from which to boot. Enter a device name. Enter vda for KVM images or xvda for Xen images.

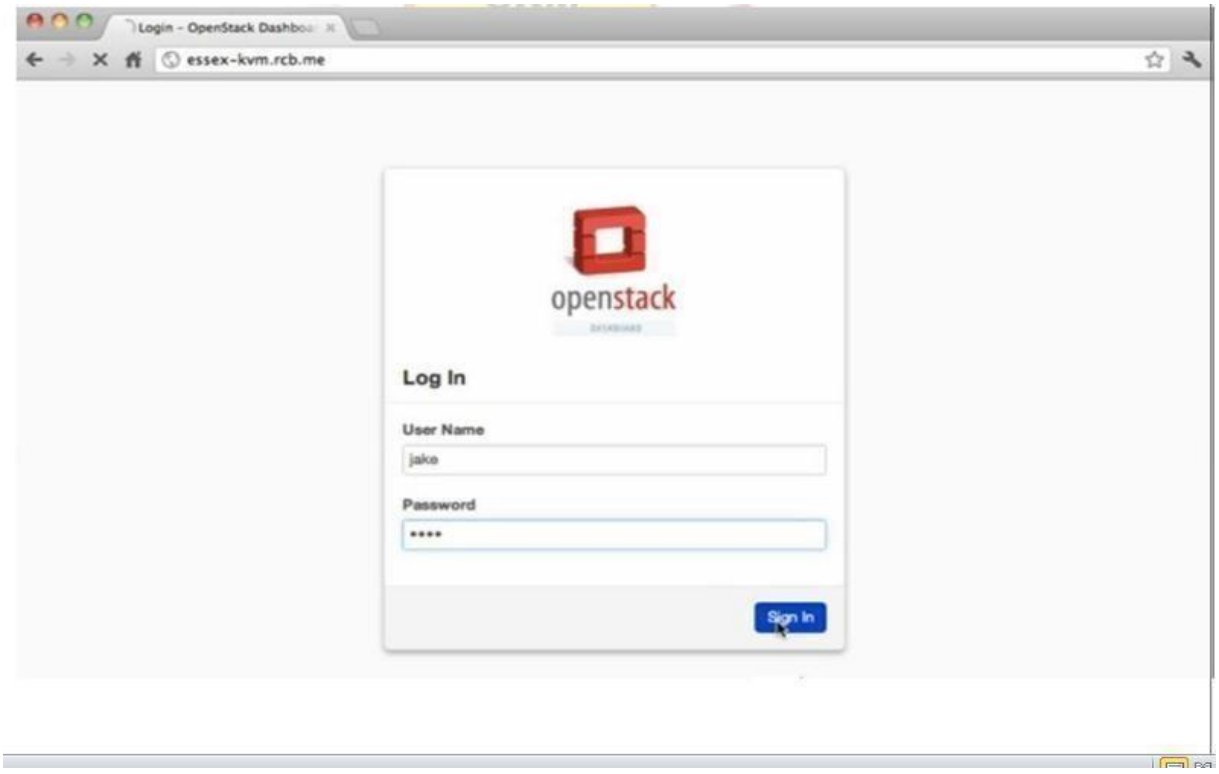


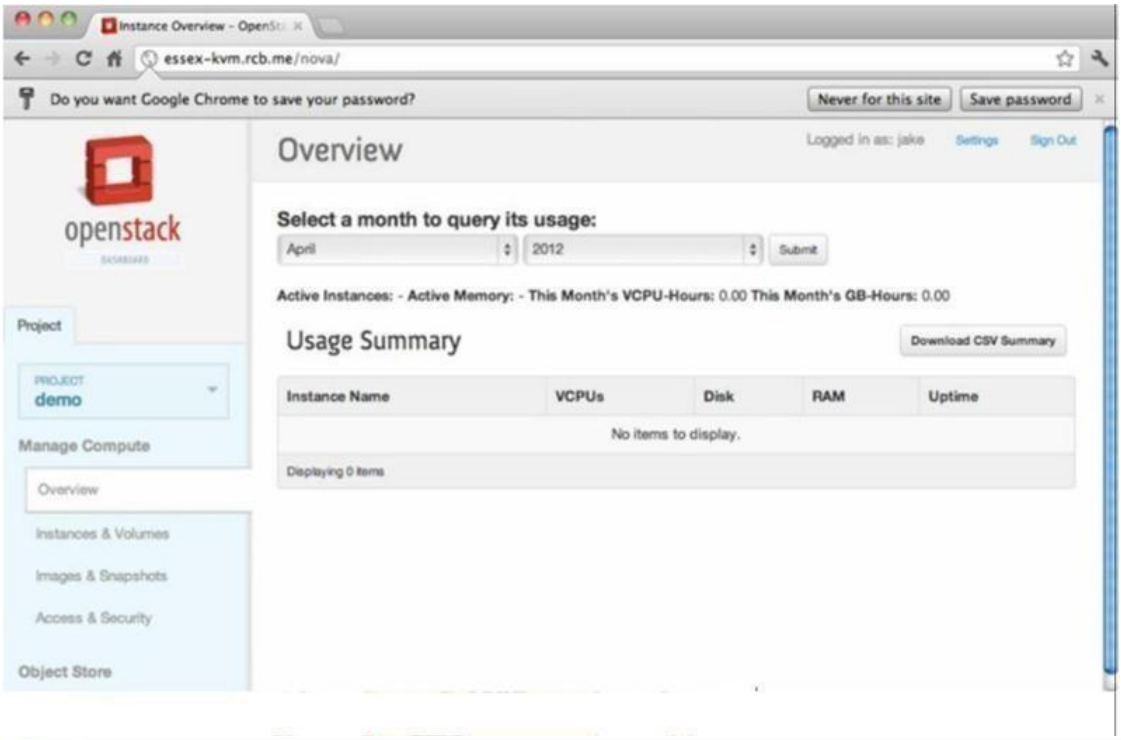

Usage Overview - OpenStack
essex-kvm.rcb.me/syspanel/
Do you want Google Chrome to save your password?
Never for this site
Save password
openstack
Overview
Logged in as: admin
Settings
Sign Out
Select a month to query its usage:
April
2012
Submit
Active Instances: 2 Active Memory: 1GB This Month's VCPU-Hours: 28.50 This Month's GB-Hours: 0.00
Usage Summary
Download CSV Summary
Project ID
VCPUs
Disk
RAM
VCPU Hours
Disk GB Hours
75c3ff32dc7c4f4385c14738acebace9
1
-
512MB
1.16
0.00
a4588506b03e4323a03cc5d398fc0edc
1
-
512MB
27.34
0.00
Displaying 2 items
Project
Admin
System Panel
Overview
Instances
Services
Flavors
Images
Projects
Users

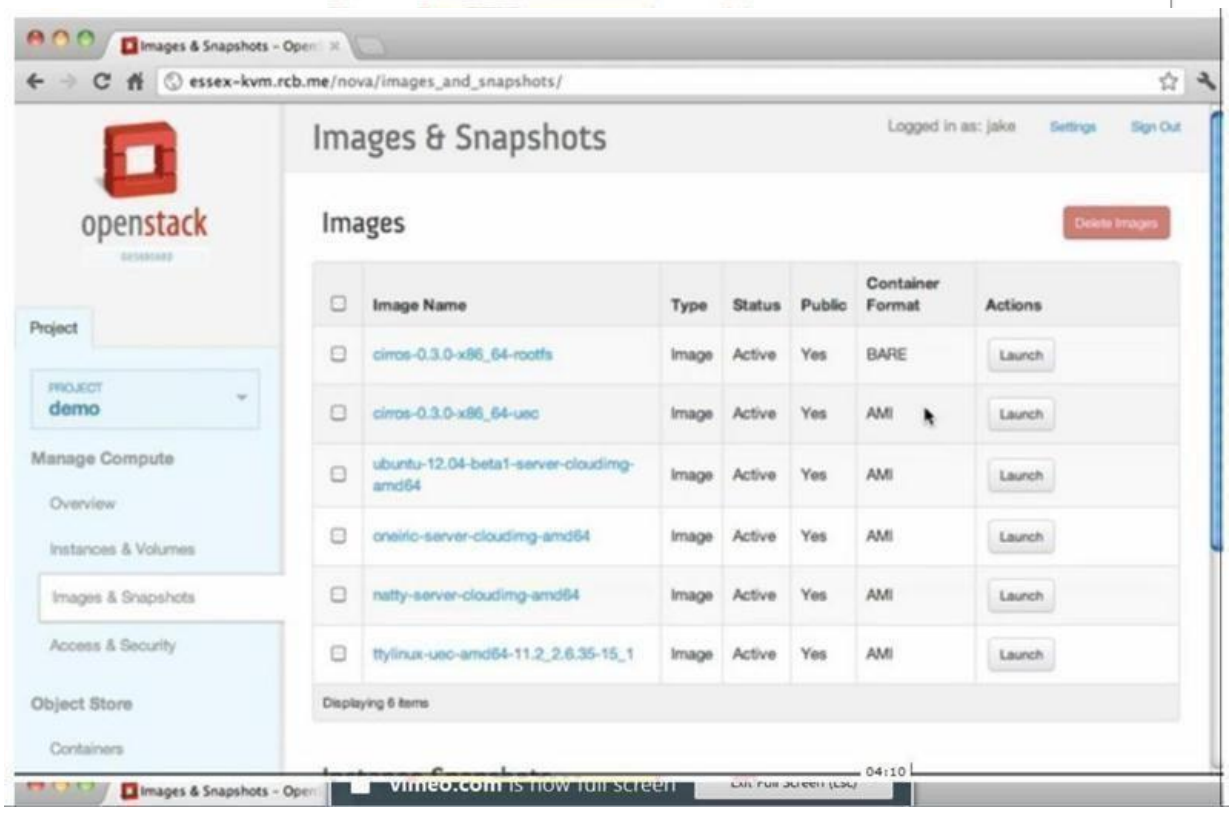

Instances - OpenStack Dash
essex-kvm.rcb.me/syspanel/instances/
openstack
All Instances
Logged in as: admin
Settings
Sign Out
Instances
Tenant
Host
Instance Name
IP Address
Size
Status
Task
Power State
jesse
356591-essex-k1
jesse's instance
10.4.128.11
512MB RAM | 1 VCPU | 0 Disk
Active
None
Running
admin
356597-essex-k2
dafa
10.4.128.16
512MB RAM | 1 VCPU | 0 Disk
Active
None
Running
Displaying 2 items
Project
Admin
System Panel
Overview
Instances
Services
Flavors
Images
Projects
Users
Quotas

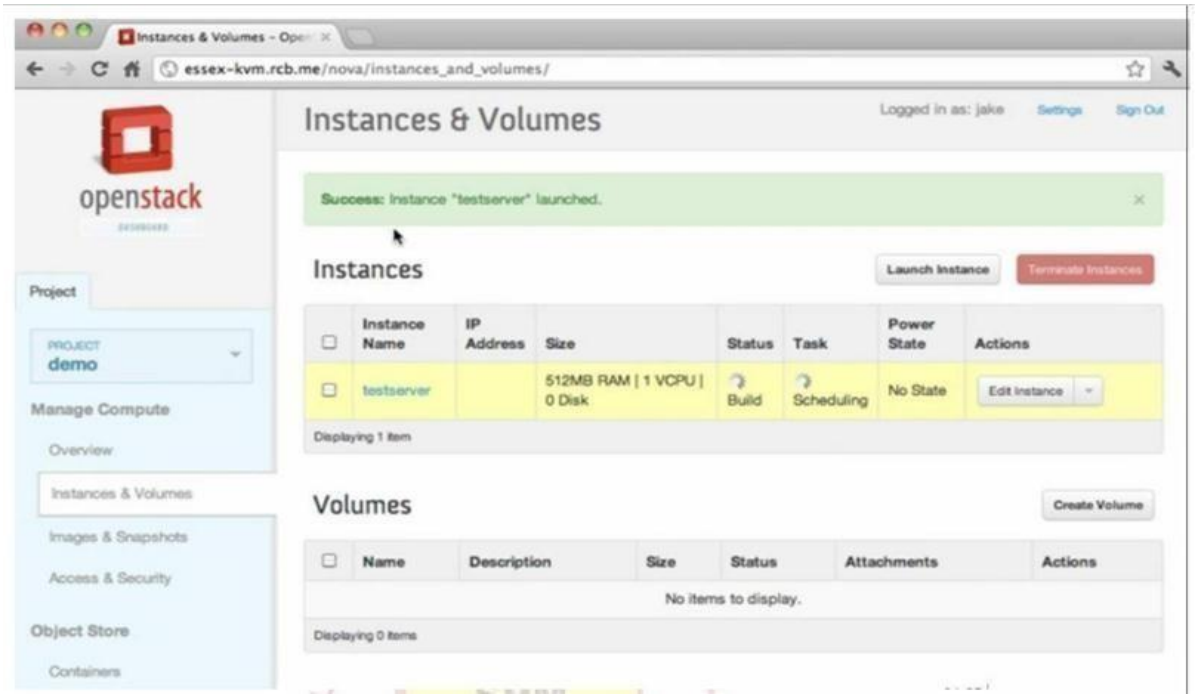
CREATING USER

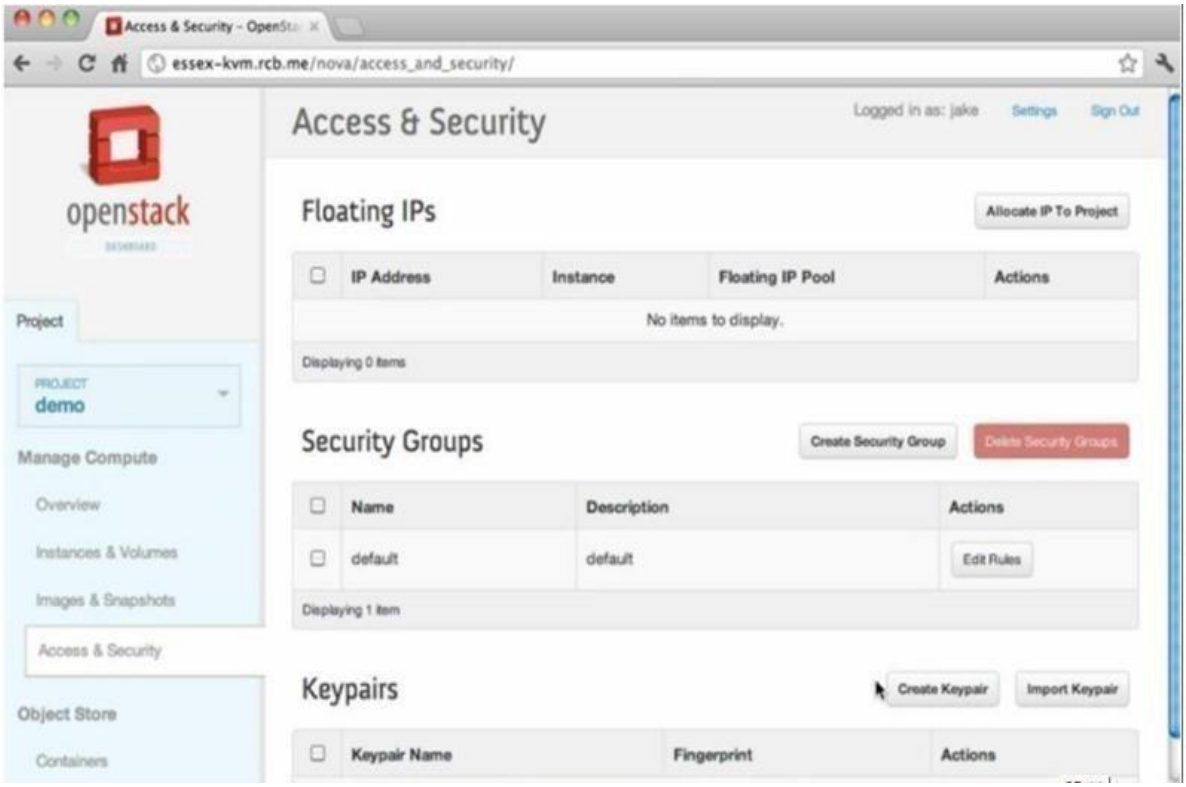


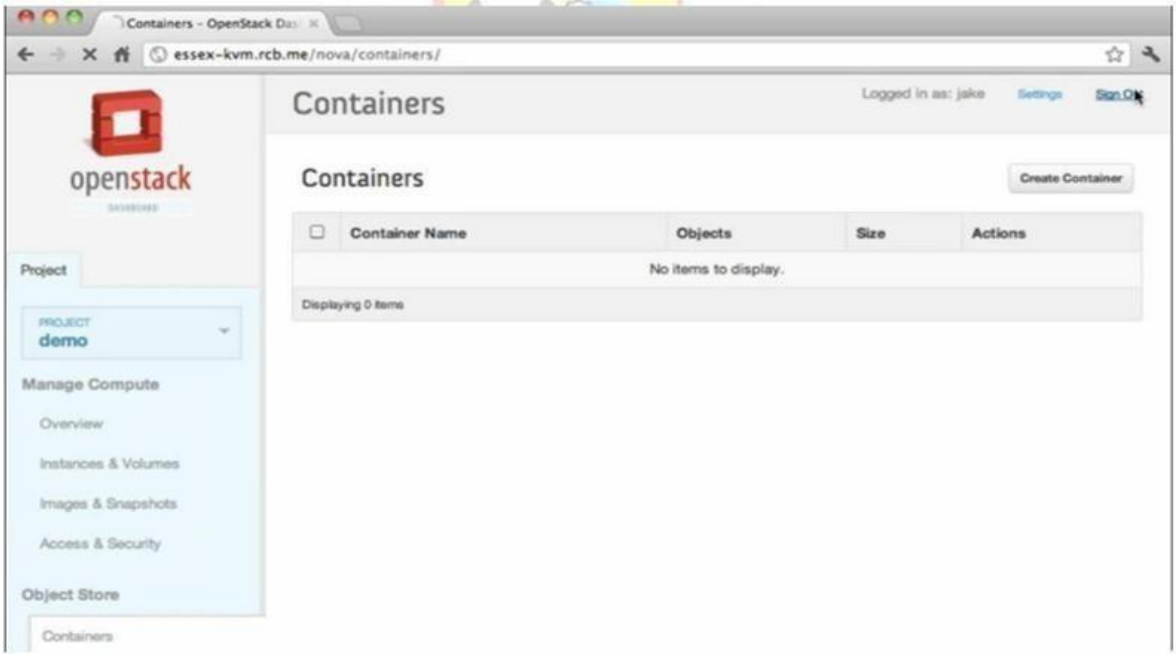



Instance Overview - OpenSta
essex-kvm.rcb.me/nova/
Do you want Google Chrome to save your password?
Never for this site
Save password
openstack
Overview
Logged in as: jake
Settings
Sign Out
Select a month to query its usage:
April
2012
Submit
Active Instances: - Active Memory: - This Month's VCPU-Hours: 0.00 This Month's GB-Hours: 0.00
Usage Summary
Download CSV Summary
Instance Name
VCPUs
Disk
RAM
Uptime
No items to display.
Displaying 0 items
Project
PROJECT
demo
Manage Compute
Overview
Instances & Volumes
Images & Snapshots
Access & Security
Object Store


Images & Snapshots - Oper
essex-kvm.rcb.me/nova/images_and_snapshots/
openstack
Images & Snapshots
Logged in as: jake
Settings
Sign Out
Images
Delete Images
Image Name
Type
Status
Public
Container Format
Actions
cirros-0.3.0-x86_64-rootfs
Image
Active
Yes
BARE
Launch
cirros-0.3.0-x86_64-uec
Image
Active
Yes
AMI
Launch
ubuntu-12.04-beta1-server-cloudimg-amd64
Image
Active
Yes
AMI
Launch
oneiric-server-cloudimg-amd64
Image
Active
Yes
AMI
Launch
natty-server-cloudimg-amd64
Image
Active
Yes
AMI
Launch
ttylinux-uec-amd64-11.2_2.6.35-15_1
Image
Active
Yes
AMI
Launch
Displaying 6 items
Project
PROJECT
demo
Manage Compute
Overview
Instances & Volumes
Images & Snapshots
Access & Security
Object Store
Containers
04:10
Images & Snapshots - Oper
vimeo.com is now full screen
Exit Full Screen (Esc)


Instances & Volumes - Open
essex-kvm.rcb.me/nova/instances_and_volumes/
openstack
Instances & Volumes
Logged in as: jake
Settings
Sign Out
Success: Instance "testserver" launched.
Project
PROJECT
demo
Manage Compute
Overview
Instances & Volumes
Images & Snapshots
Access & Security
Object Store
Containers
Instances
Launch Instance
Terminate Instances
Instance Name
IP Address
Size
Status
Task
Power State
Actions
testserver
512MB RAM | 1 VCPU | 0 Disk
Build
Scheduling
No State
Edit Instance
Displaying 1 item
Volumes
Create Volume
Name
Description
Size
Status
Attachments
Actions
No items to display.
Displaying 0 items


Access & Security - OpenSta
essex-kvm.rcb.me/nova/access_and_security/
openstack
Access & Security
Logged in as: jake
Settings
Sign Out
Project
PROJECT
demo
Manage Compute
Overview
Instances & Volumes
Images & Snapshots
Access & Security
Object Store
Containers
Floating IPs
Allocate IP To Project
IP Address
Instance
Floating IP Pool
Actions
No items to display.
Displaying 0 items
Security Groups
Create Security Group
Delete Security Groups
Name
Description
Actions
default
default
Edit Rules
Displaying 1 item
Keypairs
Create Keypair
Import Keypair
Keypair Name
Fingerprint
Actions



Result:

Thus, the virtual machine is launched in Ubuntu 14.04.