

# CLOUD COMPUTING LAB

G.SUDHEER KUMAR CSE(AIML)  
KPRIT ENGINEERING COLLEGE [GHATKESAR]

shyam-sudheer1602

EX NO: 1

Date:

**INSTALL VIRTUALBOX WITH LINUX OS ON TOP OF WINDOWS****AIM: INSTALL VIRTUALBOX WITH LINUX OS ON TOP OF WINDOWS****PROCEDURE:****Steps to install VirtualBox:**

1. Download VirtualBox installer for windows.
2. The installer can be downloaded from the link : <https://www.virtualbox.org/wiki/Downloads>
3. Click "Windows host" to download the binary version for windows host.
4. The installer file downloaded will have the file name format like "VirtualBox-VersionNumber-BuildNumber-Win.exe". Example: VirtualBox-6.1.12-139181-Win.exe.
5. Double click on the installer to launch the setup Wizard. Click on Next to continue.
6. Custom setup dialog box will be opened. Accept the default settings and click next.
7. Select the way you want the features to be installed. You can accept the default and click next.
8. A dialog box opens with Network Interfaces warning. Click Yes to proceed.
9. Click install to begin the installation process.
10. When prompted with a message to install (Trust) Oracle Universal Serial Bus, click Install to continue.
11. After the installation completes, click finish to exit the setup wizard.
12. Launch the Oracle VM VirtualBox.

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

1. Open the Oracle VM VirtualBox.
2. Click New icon or "Ctrl + N" to create a new virtual machine.
3. Enter a name for the new virtual machine. Choose the Type and Version. Note that VirtualBox automatically changes 'Type' to Linux and 'Version' to 'Ubuntu (64 bit)' if the name is given as 'Ubuntu'. Click Next.
4. Select the amount of RAM to use. The ideal amount of RAM will automatically be selected. Do not increase the RAM into the red section of the slider; keep the slider in the green section.
5. Accept the default 'Create a virtual hard drive now' and click 'Create' button.
6. Choose the hard disk file type as VDI (VirtualBox Disk Image). Click Next.
7. Click Next to accept the default option 'Dynamically allocated' for storage on physical hard drive.
8. Select the size of the virtual hard disk and click create.
9. The newly created virtual machine will be displayed in the dashboard.
10. Download the ISO file [Ubuntu disk image file]. Latest version of Ubuntu iso file can be downloaded from the link <https://ubuntu.com/download/desktop> . Click Download button.
11. For previous versions, goto <http://releases.ubuntu.com> . Choose the preferred version of Ubuntu and download the iso file.
12. To setup the Ubuntu disk image file (iso file) goto settings.
13. Click Storage. Under "Storage Devices" section click "Empty".
14. In Attributes section, click the disk image and then "Choose Virtual Optical Disk File".
15. Browse and select the downloaded iso file. Click ok.
16. Select the newly created virtual machine in the dashboard and click start button.
17. In the welcome screen, click "Install Ubuntu" button.



## Download VirtualBox

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

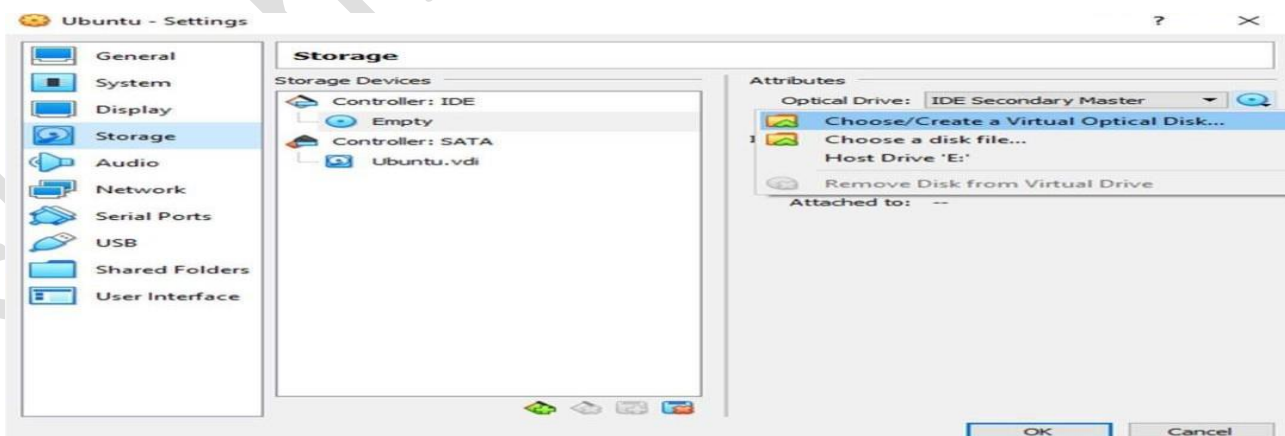
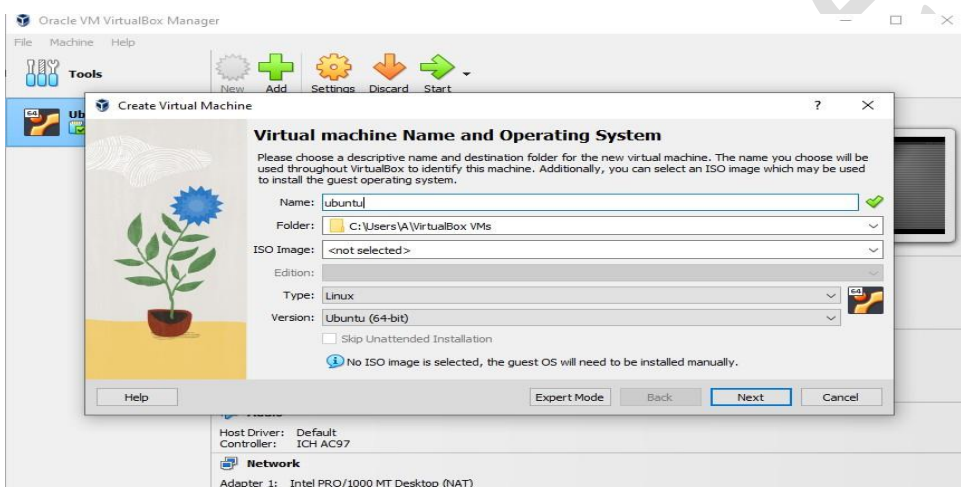
### VirtualBox binaries

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

If you're looking for the latest VirtualBox 6.1 packages, see VirtualBox 6.1 builds. Version 6.1 will remain supported until December 2023.

### VirtualBox 7.0.10 platform packages

- Windows hosts
- macOS / Intel hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

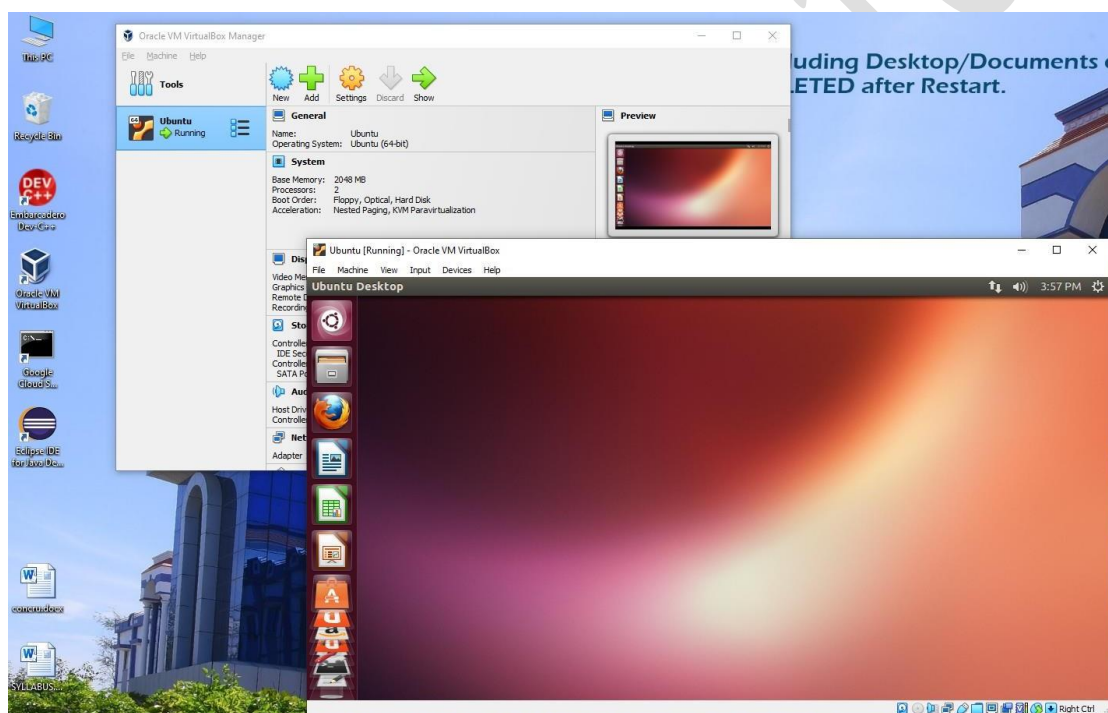


shyam-sudheer1602



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

### OUTPUT:



### Result:

EX NO: 2

Date:

## INSTALL A C COMPILER IN THE VIRTUAL MACHINE

AIM:

## PROCEDURE:

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

```
$ sudo apt-get update
```

```
$ sudo apt-get install gcc
```

It will install all the necessary packages for gcc compiler.

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

```
//fact.c
```

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int n,fact=1;
```

```
int i=1;
```

```
printf("Enter a positive integer:");
```

```
scanf("%d",&n);
```

```
if(n==0)
```

```
{
```

```
fact=1;
```

```
}
```

```
else
```

```
{
```

```
while(i<=n)
```

```
{
```

```
fact*=i;
```

```
i++;
```

```
}
```

```
printf("The factorial of %d is %d ",n,fact);
```

```
}}
```

4. Compile and Run the C Program

```
cc fact.c
```

```
./a.out
```

## OUTPUT:

```

johnnewtonj@johnnewtonj-VirtualBox: ~
johnnewtonj@johnnewtonj-VirtualBox:~$ sudo apt-get install gcc
[sudo] password for johnnewtonj:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
johnnewtonj@johnnewtonj-VirtualBox:~$

```

```

fact.c (~) - gedit
#include<stdio.h>
void main()
{
    int n,fact=1;
    int i=1;
    printf("Enter a positive integer:");
    scanf("%d",&n);
    if(n==0)
    {
        fact=1;
    }
    else
    {
        while(i<=n)
        {
            fact*=i;
            i++;
        }
        printf("The factorial of %d is %d ",n,fact);
    }
}

johnnewtonj@johnnewtonj-VirtualBox: ~
johnnewtonj@johnnewtonj-VirtualBox:~$ gcc fact.c
johnnewtonj@johnnewtonj-VirtualBox:~$ ./a.out
Enter a positive integer:3
johnnewtonj@johnnewtonj-VirtualBox:~$ gcc fact.c
johnnewtonj@johnnewtonj-VirtualBox:~$ ./a.out
Enter a positive integer:3
The factorial of 3 is 6 johnnewtonj@johnnewtonj-VirtualBox:~$

```

## RESULT:



EX NO: 3

Date:

## INSTALL GOOGLE APP ENGINE AND CREATE A WEB APPLICATIONS USING JAVA

**AIM:**

**PROCEDURE:**

**Google App Engine SDK Installation:**

1. Download the Google Cloud SDK installer using the link <https://cloud.google.com/appengine/downloads>.

2. Select the standard environment as Java.

3. Click „Download and Install the Cloud SDK“. Launch the installer and follow the prompts.

4. After installation has completed, the installer presents several options:

Make sure that the following are selected:

- Start Google Cloud SDK Shell.
  - Run 'gcloudinit' .The installer then starts a terminal window and runs the **gcloudinit** command.
5. Run the following command in your terminal to install the gcloud component that includes the App Engine extension for Java11:

**gcloud components install app-engine-java**

**Creating a new App Engine standard project in Eclipse:**

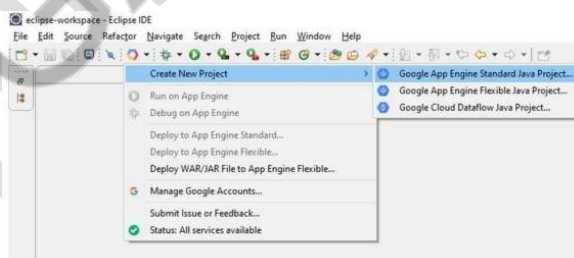
6. Eclipse with the cloud tools is used to create App Engine standard project.

7. To install the Cloud Tools in Eclipse, select Help > Eclipse Marketplace... and search for “Google Cloud Tools for Eclipse” and click install.

8. After installation restart eclipse when prompted.

9. Click the Google Cloud Platform toolbar button.

10. Select Create **New Project > Google App Engine Standard Java Project**.



11. Enter the project name and package name.

12. Click Next. Select the libraries required for the project.

13. Click Finish.

14. The wizard generates a native Eclipse project, with a simple servlet, that you can run and deploy from the IDE.

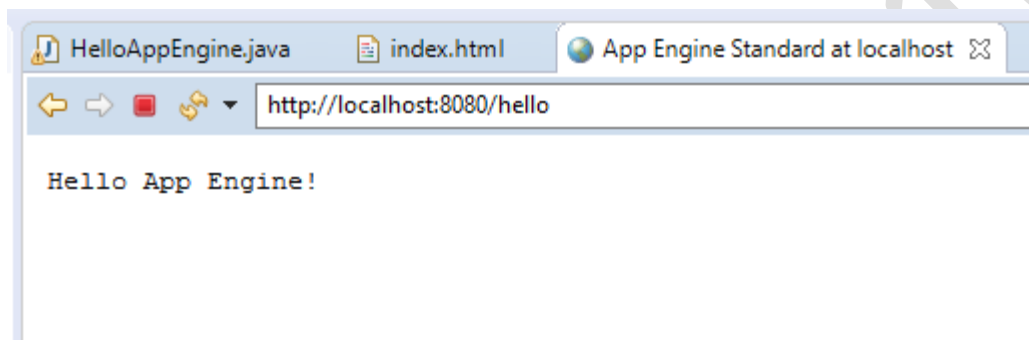
15. App Engine Java applications use the Java Servlet API to interact with the web server. Modify the defaultHelloAppEngine.java file with your application code.



# Hello App Engine!

## Available Servlets:

[The servlet](#)



16. appengine-web.xml is a Google App Engine specific configuration file.
17. web.xml is a standard web application configuration file.
18. Right click the project in the Package Explorer, select **Run As > App Engine**.
19. Eclipse opens its internal web browser to your application.

You can also open an external browser and navigate to <http://localhost:8080>.

Either way, you'll see a static HTML page with a link to the servlet.

### Index.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
  <head>
    <meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8" />
    <title>Hello App Engine</title>
  </head>
  <body>
    <h1>Hello App Engine!</h1>
    <table>
      <tr>
        <td colspan="2" style="font-weight:bold;">Available Servlets:</td>
      </tr>
      <tr>
        <td><a href="/hello">The servlet</a></td>
      </tr>
    </table>
  </body>
</html>
```

```

HelloAppEngine.java
package com.pack;
import java.io.IOException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(
    name = "HelloAppEngine",
    urlPatterns = {"/hello"}
)
public class HelloAppEngine extends HttpServlet {
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {
        response.setContentType("text/plain");
        response.setCharacterEncoding("UTF-8");
        response.getWriter().print("Hello App Engine!\r\n");
    }
}

```

**RESULT:**

EX NO: 4

Date:

## Install Google App Engine and create a simple web applications using python/JAVA

**AIM:****PROCEDURE:**

1. Download the python from <http://www.python.org/downloads> and install it in the system.
2. Download the SDK for python from <https://cloud.google.com/appengine/docs> .
3. Login into Cloud SDK with the Google account .
4. Create a python file and save it as index.py
5. Create a YAML file for configuration and name it as app.yaml
6. Open the Cloud SDK window and type the command Google-cloud-sdk\bin\dev-appserver.py "<>"
7. In web browser obtain the result from address localhost:8080.

**PROGRAM:****app.yaml**

```
runtime: python27
threadsafe: true
handlers:
- url: /
  script: main.app
```

**main.py**

```
import os
import json
import urllib
import webapp2
from google.appengine.ext.webapp import template

class MainPage(webapp2.RequestHandler):
    def get(self):
        template_values = {}
        path = os.path.join(os.path.dirname(__file__), 'index.html')
        self.response.out.write(template.render(path, template_values))

    def post(self):
        pincode = self.request.get('zipCode')
        if not pincode.isnumeric() or not len(pincode) == 6:
            template_values = {
                "error": "Incorrect Pin Code (String / False Code entered)"
            }
            path = os.path.join(os.path.dirname(__file__), 'index.html')
            return self.response.out.write(template.render(path, template_values))
        url = "https://api.postalpincode.in/pincode/" + pincode
        data = urllib.urlopen(url).read()
        data = json.loads(data)
        if(data[0]['Status'] == 'Success'):
            post_office = data[0]['PostOffice'][0]['State']
```

## OUTPUT:

```

C:\Windows\System32\cmd.exe - dev_appserver.py "D:\sem7\cloud\WebApp"
C:\Users\A\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin>dev_appserver.py "D:\sem7\cloud\WebApp"
INFO 2023-08-04 09:36:31,894 devappserver2.py:321] Skipping SDK update check.
INFO 2023-08-04 09:36:32,066 <string>:398] Starting API server at: http://localhost:63218
INFO 2023-08-04 09:36:32,066 dispatcher.py:276] Starting module "default" running at: http://localhost:8080
INFO 2023-08-04 09:36:32,066 admin_server.py:70] Starting admin server at: http://localhost:8000
INFO 2023-08-04 09:36:34,082 instance.py:294] Instance PID: 3964
INFO 2023-08-04 09:36:37,401 module.py:426] [default] Detected file changes:
main.pyc
INFO 2023-08-04 09:36:37,878 module.py:862] default: "GET / HTTP/1.1" 200 1828
INFO 2023-08-04 09:36:38,075 module.py:862] default: "GET /favicon.ico HTTP/1.1" 404 -
INFO 2023-08-04 09:36:38,413 instance.py:294] Instance PID: 1524
INFO 2023-08-04 09:36:40,443 instance.py:294] Instance PID: 5916
WARNING 2023-08-04 04:06:51,296 sandbox.py:1135] The module _winreg is whitelisted for local dev only. If your application relies on _winreg, it is likely that it will not function properly in production.
WARNING 2023-08-04 09:36:51,325 urlfetch_stub.py:575] Stripped prohibited headers from URLFetch request: ['Host']
INFO 2023-08-04 09:36:53,628 module.py:862] default: "POST / HTTP/1.1" 200 1501
INFO 2023-08-04 09:37:05,111 module.py:862] default: "GET / HTTP/1.1" 200 1828
WARNING 2023-08-04 09:37:12,871 urlfetch_stub.py:575] Stripped prohibited headers from URLFetch request: ['Host']
INFO 2023-08-04 09:37:13,946 module.py:862] default: "POST / HTTP/1.1" 200 1495
INFO 2023-08-04 09:37:21,838 module.py:862] default: "GET / HTTP/1.1" 200 1828
WARNING 2023-08-04 09:37:30,270 urlfetch_stub.py:575] Stripped prohibited headers from URLFetch request: ['Host']
INFO 2023-08-04 09:37:31,355 module.py:862] default: "POST / HTTP/1.1" 200 1489
INFO 2023-08-04 09:37:50,053 module.py:862] default: "GET / HTTP/1.1" 200 1828
WARNING 2023-08-04 09:38:05,088 urlfetch_stub.py:575] Stripped prohibited headers from URLFetch request: ['Host']
INFO 2023-08-04 09:38:06,150 module.py:862] default: "POST / HTTP/1.1" 200 1501
INFO 2023-08-04 09:38:37,418 module.py:862] default: "GET / HTTP/1.1" 200 1828
WARNING 2023-08-04 09:38:43,328 urlfetch_stub.py:575] Stripped prohibited headers from URLFetch request: ['Host']
INFO 2023-08-04 09:38:44,714 module.py:862] default: "POST / HTTP/1.1" 200 1495
INFO 2023-08-04 09:39:01,151 module.py:862] default: "GET / HTTP/1.1" 200 1828
WARNING 2023-08-04 09:47:06,903 urlfetch_stub.py:575] Stripped prohibited headers from URLFetch request: ['Host']
INFO 2023-08-04 09:47:08,005 module.py:862] default: "POST / HTTP/1.1" 200 1489

```



## Post Office Finder Using WebApp

Location Zip Code:





State of Post Office : Tamil Nadu

Name of Post Office : Kurumbur

Block of Post Office: Tiruchendur

District of Post Office: Tuticorin

[Back to the Home page](#)

```

name = data[0]['PostOffice'][0]['Name']
block = data[0]['PostOffice'][0]['Block']
district = data[0]['PostOffice'][0]['District']
template_values = {
    "post_office": post_office,
    "name": name,
    "block": block,
    "district": district
}
path = os.path.join(os.path.dirname(__file__), 'results.html')
self.response.out.write(template.render(path, template_values))
else:
    template_values = {}
    path = os.path.join(os.path.dirname(__file__), 'error.html')
    self.response.out.write(template.render(path, template_values))

```

```
app = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

### index.html

```

<html>
<style>
    .weatherText {
        font-family: 'Lato', 'sans-serif';
        font-size: 24px;
        text-align: center;
    }

    #weatherForm {
        padding: 20px;
    }

    #weatherSubmit {
        color: white;
        background-color: #083375;
        padding: 5px 20px;
        border-radius: 5px;
        margin-top: 20px;
    }

    #weatherSubmit:hover {
        cursor: pointer;
    }
    body {
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .card {
        border: 2px solid black;
        width: 50%;

        justify-content: center;
        align-items: center;
    }
</style>
<head>
<title class="alignct">Post Office Finder</title>

```

```

<link href=https://fonts.googleapis.com/css2?family=Lato:wght@400;700&display=swap rel="stylesheet" />
</head>
<body>
  <div class="card">
    <h2 class="weatherText">Post Office Finder Using WebApp</h2>
    <h1 id="error_head" style="display: none" value="{{error}}">{{error}}</h1>
    <form class="weatherText" id="weatherForm" action="/" method="post">
      Location Zip Code:
      <input class="weatherText" id="weatherInput" type="text" name="zipCode"/><br />
      <input class="weatherText" id="weatherSubmit" type="submit" value="Submit"/>
    <button id="weatherSubmit" class="weatherText" onclick="document.getElementById('weatherInput').value = ''">
      Clear</button>
    </form>
  </div>
  <script>
    let err = document.getElementById('error_head');
    function myFunction() {
      alert('Please Enter the Valid Pin Code!');
    }
    if (err) {
      myFunction();
    }
  </script>
</body>
</html>

```

**results.html**

```

<!DOCTYPE html>
<html lang="en">
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
    }
    #weatherResults {
      background-color: #83e9c2;
      font-family: 'Lato', sans-serif;
      font-size: 24px;
      padding: 30px;
      display: inline-block;
      text-align: center;
      margin: 20px;
      margin-top: 10%;
      border: 2px solid black;
      border-radius: 5px;
    }
  </style>
  <head>
    <meta charset="UTF-8" />
    <title>Post Office Information</title>
    <link href=https://fonts.googleapis.com/css2?family=Lato:wght@400;700&display=swap rel="stylesheet"/>
  </head>
  <body>
    <div id="weatherResults">
      <table>
        <tr>
          <th>
            <h3>State of Post Office :</h3>
          </th>
          <th>
            <h3>{{ post_office }}</h3>
          </th>
        </tr>
      </table>
    </div>
  </body>

```

```

<tr>
  <th>
    <h3>Name of Post Office :</h3>
  </th>
  <th>
    <h3>{{ name }}</h3>
  </th>
</tr>
<tr>
  <th>
    <h3>Block of Post Office:</h3>
  </th>
  <th>
    <h3>{{ block }}</h3>
  </th>
</tr>
<tr>
  <th>
    <h3>District of Post Office:</h3>
  </th>
  <th>
    <h3>{{ district }}</h3>
  </th>
</tr>
</table>
<a href="http://localhost:8080/"><h4>Back to the Home page</h4></a>
</div>
</body>
</html>

```

**error.html**

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Error page</title>
    <link href="https://fonts.googleapis.com/css2?family=Lato:wght@400;700&display=swap" rel="stylesheet"/>
  </head>
  <body>
    <div style="text-align: center">
      <h2>No such pin exists</h2>
      <a href="http://localhost:8080/"><h3>Back to the Home page</h3></a>
    </div>
  </body>
</html>

```

**RESULT:**



EX NO: 5

Date:

## SIMULATE A CLOUD SCENARIO USING CLOUDSIM AN RUN A SCHEDULING ALGORITHM THIS IS NOT IN CLOUDSIM

**AIM:**

**PROCEDURE:**

**What is Cloudsim?**

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

This toolkit allows to:

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

Core features of CloudSim are:

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

**How to use CloudSim in Eclipse:**

CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful. It is a library and, hence, CloudSim does not have to be installed. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used. Please verify whether Java is available on your system.

**To use Cloudsim in eclipse:**

1. Download CloudSim installable files from <https://code.google.com/p/cloudsim/downloads/list> and unzip
2. Open Eclipse
3. Create a new Java Project: **File -> New**
4. Import an unpacked CloudSim project into the new Java Project
5. The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows:

**CloudSim.init(num\_user, calendar, trace\_flag)**

6. Data centres are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or spaceshared, the time zone and its price:

```
Datacenter datacenter9883 = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList)
```

7. The third step is to create a broker: **DatacenterBroker broker = createBroker();**

8. The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM's owner, mips, number Of Pes amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:

```
Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())
```

9. Submit the VM list to the broker: **broker.submitVmList(vmlist)**

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

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

11. Submit the cloudlet list to the broker: **broker.submitCloudletList(cloudletList)**

12. Start the simulation: **CloudSim.startSimulation().**

#### PROGRAM:

##### SJF\_Scheduler.java

```
package com.sjfs;
import org.cloudbus.cloudsim.*;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;

public class SJF_Scheduler {

    private static List<Cloudlet> cloudletList;
    private static List<Vm> vmList;
    private static Datacenter[] datacenter;
    private static double[][] commMatrix;
    private static double[][] execMatrix;

    private static List<Vm> createVM(int userId, int vms) {
        //Creates a container to store VMs. This list is passed to the broker later
        LinkedList<Vm> list = new LinkedList<Vm>();

        //VM Parameters
        long size = 10000; //image size (MB)
        int ram = 512; //vm memory (MB)
        int mips = 250;
        long bw = 1000;
        int pesNumber = 1; //number of cpus
```

```

String vmm = "Xen"; //VMM name

//create VMs
Vm[] vm = new Vm[vms];

for (int i = 0; i < vms; i++) {
    vm[i] = new Vm(datacenter[i].getId(), userId, mips, pesNumber, ram, bw, size, vmm, new
CloudletSchedulerSpaceShared());
    list.add(vm[i]);
}

return list;
}

private static List<Cloudlet> createCloudlet(int userId, int cloudlets, int idShift) {
    // Creates a container to store Cloudlets
    LinkedList<Cloudlet> list = new LinkedList<Cloudlet>();

    //cloudlet parameters
    long fileSize = 300;
    long outputSize = 300;
    int pesNumber = 1;
    UtilizationModel utilizationModel = new UtilizationModelFull();

    Cloudlet[] cloudlet = new Cloudlet[cloudlets];

    for (int i = 0; i < cloudlets; i++) {
        int dcId = (int) (Math.random() * Constants.NO_OF_DATA_CENTERS);
        long length = (long) (1e3 * (commMatrix[i][dcId] + execMatrix[i][dcId]));
        cloudlet[i] = new Cloudlet(idShift + i, length, pesNumber, fileSize, outputSize, utilizationModel,
utilizationModel, utilizationModel);
        // setting the owner of these Cloudlets
        cloudlet[i].setUserId(userId);
        cloudlet[i].setVmId(dcId + 2);
        list.add(cloudlet[i]);
    }
    return list;
}

public static void main(String[] args) {
    Log.println("Starting SJF Scheduler...");

    new GenerateMatrices();
    execMatrix = GenerateMatrices.getExecMatrix();
    commMatrix = GenerateMatrices.getCommMatrix();

    try {
        int num_user = 1; // number of grid users
        Calendar calendar = Calendar.getInstance();
        boolean trace_flag = false; // mean trace events

        CloudSim.init(num_user, calendar, trace_flag);

        // Second step: Create Datacenters
        datacenter = new Datacenter[Constants.NO_OF_DATA_CENTERS];
        for (int i = 0; i < Constants.NO_OF_DATA_CENTERS; i++) {

```

## OUTPUT:

```

<terminated> SJF_Scheduler [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (04-Aug-2023, 9:49:46 am)
Starting SJF Scheduler...
Initializing new Matrices...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Datacenter_2 is starting...
Datacenter_3 is starting...
Datacenter_4 is starting...
Broker_0 is starting...
Entities started.
0.0: Broker_0: Cloud Resource List received with 5 resource(s)
0.0: Broker_0: Trying to Create VM #2 in Datacenter_0
0.0: Broker_0: Trying to Create VM #3 in Datacenter_1
0.0: Broker_0: Trying to Create VM #4 in Datacenter_2
0.0: Broker_0: Trying to Create VM #5 in Datacenter_3
0.0: Broker_0: Trying to Create VM #6 in Datacenter_4
0.1: Broker_0: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker_0: VM #3 has been created in Datacenter #3, Host #0
0.1: Broker_0: VM #4 has been created in Datacenter #4, Host #0
0.1: Broker_0: VM #5 has been created in Datacenter #5, Host #0
0.1: Broker_0: VM #6 has been created in Datacenter #6, Host #0
0.1: Broker_0: Sending cloudlet 0 to VM #3
0.1: Broker_0: Sending cloudlet 1 to VM #2
0.1: Broker_0: Sending cloudlet 2 to VM #2
0.1: Broker_0: Sending cloudlet 3 to VM #6
0.1: Broker_0: Sending cloudlet 4 to VM #6
0.1: Broker_0: Sending cloudlet 5 to VM #2
0.1: Broker_0: Sending cloudlet 6 to VM #3
0.1: Broker_0: Sending cloudlet 7 to VM #5
0.1: Broker_0: Sending cloudlet 8 to VM #3
0.1: Broker_0: Sending cloudlet 9 to VM #6
0.1: Broker_0: Sending cloudlet 10 to VM #2
0.1: Broker_0: Sending cloudlet 11 to VM #4
0.1: Broker_0: Sending cloudlet 12 to VM #2
0.1: Broker_0: Sending cloudlet 13 to VM #6
0.1: Broker_0: Sending cloudlet 14 to VM #5
0.1: Broker_0: Sending cloudlet 15 to VM #4
0.1: Broker_0: Sending cloudlet 16 to VM #3
0.1: Broker_0: Sending cloudlet 17 to VM #5
0.1: Broker_0: Sending cloudlet 18 to VM #3
0.1: Broker_0: Sending cloudlet 19 to VM #2
0.1: Broker_0: Sending cloudlet 20 to VM #3
0.1: Broker_0: Sending cloudlet 21 to VM #5
0.1: Broker_0: Sending cloudlet 22 to VM #6
0.1: Broker_0: Sending cloudlet 23 to VM #4
0.1: Broker_0: Sending cloudlet 24 to VM #5
0.1: Broker_0: Sending cloudlet 25 to VM #2
0.1: Broker_0: Sending cloudlet 26 to VM #3
0.1: Broker_0: Sending cloudlet 27 to VM #6
0.1: Broker_0: Sending cloudlet 28 to VM #6
0.1: Broker_0: Sending cloudlet 29 to VM #2
2674.024: Broker_0: Cloudlet 1 received
2761.18: Broker_0: Cloudlet 11 received
2810.9159999999997: Broker_0: Cloudlet 7 received
2850.116: Broker_0: Cloudlet 3 received
3235.624: Broker_0: Cloudlet 0 received
3909.456: Broker_0: Cloudlet 15 received

```

```

datacenter[i] = DatacenterCreator.createDatacenter("Datacenter_" + i);
}

//Third step: Create Broker
SJFDatacenterBroker broker = createBroker("Broker_0");
int brokerId = broker.getId();

//Fourth step: Create VMs and Cloudlets and send them to broker
vmList = createVM(brokerId, Constants.NO_OF_DATA_CENTERS);
cloudletList = createCloudlet(brokerId, Constants.NO_OF_TASKS, 0);

broker.submitVmList(vmList);
broker.submitCloudletList(cloudletList);

// Fifth step: Starts the simulation
CloudSim.startSimulation();

// Final step: Print results when simulation is over
List<Cloudlet> newList = broker.getCloudletReceivedList();
//newList.addAll(globalBroker.getBroker().getCloudletReceivedList());

CloudSim.stopSimulation();

printCloudletList(newList);

Log.println(SJF_Scheduler.class.getName() + " finished!");
} catch (Exception e) {
    e.printStackTrace();
    Log.println("The simulation has been terminated due to an unexpected error");
}
}

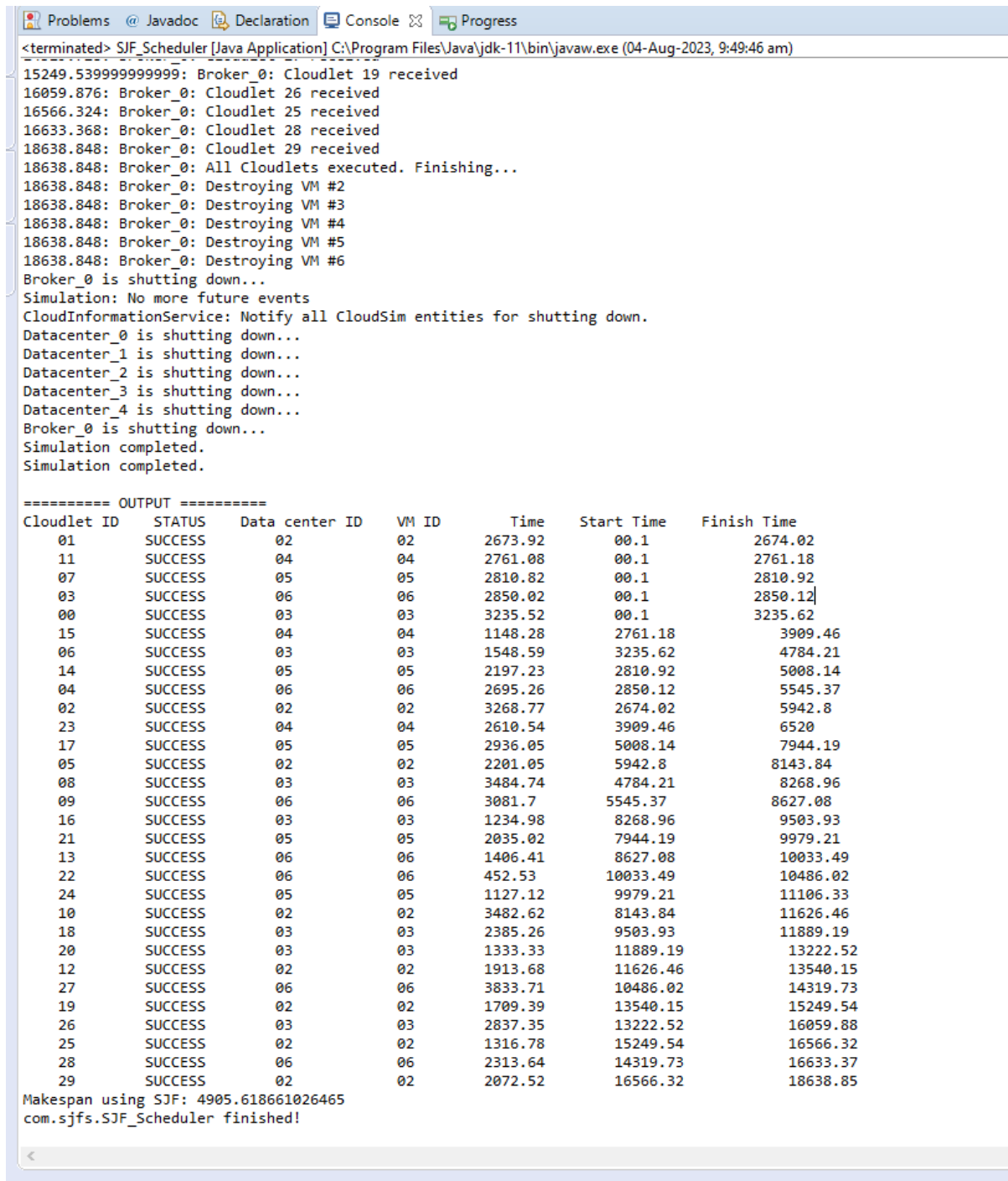
private static SJFDatacenterBroker createBroker(String name) throws Exception {
    return new SJFDatacenterBroker(name);
}

/**
 * Prints the Cloudlet objects
 *
 * @param list list of Cloudlets
 */
private static void printCloudletList(List<Cloudlet> list) {
    int size = list.size();
    Cloudlet cloudlet;

    String indent = " ";
    Log.println();
    Log.println("===== OUTPUT =====");
    Log.println("Cloudlet ID" + indent + "STATUS" +
        indent + "Data center ID" +
        indent + "VM ID" +
        indent + indent + "Time" +
        indent + "Start Time" +
        indent + "Finish Time");

    DecimalFormat dft = new DecimalFormat("###.##");
    dft.setMinimumIntegerDigits(2);

```



```

Problems @ Javadoc Declaration Console Progress
<terminated> SJF_Scheduler [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (04-Aug-2023, 9:49:46 am)
15249.539999999999: Broker_0: Cloudlet 19 received
16059.876: Broker_0: Cloudlet 26 received
16566.324: Broker_0: Cloudlet 25 received
16633.368: Broker_0: Cloudlet 28 received
18638.848: Broker_0: Cloudlet 29 received
18638.848: Broker_0: All Cloudlets executed. Finishing...
18638.848: Broker_0: Destroying VM #2
18638.848: Broker_0: Destroying VM #3
18638.848: Broker_0: Destroying VM #4
18638.848: Broker_0: Destroying VM #5
18638.848: Broker_0: Destroying VM #6
Broker_0 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Datacenter_2 is shutting down...
Datacenter_3 is shutting down...
Datacenter_4 is shutting down...
Broker_0 is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
01            SUCCESS    02             02     2673.92    00.1         2674.02
11            SUCCESS    04             04     2761.08    00.1         2761.18
07            SUCCESS    05             05     2810.82    00.1         2810.92
03            SUCCESS    06             06     2850.02    00.1         2850.12
00            SUCCESS    03             03     3235.52    00.1         3235.62
15            SUCCESS    04             04     1148.28    2761.18      3909.46
06            SUCCESS    03             03     1548.59    3235.62      4784.21
14            SUCCESS    05             05     2197.23    2810.92      5008.14
04            SUCCESS    06             06     2695.26    2850.12      5545.37
02            SUCCESS    02             02     3268.77    2674.02      5942.8
23            SUCCESS    04             04     2610.54    3909.46      6520
17            SUCCESS    05             05     2936.05    5008.14      7944.19
05            SUCCESS    02             02     2201.05    5942.8       8143.84
08            SUCCESS    03             03     3484.74    4784.21      8268.96
09            SUCCESS    06             06     3081.7     5545.37      8627.08
16            SUCCESS    03             03     1234.98    8268.96      9503.93
21            SUCCESS    05             05     2035.02    7944.19      9979.21
13            SUCCESS    06             06     1406.41    8627.08     10033.49
22            SUCCESS    06             06     452.53     10033.49    10486.02
24            SUCCESS    05             05     1127.12    9979.21     11106.33
10            SUCCESS    02             02     3482.62    8143.84     11626.46
18            SUCCESS    03             03     2385.26    9503.93     11889.19
20            SUCCESS    03             03     1333.33    11889.19    13222.52
12            SUCCESS    02             02     1913.68    11626.46    13540.15
27            SUCCESS    06             06     3833.71    10486.02    14319.73
19            SUCCESS    02             02     1709.39    13540.15    15249.54
26            SUCCESS    03             03     2837.35    13222.52    16059.88
25            SUCCESS    02             02     1316.78    15249.54    16566.32
28            SUCCESS    06             06     2313.64    14319.73    16633.37
29            SUCCESS    02             02     2072.52    16566.32    18638.85

Makespan using SJF: 4905.618661026465
com.sjfs.SJF_Scheduler finished!

```

```

for (int i = 0; i < size; i++) {
    cloudlet = list.get(i);
    Log.print(indent + dft.format(cloudlet.getCloudletId()) + indent + indent);

    if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
        Log.print("SUCCESS");

        Log.println(indent + indent + dft.format(cloudlet.getResourceId()) +
            indent + indent + indent + dft.format(cloudlet.getVmId()) +
            indent + indent + dft.format(cloudlet.getActualCPUTime()) +
            indent + indent + dft.format(cloudlet.getExecStartTime()) +
            indent + indent + indent + dft.format(cloudlet.getFinishTime()));
    }
}

double makespan = calcMakespan(list);
Log.println("Makespan using SJF: " + makespan);
}

private static double calcMakespan(List<Cloudlet> list) {
    double makespan = 0;
    double[] dcWorkingTime = new double[Constants.NO_OF_DATA_CENTERS];

    for (int i = 0; i < Constants.NO_OF_TASKS; i++) {
        int dcId = list.get(i).getVmId() % Constants.NO_OF_DATA_CENTERS;
        if (dcWorkingTime[dcId] != 0) --dcWorkingTime[dcId];
        dcWorkingTime[dcId] += execMatrix[i][dcId] + commMatrix[i][dcId];
        makespan = Math.max(makespan, dcWorkingTime[dcId]);
    }
    return makespan;
}
}

```

#### SJFDatacenterBroker.java

```

package com.sjfs;

import org.cloudbus.cloudsim.*;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.core.CloudSimTags;
import org.cloudbus.cloudsim.core.SimEvent;
import java.util.ArrayList;
import java.util.List;

public class SJFDatacenterBroker extends DatacenterBroker {

    SJFDatacenterBroker(String name) throws Exception {
        super(name);
    }

    public void scheduleTaskstoVms() {
        int reqTasks = cloudletList.size();
        int reqVms = vmList.size();
        Vm vm = vmList.get(0);

        for (int i = 0; i < reqTasks; i++) {
            bindCloudletToVm(i, (i % reqVms));
        }
    }
}

```

```

        System.out.println("Task" + cloudletList.get(i).getCloudletId() + " is bound with VM" + vmList.get(i %
reqVms).getId());
    }

    //System.out.println("reqTasks: "+ reqTasks);

    ArrayList<Cloudlet> list = new ArrayList<Cloudlet>();
    for (Cloudlet cloudlet : getCloudletReceivedList()) {
        list.add(cloudlet);
    }

    //setCloudletReceivedList(null);

    Cloudlet[] list2 = list.toArray(new Cloudlet[list.size()]);

    //System.out.println("size :"+list.size());

    Cloudlet temp = null;

    int n = list.size();

    for (int i = 0; i < n; i++) {
        for (int j = 1; j < (n - i); j++) {
            if (list2[j - 1].getCloudletLength() / (vm.getMips() * vm.getNumberOfPes()) > list2[j].getCloudletLength() /
(vm.getMips() * vm.getNumberOfPes())) {
                //swap the elements!
                //swap(list2[j-1], list2[j]);
                temp = list2[j - 1];
                list2[j - 1] = list2[j];
                list2[j] = temp;
            }
            // printNumbers(list2);
        }
    }

    ArrayList<Cloudlet> list3 = new ArrayList<Cloudlet>();

    for (int i = 0; i < list2.length; i++) {
        list3.add(list2[i]);
    }
    //printNumbers(list);

    setCloudletReceivedList(list);

    //System.out.println("\n\tSJFS Broker Schedules\n");
    //System.out.println("\n");
}

public void printNumber(Cloudlet[] list) {
    for (int i = 0; i < list.length; i++) {
        System.out.print(" " + list[i].getCloudletId());
        System.out.println(list[i].getCloudletStatusString());
    }
    System.out.println();
}

```



```

public void printNumbers(ArrayList<Cloudlet> list) {

    for (int i = 0; i < list.size(); i++) {
        System.out.print(" " + list.get(i).getCloudletId());
    }
    System.out.println();
}

@Override
protected void processCloudletReturn(SimEvent ev) {
    Cloudlet cloudlet = (Cloudlet) ev.getData();
    getCloudletReceivedList().add(cloudlet);
    Log.println(CloudSim.clock() + ": " + getName() + ": Cloudlet " + cloudlet.getCloudletId()
        + " received");
    cloudletsSubmitted--;
    if (getCloudletList().size() == 0 && cloudletsSubmitted == 0) {
        scheduleTaskstoVms();
        cloudletExecution(cloudlet);
    }
}

protected void cloudletExecution(Cloudlet cloudlet) {

    if (getCloudletList().size() == 0 && cloudletsSubmitted == 0) { // all cloudlets executed
        Log.println(CloudSim.clock() + ": " + getName() + ": All Cloudlets executed. Finishing...");
        clearDatacenters();
        finishExecution();
    } else { // some cloudlets haven't finished yet
        if (getCloudletList().size() > 0 && cloudletsSubmitted == 0) {
            // all the cloudlets sent finished. It means that some bount
            // cloudlet is waiting its VM be created
            clearDatacenters();
            createVmsInDatacenter(0);
        }
    }
}

@Override
protected void processResourceCharacteristics(SimEvent ev) {
    DatacenterCharacteristics characteristics = (DatacenterCharacteristics) ev.getData();
    getDatacenterCharacteristicsList().put(characteristics.getId(), characteristics);

    if (getDatacenterCharacteristicsList().size() == getDatacenterIdsList().size()) {
        distributeRequestsForNewVmsAcrossDatacenters();
    }
}

protected void distributeRequestsForNewVmsAcrossDatacenters() {
    int numberOfVmsAllocated = 0;
    int i = 0;

    final List<Integer> availableDatacenters = getDatacenterIdsList();

    for (Vm vm : getVmList()) {
        int datacenterId = availableDatacenters.get(i++ % availableDatacenters.size());
        String datacenterName = CloudSim.getEntityName(datacenterId);

```

```
if (!getVmsToDatacentersMap().containsKey(vm.getId())) {  
  
    Log.println(CloudSim.clock() + ": " + getName() + ": Trying to Create VM #" + vm.getId() + " in " +  
datacenterName);  
    sendNow(datacenterId, CloudSimTags.VM_CREATE_ACK, vm);  
    numberOfVmsAllocated++;  
}  
}  
  
setVmsRequested(numberOfVmsAllocated);  
setVmsAcks(0);  
}  
}
```

**RESULT:**

EX NO: 6

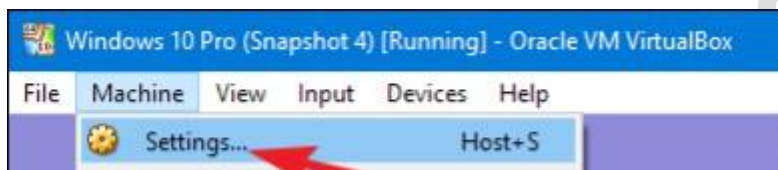
Date:

## COPY FILES OR TRANSFER FROM ONE VIRTUAL MACHINE TO ANOTHER

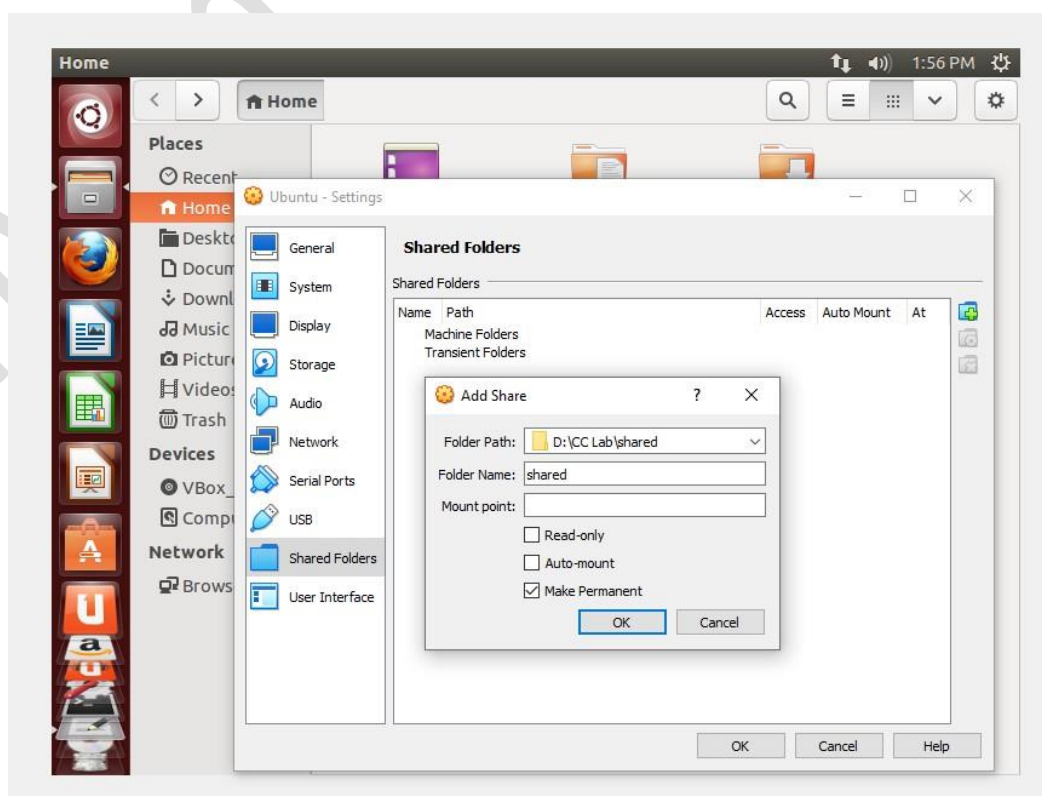
**AIM:**

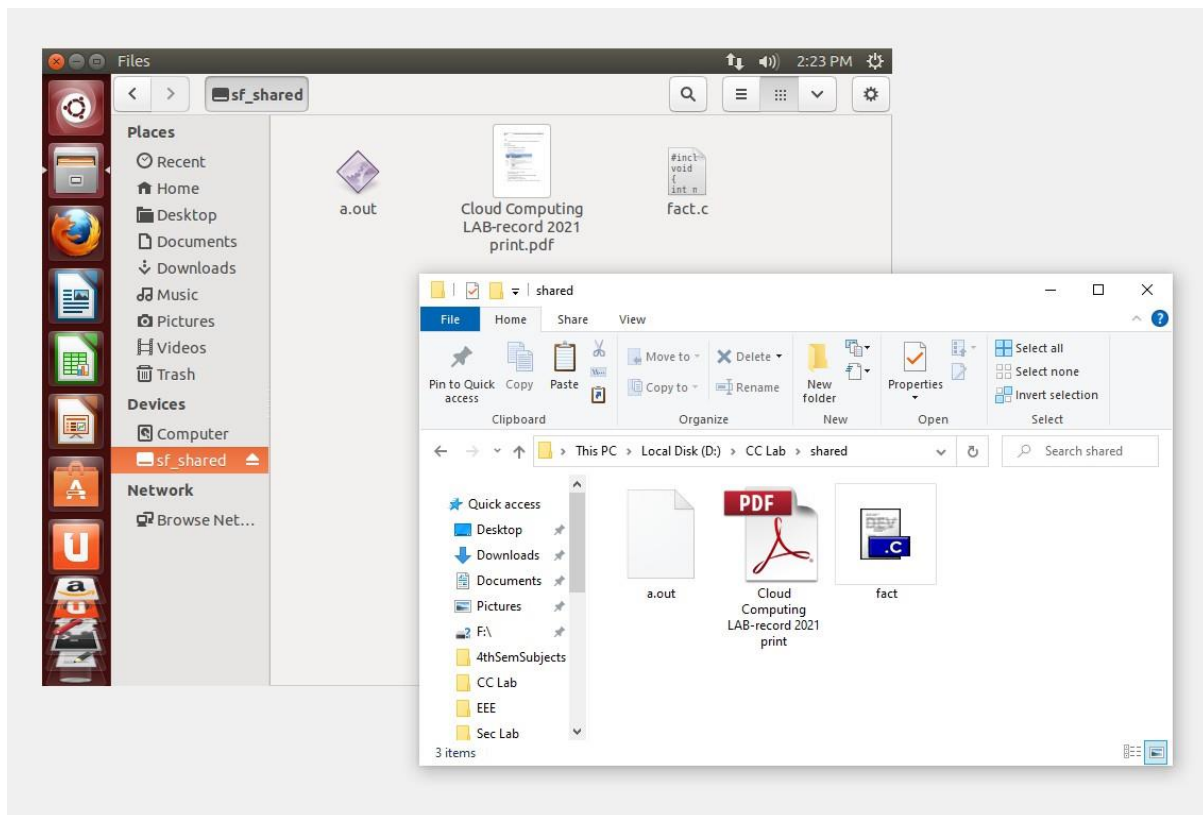
**PROCEDURE:**

1. Create one shared folder in virtual box.
2. VirtualBox's Shared Folders feature works with both Windows and Linux guest operating systems.
3. To use the feature, you first need to install VirtualBox's Guest Additions in the guest virtual machine.
4. With the virtual machine running, click the "Devices" menu and choose the "Insert Guest Additions CD image" option.
5. This inserts a virtual CD that you can use within the guest operating system to install the Guest Additions.
6. After the Guest Additions are installed, open the "Machine" menu and click the "Settings" option.



7. In the "Settings" window, switch to the "Shared Folders" tab.
8. Here you can see any shared folders you've set up.
9. There are two types of shared folders.
10. Machine Folders are permanent folders that are shared until you remove them.
11. Transient Folders are temporary and are automatically removed when you restart or shut down the virtual machine.
12. Click the "Add" button (the folder with a plus on it) to create a new shared folder.



**OUTPUT:**

In the “Add Share” window, you can specify the following:

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

Make all your choices and then hit the “OK” button.

**RESULT:**

EX NO: 7

Date:

## CREATE, DEPLOY AND LAUNCH VIRTUAL MACHINES IN OPENSTACK

AIM:

### REQUIREMENTS:

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

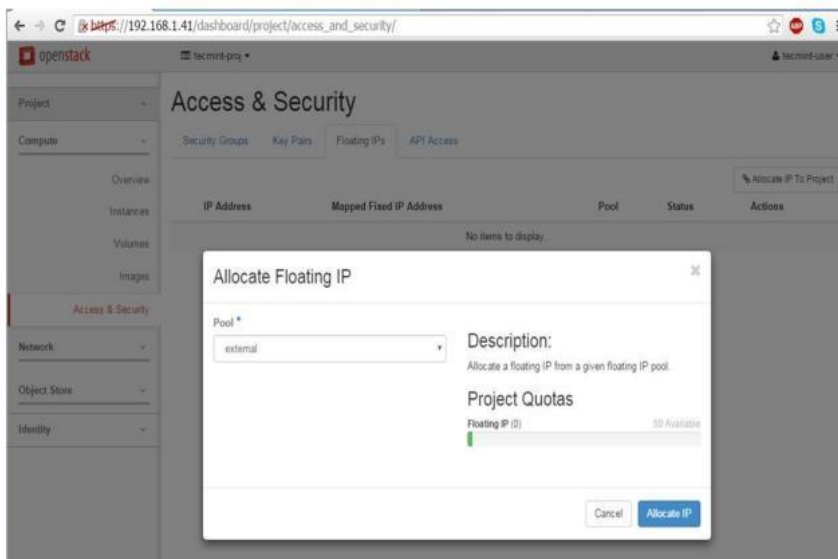
### PROCEDURE:

#### Step 1: Allocate Floating IP to OpenStack

1. Before you deploy an OpenStack image, first you need to assure that all pieces are in place and we'll start by allocating floating IP.
2. Floating IP allows external access from outside networks or internet to an Openstack virtual machine.
3. In order to create floating IPs for your project, login with your user credentials and go to **Project -> Compute -> Access & Security -> Floating IPs** tab and click on Allocate IP to The Project.
4. Choose external Pool and hit on Allocate IP button and the IP address should appear in dashboard.
5. It's a good idea to allocate a Floating IP for each instance you run.



6. Allocate Floating IP to Project in OpenStack.



## 7. Allocate Floating IP to External Pool



## 8. Confirmation of Adding Floating IP.

### Step 2: Create an OpenStack Image

1. OpenStack images are just virtual machines already created by third-parties.
2. You can create your own customized images on your machine by installing an Linux OS in a virtual machine using a virtualization tool, such as KVM, VirtualBox, VMware or Hyper-V.
3. Once you have installed the OS, just convert the file to raw and upload it to your OpenStack cloud infrastructure.

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

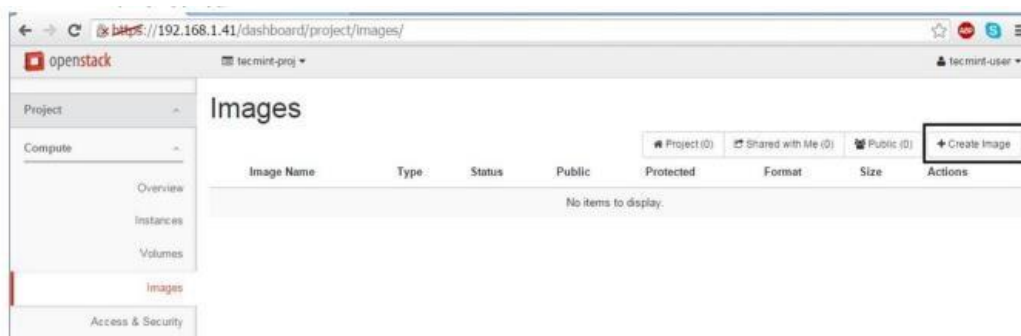
On this guide we'll deploy a test image, for demonstration purposes, based on a lightweight Cirros cloud image which can be obtained by visiting the following link <http://download.cirroscloud.net/0.3.4/>.

The image file can be used directly from the HTTP link or downloaded locally on your machine and uploaded to OpenStack cloud.

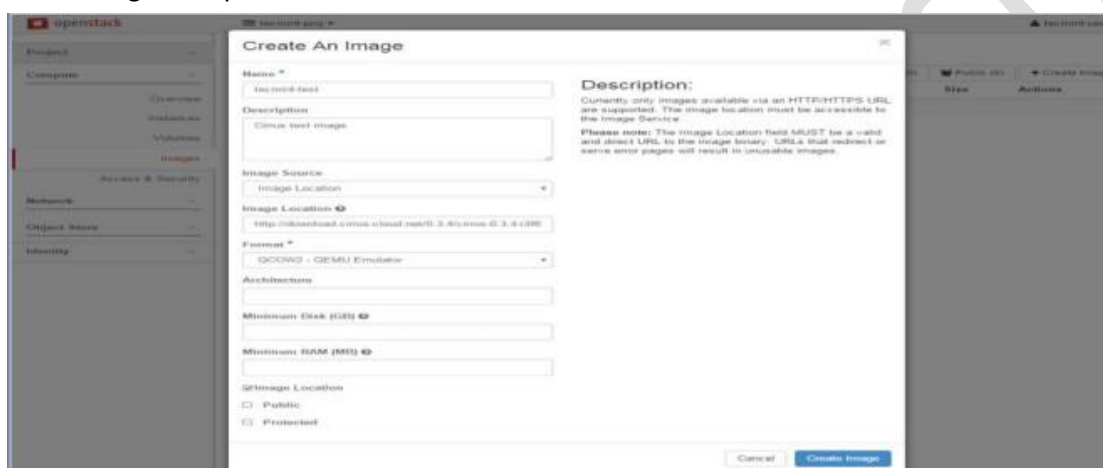
To create an image, go OpenStack web panel and navigate to Project -> Compute -> Images and hit on Create Image button.

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

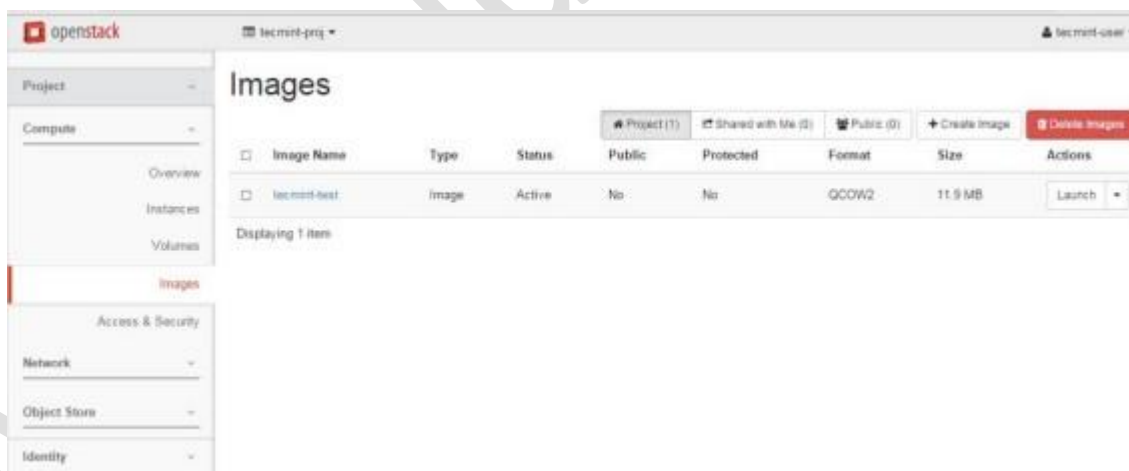
1. Name: tecmint-test
2. Description: Cirros test image
3. Image Source: Image Location #Use Image File if you've downloaded the file locally on your hard disk Image
4. Location: <http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-i386-disk.img>
5. Format: QCOW2 – QEMU Emulator
6. Architecture: leave blank Minimum
7. Disk: leave blank Minimum
8. RAM: leave blank Image
9. Location: checked
10. Public: unchecked
11. Protected: unchecked



### Create Images in OpenStack



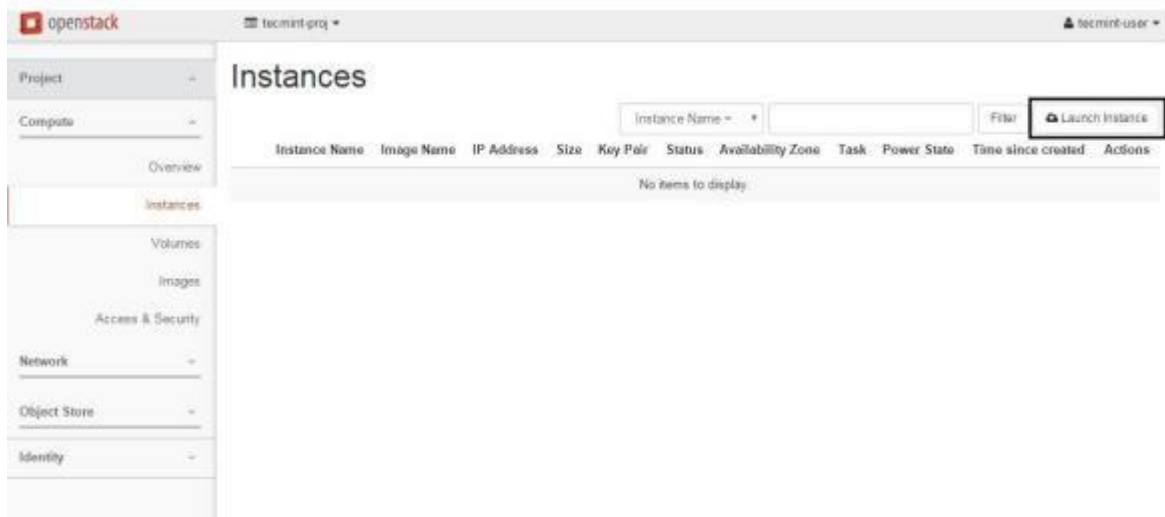
### Add OpenStack Image Details



### Step 3: Launch an Image Instance in OpenStack

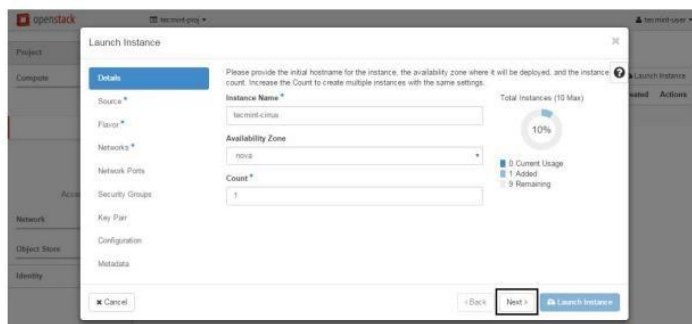
Once you've created an image you're good to go. Now you can run the virtual machine based on the image created earlier in your cloud environment.

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



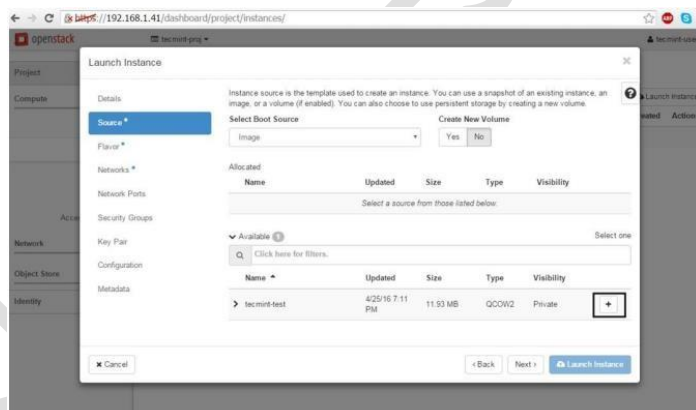
### Launch Image Instance in Openstack

1. On the first screen add a name for your instance, leave the Availability Zone to nova, use one instance count and hit on Next button to continue.
2. Choose a descriptive Instance Name for your instance because this name will be used to form the virtual machine hostname.



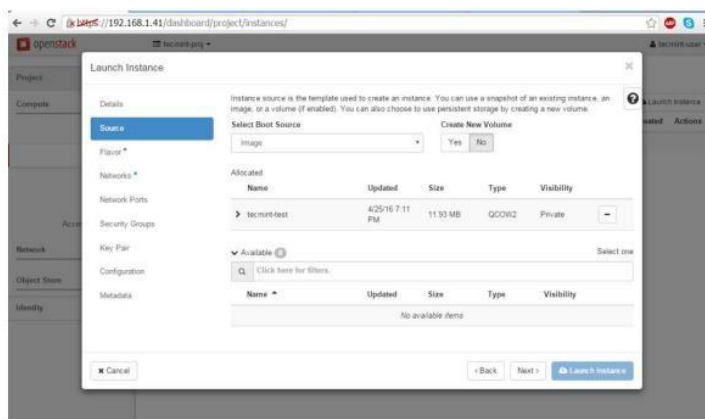
### Add Hostname to OpenStack Instance

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



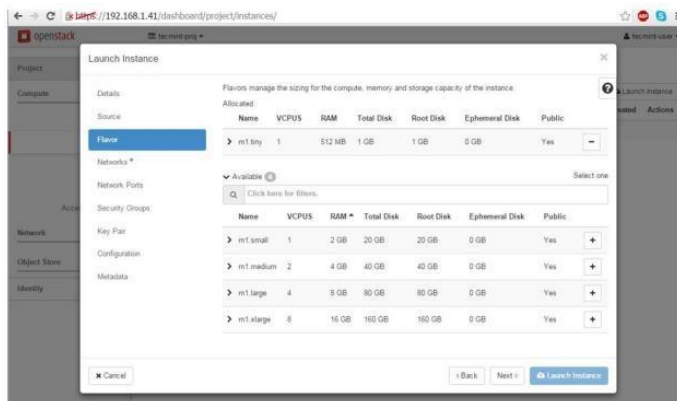
### Select OpenStack Instance Boot Source





Add Cirros Text Image .

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



Add Resources to OpenStack Instance.

OUTPUT:

Instances

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

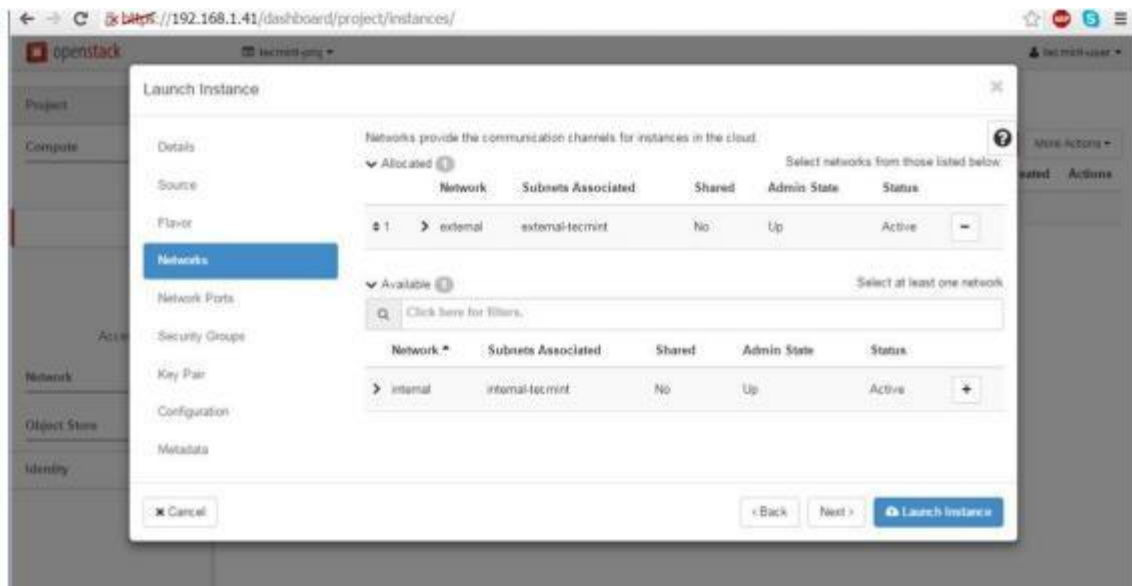
Displaying 1 item

```

192.168.1.5 - PuTTY
C:\> uname -a
Linux tecmini-test 3.2.0-30-virtual #116-Ubuntu SMP Mon Mar 23 17:28:52 UTC 2015
x86_64 GNU/Linux
C:\> ping -c 2 google.com
PING google.com (172.217.20.142): 56 data bytes
64 bytes from 172.217.20.142: seq=0 ttl=55 time=42.433 ms
64 bytes from 172.217.20.142: seq=1 ttl=55 time=39.888 ms

--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 39.888/41.009/42.433 ms
C:\> cat /etc/resolv.conf
search openstacklocal
nameserver 192.168.1.1
nameserver 8.8.8.8
nameserver 8.8.4.4
C:\>
  
```

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



Add Network to OpenStack Instance .

- Once the instance has been started, hit on the right arrow from Create Snapshot menu button and choose Associate Floating IP.
- Select one of the floating IP created earlier and hit on Associate button in order to make the instance reachable from your internal LAN.

**RESULT:**

EX NO: 8

Date:

**INSTALL HADOOP SINGLE NODE CLUSTER****AIM:****PROCEDURE:**

1. Download Hadoop 2.8.0
2. Check either Java 1.8.0 is already installed on your system or not, use "Javac - version" to check Java version
3. If Java is not installed on your system then first install java under "C:\JAVA" Java setup
4. Extract files Hadoop 2.8.0.tar.gz or Hadoop-2.8.0.zip and place under "C:\Hadoop-2.8.0" hadoop
5. Set the path HADOOP\_HOME Environment variable.
6. Set the path JAVA\_HOME Environment variable.
7. Next we set the Hadoop bin directory path and JAVA bin directory path.

**HADOOP CONFIGURATION:**

- a) File C:/Hadoop-2.8.0/etc/hadoop/core-site.xml, paste below xml paragraph and save this file.

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

- b) Rename mapred-site.xml.template" to "mapred-site.xml" and edit this file

C:/Hadoop2.8.0/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

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

- c) Create folder "data" under "C:\Hadoop-2.8.0"

- Create folder "datanode" under "C:\Hadoop-2.8.0\data"
- Create folder "namenode" under "C:\Hadoop-2.8.0\data" data

- d) Edit file C:\Hadoop-2.8.0/etc/hadoop/hdfs-site.xml, paste below xml paragraph and save this file.

`<configuration>``<property>``<name>dfs.replication</name>``<value>1</value>``</property>``<property>``<name>dfs.namenode.name.dir</name>``<value>C:\hadoop-2.8.0\data\namenode</value>``</property>``<property>``<name>dfs.datanode.data.dir</name>``<value>C:\hadoop-2.8.0\data\datanode</value>``</property>``</configuration>`



e) Edit file C:/Hadoop-2.8.0/etc/hadoop/yarn-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

f) Edit file C:/Hadoop-2.8.0/etc/hadoop/yarn-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

g) Edit file C:/Hadoop-2.8.0/etc/hadoop/hadoop-env.cmd by closing the command line "JAVA\_HOME=%JAVA\_HOME%" instead of set "JAVA\_HOME=C:\Java"

### Hadoop Configuration:

- 1) Download file Hadoop Configuration.zip
- 2) Delete file bin on C:\Hadoop-2.8.0\bin, replaced by file bin on file just download (from Hadoop Configuration.zip).
- 3) Open cmd and typing command "hdfs namenode –format". You will see hdfs namenode –format Testing.
- 4) Open cmd and change directory to "C:\Hadoop-2.8.0\sbin" and type "start- all.cmd" to start apache.
- 5) Make sure these apps are running.
  - a) Name node
  - b) Hadoop data node
  - c) YARN Resource Manager
  - d) YARN Node Manager hadoop nodes .
- 6) Open: <http://localhost:8088> .

Open cmd in Administrative mode and move to "C:/Hadoop-2.8.0/sbin" and start cluster

1. Create an input directory in HDFS.  
**hadoop fs –mkdir/input\_dir**
2. Copy the input text file named input\_file.txt in the input directory (input\_dir) of HDFS.  
**hadoop fs –put C:/input\_file.txt/input\_dir**
3. Verify input\_file.txt available in HDFS input directory (input\_dir).  
**hadoop fs –ls/input\_dir/**
4. Verify content of the copied file.  
**hadoop dfs –cat/input\_dir/input\_file.txt**
5. Run MapReduceClient.jar and also provide input and out directories.  
**hadoop jar C:/MapReduceClient.jar wordcount /input\_dir /output\_dir**
6. Verify content for generated output file.  
**hadoop dfs -cat /output\_dir/\***

```

C:\hadoop-2.8.0\sbin>hadoop dfs -cat /output_dir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
23      12
24      6
25      18
26      36
27      12
28      24
29      6
30      24
31      24
32      18
33      6
34      30
35      6
36      12
38      24
39      66
40      18
41      24
42      6
43      12
45      6
cloud   2
manickam      2
C:\hadoop-2.8.0\sbin>

```

**Overview 'localhost:9000' (active)**

<b>Started:</b>	Mon Aug 21 13:46:01 +0530 2023
<b>Version:</b>	2.8.0. r912b7a13d1e97be5db92d3abc627cc29ac0009
<b>Compiled:</b>	Fri Mar 17 09:42:00 +0530 2017 by jdu from branch-2.8.0
<b>Cluster ID:</b>	CID-25afe211-d056-4077-8071-4b416549e669
<b>Block Pool ID:</b>	BP-670754356-192.168.56.1-1692004268364

**Summary**

Security is off.

Safe mode is ON. The reported blocks 0 needs additional 5 blocks to reach the threshold 0.9990 of total blocks 6. The number of live datanodes 0 has reached the minimum number 0. Safe mode will be turned off automatically once the thresholds have been reached.

19 files and directories, 6 blocks = 25 total filesystem object(s).

Heap Memory used 46.33 MB of 129.5 MB Heap Memory. Max Heap Memory is 689 MB.

Non Heap Memory used 40.59 MB of 41.53 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

<b>Configured Capacity:</b>	0 B
<b>DFS Used:</b>	0 B (100%)
<b>Non DFS Used:</b>	0 B
<b>DFS Remaining:</b>	0 B (0%)
<b>Block Pool Used:</b>	0 B (100%)
<b>DataNodes usages% (Min/Median/Max/stdDev):</b>	0.00% / 0.00% / 0.00% / 0.00%
<b>Live Nodes</b>	0 (Decommissioned: 0)
<b>Dead Nodes</b>	0 (Decommissioned: 0)

**RESULT:**