

**M.Sc C.S - II
SEM III
Journal**

Roll No.	056
Name	OJHA ABHISHEK DEVMANI
Subject	Cloud Computing - II



CERTIFICATE

This is here to certify that Mr. OJHA ABHISHEK DEVMANI, Seat Number 056 of M.Sc. II Computer Science, has satisfactorily completed the required number of experiments prescribed by the **THAKUR COLLEGE OF SCIENCE & COMMERCE AUTONOMOUS COLLEGE, PERMANENTLY AFFILIATED TO UNIVERSITY OF MUMBAI** during the academic year 2021 - 2022.

Date: 23-09-2022

Place: Mumbai

Teacher In-Charge

Head of Department

External Examiner

INDEX

Sr. No.	Experiments	Page No	Date
1	Write a program to Execute & check the performance of existing algorithms using	01-15	
2	Write a program to Install a Cloud Analyst and Integrate with Eclipse/Netbeans. Monitor the performance of an Existing Algorithms.	16-27	
3	Write a program Build an application on private cloud	28-43	
4	Write a program to Demonstrate any Cloud Monitoring tool.	44-45	
5	Write a program to Evaluate a Private IAAS Cloud .	46-63	
6	Write a program to Implement FOSS-Cloud Functionality - VDI (Virtual Desktop Infrastructure).	64-75	
7	Write a program to Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS)	76-83	
8	Write a program to Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS).	84-84	
9	Write a program to Implement FOSS-Cloud Functionality - VSI Software as a Service (SaaS)	85-86	
10	Write a program to Explore GCP / AWS / Azure	87-92	

Practical No: 1

Aim : Execute & check the performance of existing algorithms using CloudSim.

Theory:

CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the CLOUDS Lab organization and is written entirely in Java. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.

For example, if you were to deploy an application or a website on the cloud and wanted to test the services and load that your product can handle and also tune its performance to overcome bottlenecks before risking deployment, then such evaluations could be performed by simply coding a simulation of that environment with the help of various flexible and scalable classes provided by the CloudSim package, free of cost.

Download CloudSim:

<http://code.google.com/p/cloudsim/downloads>

Cloud Computing - II

The screenshot shows the Google Code Archive interface. The top navigation bar includes 'Projects', 'Search', and 'About'. The main content area is for the 'cloudsim' project. On the left, there are navigation links for 'Project', 'Source', 'Issues', and 'Wikis'. The 'Downloads' section is expanded, showing a list of files:

File	Summary + Labels	Uploaded	Size
cloudsim-3.0.3.tar.gz	CloudSim 3.0.3: bug fix release Featured Type-Package Op Sys-All	May 3, 2013	9.9MB
cloudsim-3.0.3.zip	CloudSim 3.0.3: bug fix release Featured Type-Package Op Sys-All	May 3, 2013	13.05MB
cloudsim-3.0.2.tar.gz	CloudSim 3.0.2: bug fix release Type-Package Op Sys-All	Nov 7, 2012	9.9MB
cloudsim-3.0.2.zip	CloudSim 3.0.2: bug fix release Type-Package Op Sys-All	Nov 7, 2012	13.05MB
cloudsim-3.0.1.tar.gz	CloudSim 3.0.1: bug fix release Type-Package Op Sys-All	Oct 17, 2012	9.89MB
cloudsim-3.0.1.zip	CloudSim 3.0.1: bug fix release Type-Package Op Sys-All	Oct 17, 2012	13.04MB
cloudsim-3.0.tar.gz	CloudSim 3.0 Type-Package Op Sys-All	Jan 11, 2012	9.89MB

The URL at the bottom of the page is <https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/cloudsim/cloudsim-3.0.3.tar.gz>.

Select 1st option

The screenshot shows the Google Code Archive interface with a download dialog box overlaid. The dialog is titled 'Opening cloudsim-3.0.3.tar.gz' and contains the following text:

You have chosen to open:
 [cloudsim-3.0.3.tar.gz](#)
which is: WinRAR archive (9.9 MB)
from: <https://storage.googleapis.com>

What should Firefox do with this file?

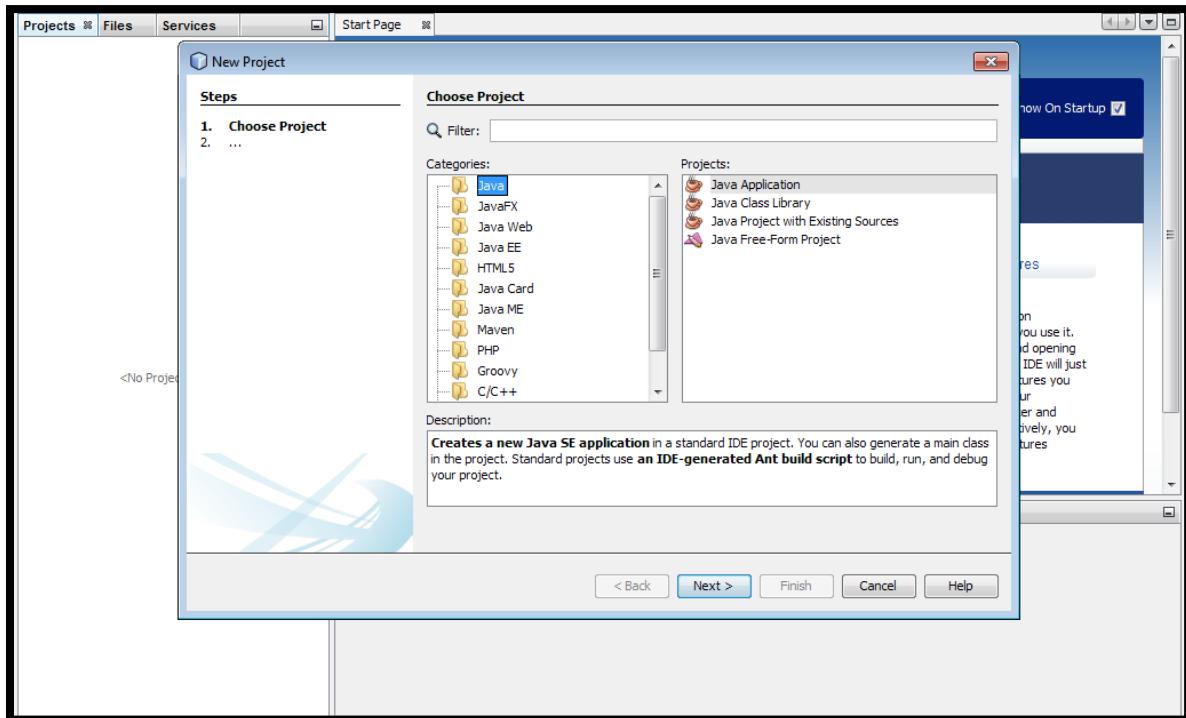
Open with WinRAR archiver (default)
 Save File

Do this automatically for files like this from now on.

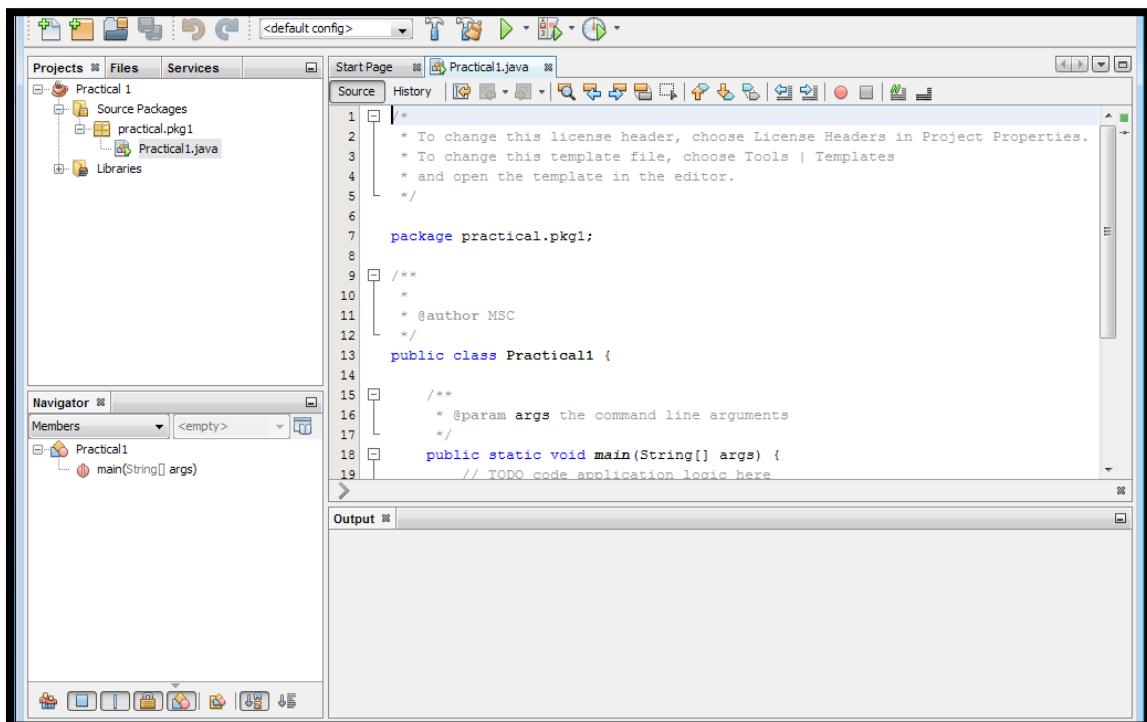
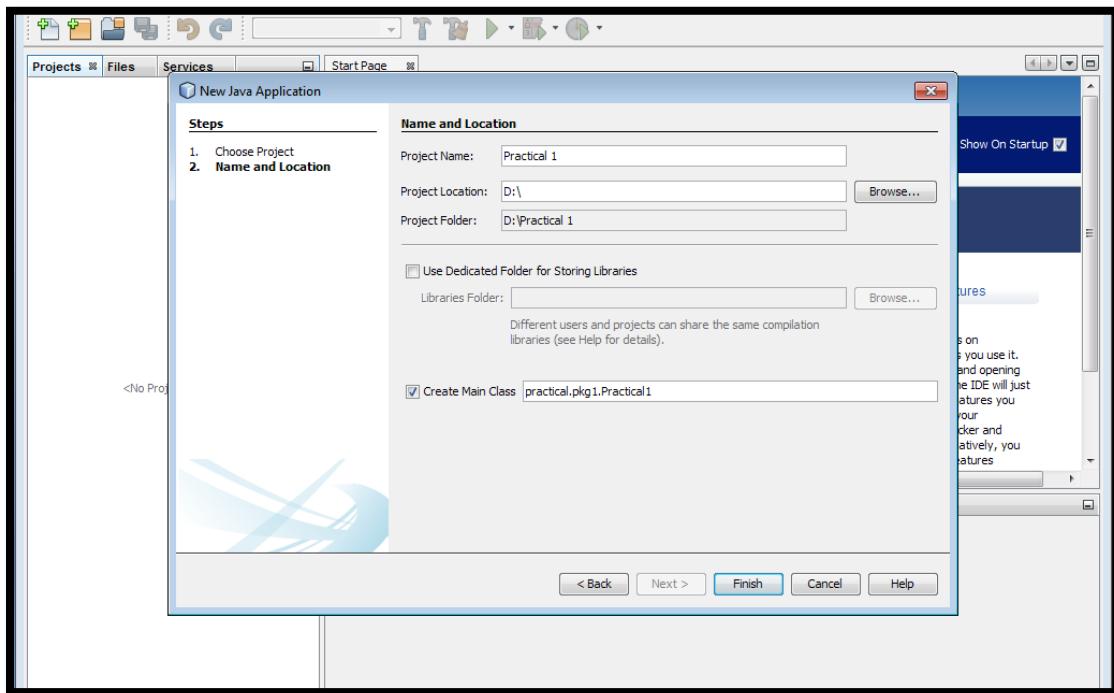
OK **Cancel**

The background table of files is partially visible on the right side of the dialog.

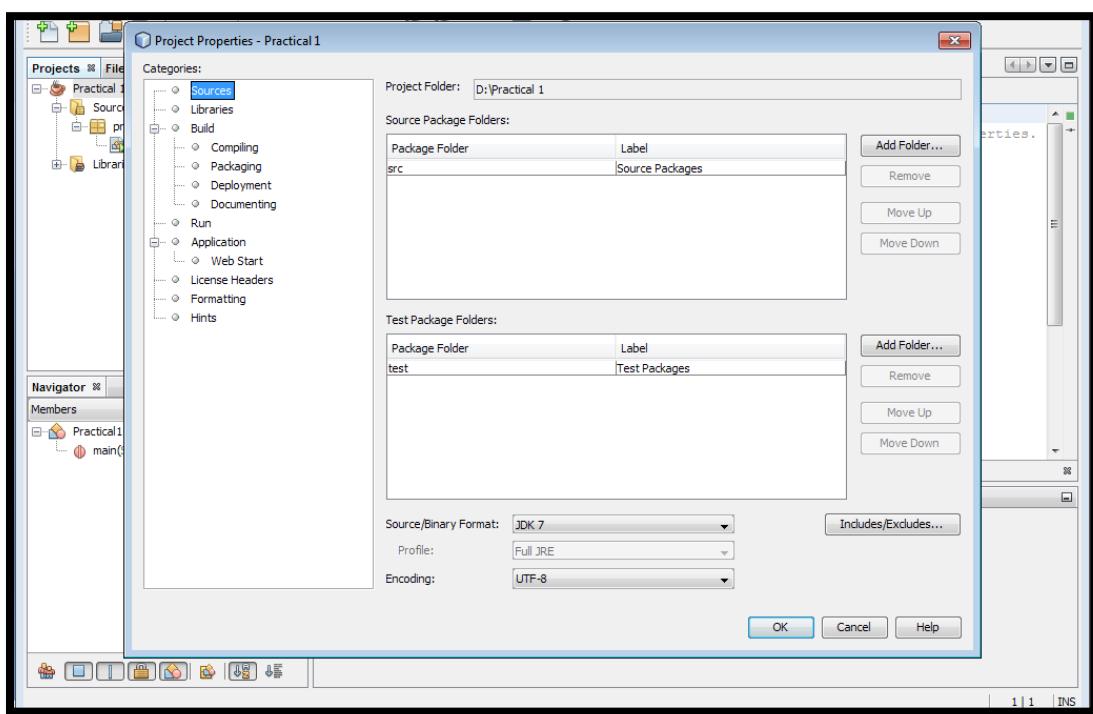
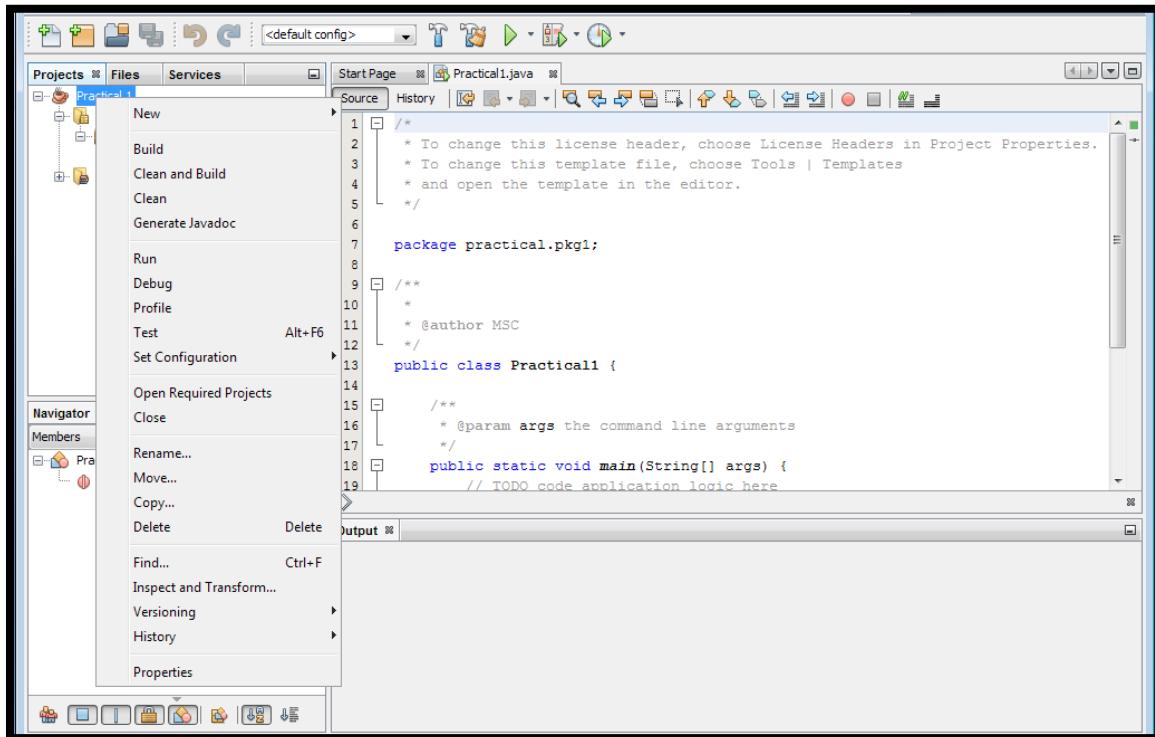
Cloud Computing - II



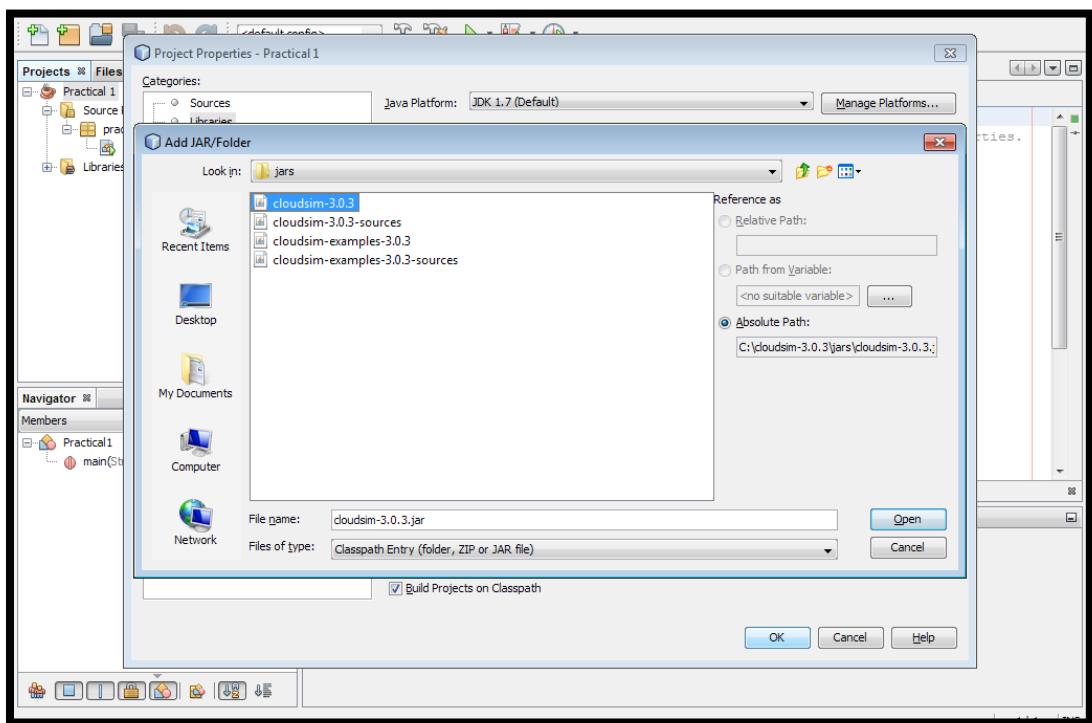
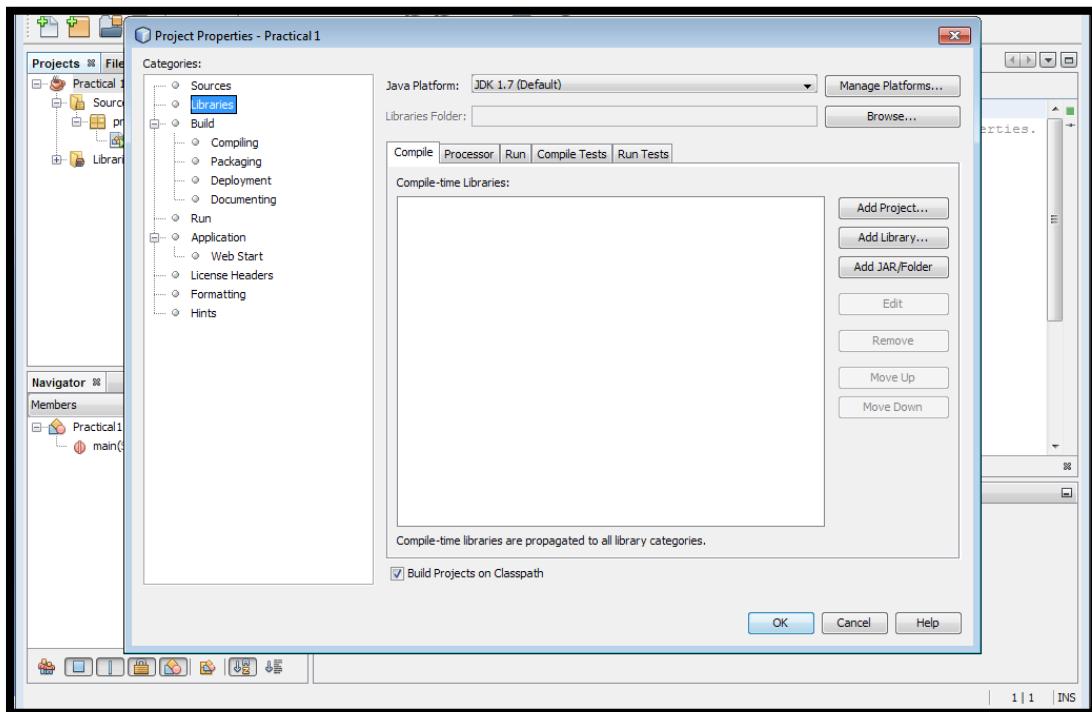
Cloud Computing - II



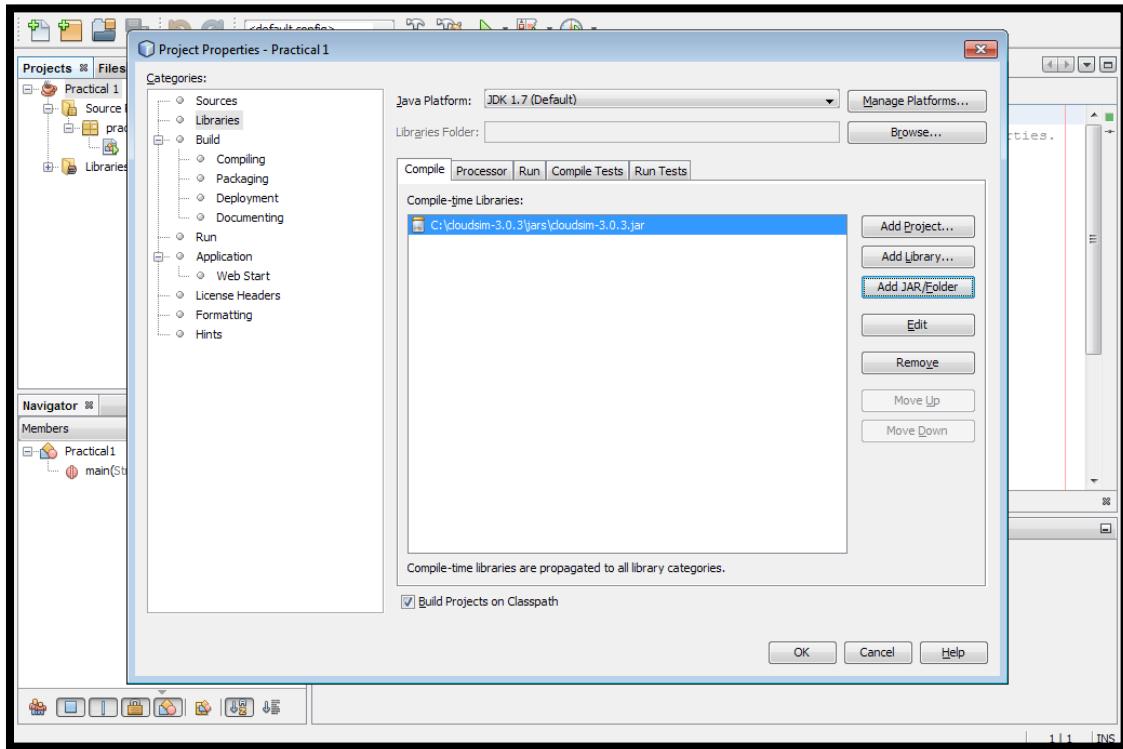
Cloud Computing - II



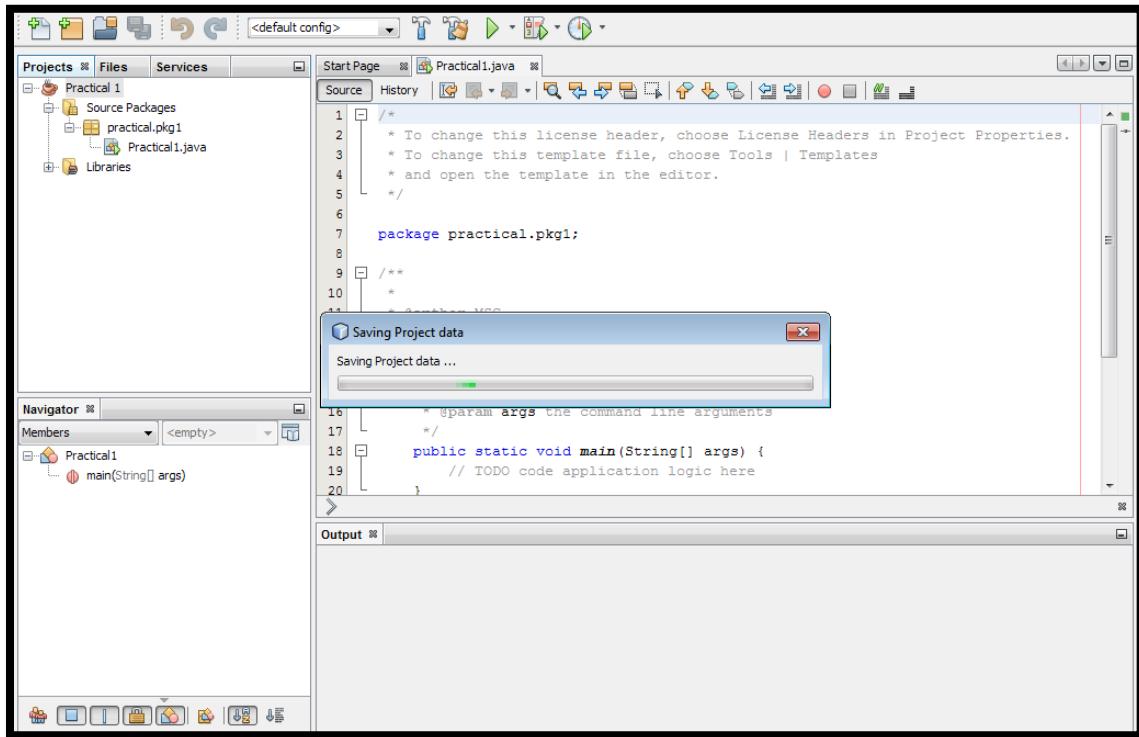
Cloud Computing - II



Cloud Computing - II



Cloud Computing - II



Code:

```
package practical.pkg1;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;
import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
import org.cloudbus.cloudsim.UtilizationModel;
```

Cloud Computing - II

```
import org.cloudbus.cloudsim.UtilizationModelFull;
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

/**
 * A simple example showing how to create a datacenter with one host and run one
 * cloudlet on it.
 */
public class Practical1 {

    /** The cloudlet list. */
    private static List<Cloudlet> cloudletList;

    /** The vmlist. */
    private static List<Vm> vmlist;

    /**
     * Creates main() to run this example.
     *
     * @param args the args
     */
    @SuppressWarnings("unused")
    public static void main(String[] args) {
        Log.printLine("Starting CloudSimExample1...");
        try {
            // First step: Initialize the CloudSim package. It should be called
            // before creating any entities.
            int num_user = 1; // number of cloud users
```

Cloud Computing - II

```
Calendar calendar = Calendar.getInstance();

boolean trace_flag = false; // mean trace events

// Initialize the CloudSim library

CloudSim.init(num_user, calendar, trace_flag);

// Second step: Create Datacenters

// Datacenters are the resource providers in CloudSim. We need at

// list one of them to run a CloudSim simulation

Datacenter datacenter0 = createDatacenter("Datacenter_0");

// Third step: Create Broker

DatacenterBroker broker = createBroker();

int brokerId = broker.getId();

// Fourth step: Create one virtual machine

vmlist = new ArrayList<Vm>();

// VM description

int vmid = 0;

int mips = 1000;

long size = 10000; // image size (MB)

int ram = 512; // vm memory (MB)

long bw = 1000;

int pesNumber = 1; // number of cpus

String vmm = "Xen"; // VMM name

// create VM

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

CloudletSchedulerTimeShared());

// add the VM to the vmList

vmlist.add(vm);

// submit vm list to the broker

broker.submitVmList(vmlist);

// Fifth step: Create one Cloudlet

cloudletList = new ArrayList<Cloudlet>();

// Cloudlet properties

int id = 0;

long length = 400000;
```

Cloud Computing - II

```
long fileSize = 300;
long outputSize = 300;
UtilizationModel utilizationModel = new UtilizationModelFull();
Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel,
utilizationModel, utilizationModel);
cloudlet.setUserId(brokerId);
cloudlet.setVmId(vmid);
// add the cloudlet to the list
cloudletList.add(cloudlet);
// submit cloudlet list to the broker
broker.submitCloudletList(cloudletList);
// Sixth step: Starts the simulation
CloudSim.startSimulation();
CloudSim.stopSimulation();
//Final step: Print results when simulation is over
List<Cloudlet> newList = broker.getCloudletReceivedList();
printCloudletList(newList);

Log.println("CloudSimExample1 finished!");
} catch (Exception e) {
    e.printStackTrace();
    Log.println("Unwanted errors happen");
}
/**
 * Creates the datacenter.
 *
 * @param name the name
 *
 * @return the datacenter
 */
private static Datacenter createDatacenter(String name) {
```

Cloud Computing - II

```
// Here are the steps needed to create a PowerDatacenter:  
  
// 1. We need to create a list to store  
// our machine  
  
List<Host> hostList = new ArrayList<Host>();  
  
// 2. A Machine contains one or more PEs or CPUs/Cores.  
// In this example, it will have only one core.  
  
List<Pe> peList = new ArrayList<Pe>();  
  
int mips = 1000;  
  
// 3. Create PEs and add these into a list.  
  
peList.add(new Pe(0, new PeProvisionerSimple(mips))); // need to store Pe id and MIPS Rating  
  
// 4. Create Host with its id and list of PEs and add them to the list  
// of machines  
  
int hostId = 0;  
  
int ram = 2048; // host memory (MB)  
  
long storage = 1000000; // host storage  
  
int bw = 10000;  
  
hostList.add(  
    new Host(  
        hostId,  
        new RamProvisionerSimple(ram),  
        new BwProvisionerSimple(bw),  
        storage,  
        peList,  
        new VmSchedulerTimeShared(peList)  
    )  
); // This is our machine  
  
// 5. Create a DatacenterCharacteristics object that stores the  
// properties of a data center: architecture, OS, list of  
// Machines, allocation policy: time- or space-shared, time zone  
// and its price (G$/Pe time unit).  
  
String arch = "x86"; // system architecture  
  
String os = "Linux"; // operating system
```

Cloud Computing - II

```
String vmm = "Xen";
double time_zone = 10.0; // time zone this resource located
double cost = 3.0; // the cost of using processing in this resource
double costPerMem = 0.05; // the cost of using memory in this resource
double costPerStorage = 0.001; // the cost of using storage in this
                             // resource
double costPerBw = 0.0; // the cost of using bw in this resource
LinkedList<Storage> storageList = new LinkedList<Storage>(); // we are not adding SAN
                                                               // devices by now

DatacenterCharacteristics characteristics = new DatacenterCharacteristics(
    arch, os, vmm, hostList, time_zone, cost, costPerMem,
    costPerStorage, costPerBw);

// 6. Finally, we need to create a PowerDatacenter object.
Datacenter datacenter = null;
try {
    datacenter = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList),
storageList, 0);
} catch (Exception e) {
    e.printStackTrace();
}
return datacenter;
}

// We strongly encourage users to develop their own broker policies, to
// submit vms and cloudlets according
// to the specific rules of the simulated scenario
/**
 * Creates the broker.
 *
 * @return the datacenter broker
 */
private static DatacenterBroker createBroker() {
    DatacenterBroker broker = null;
    try {
        broker = new DatacenterBroker("Broker");
    }
}
```

Cloud Computing - II

```
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
        return broker;
    }

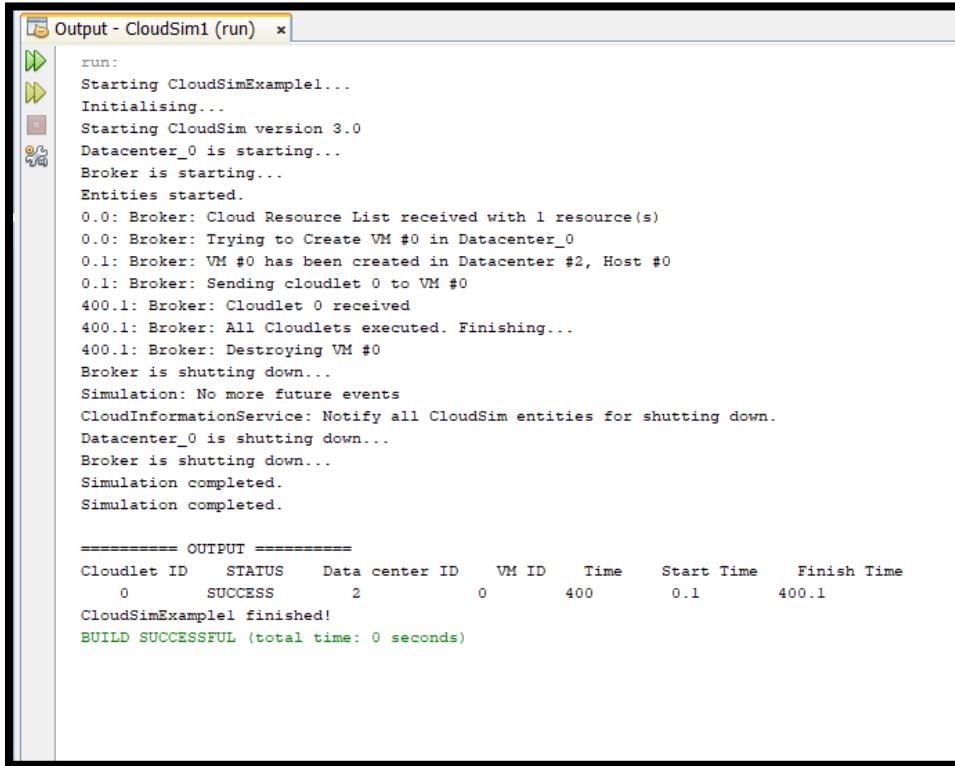
    // Prints the Cloudlet objects.

    //@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 + "Time" + indent
                + "Start Time" + indent + "Finish Time");
        DecimalFormat dft = new DecimalFormat("###.##");
        for (int i = 0; i < size; i++) {
            cloudlet = list.get(i);
            Log.print(indent + cloudlet.getCloudletId() + indent + indent);
            if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
                Log.print("SUCCESS");
                Log.println(indent + indent + cloudlet.getResourceId()
                        + indent + indent + indent + cloudlet.getVmId()
                        + indent + indent
                        + dft.format(cloudlet.getActualCPUTime()) + indent
                        + indent + dft.format(cloudlet.getExecStartTime())
                        + indent + indent
                        + dft.format(cloudlet.getFinishTime()));
            }
        }
    }
}
```

Cloud Computing - II

Output:



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

=====
Cloudlet ID      STATUS      Data center ID      VM ID      Time      Start Time      Finish Time
      0          SUCCESS           2              0       400        0.1        400.1
CloudSimExample1 finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

Conclusion: Successfully executed and checked the performance of algorithm using CloudSim.

Practical No 2

Aim : Install a Cloud Analyst and Integrate with Eclipse/Netbeans. Monitor the performance of an Existing Algorithms.

Theory:

CloudAnalyst

Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. In this tool, communities of users and data centers supporting the social networks are characterized and, based on their location; parameters such as user experience while using the social network application and load on the data center are obtained/logged

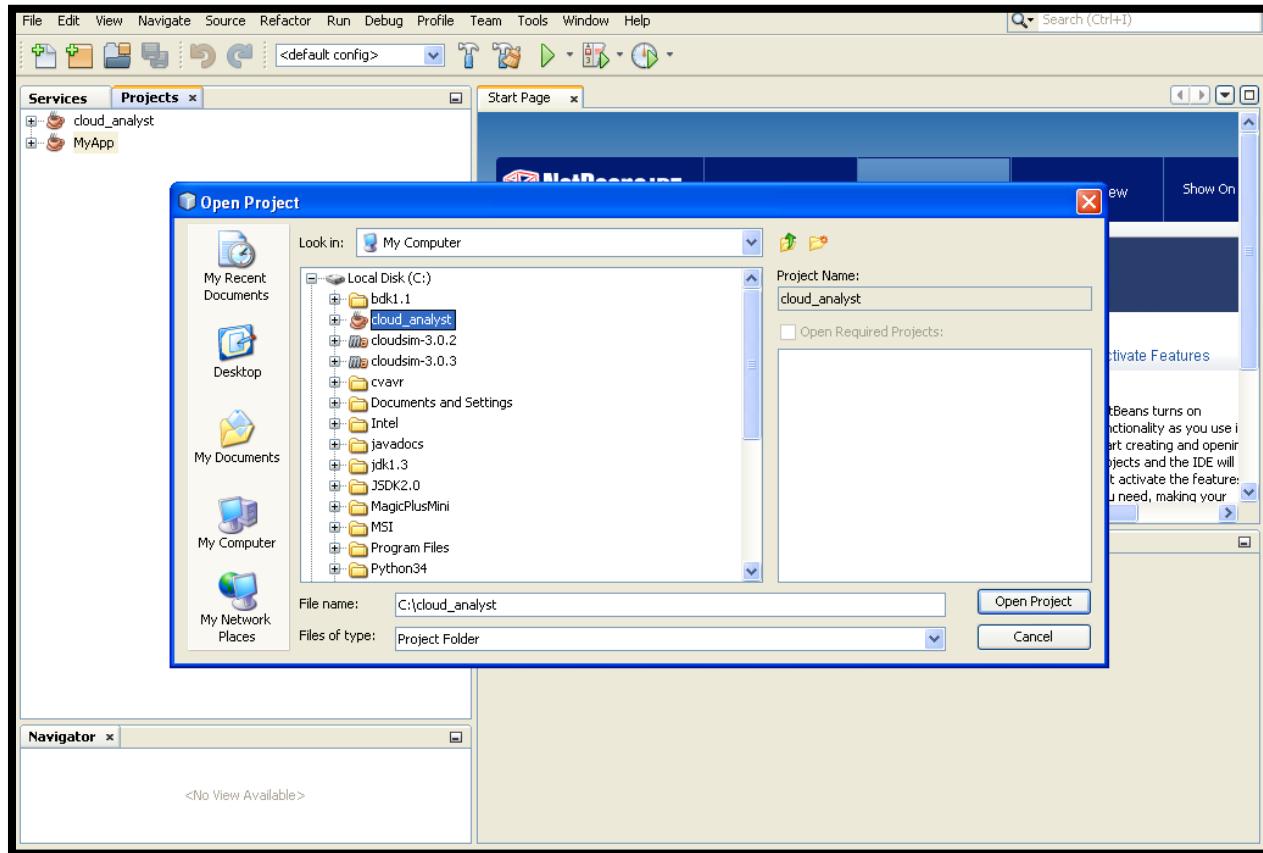
Installation is just download cloud_analyst from

<https://sourceforge.net/projects/cloudanalystnetbeans/>

and copy in drivethen unzip it to use

After installation.....In netbeans7.4.....To run it simply go to file->open project --> simply browse the unzipped folder

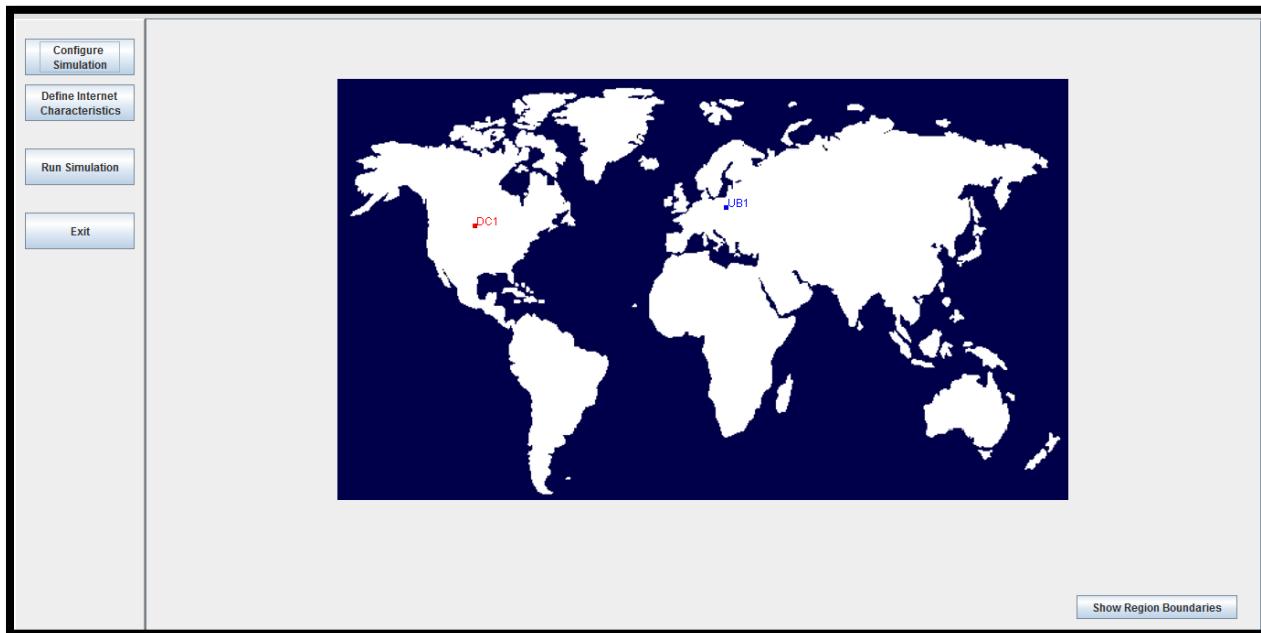
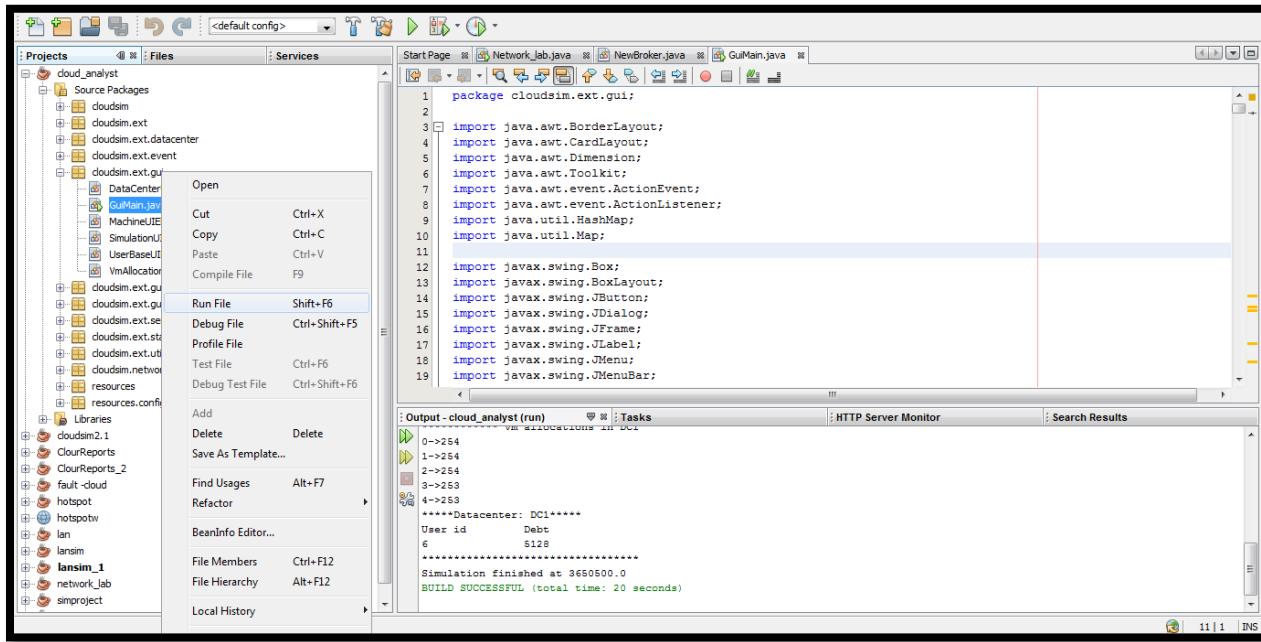
Cloud Computing - II



Expand source package folder inside which open cloudsim.ext.gui

right click on the GuiMain.java and click run

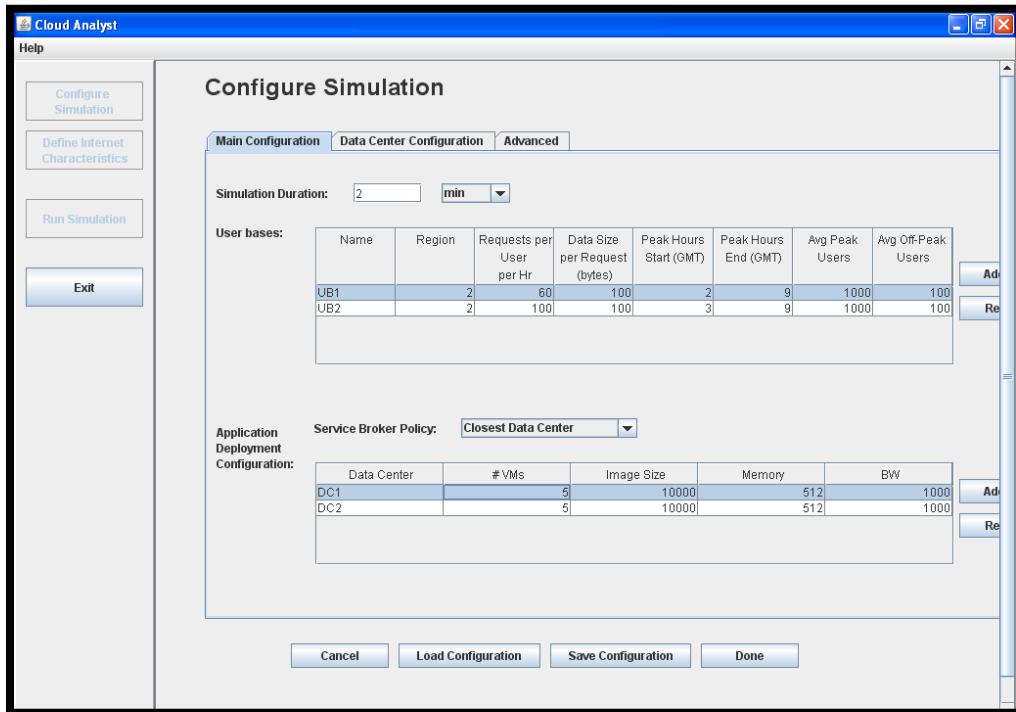
Cloud Computing - II



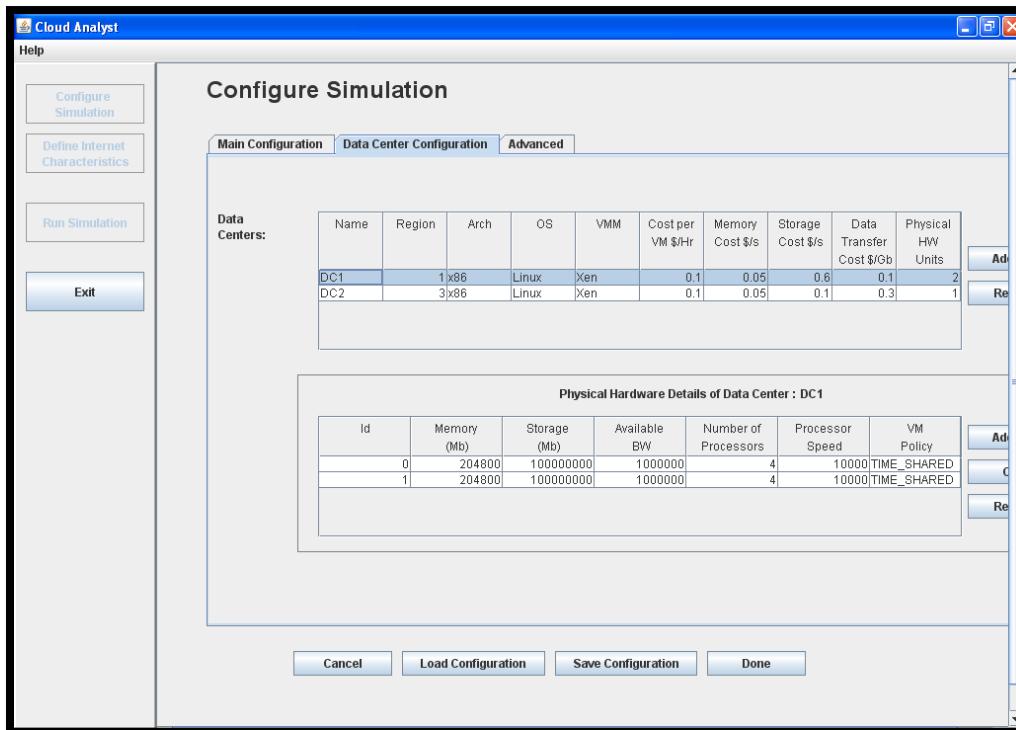
Now we are ready for checking algorithm performance

In Configure simulation window -> Main configuration tab set the values to

Cloud Computing - II

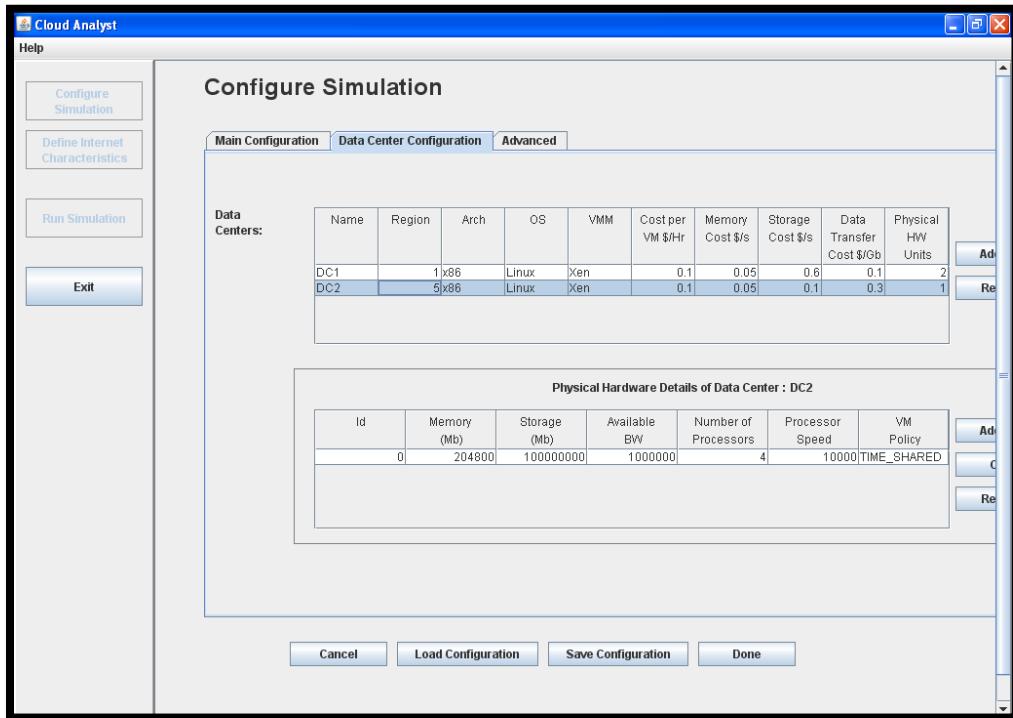


In data center configuration tab

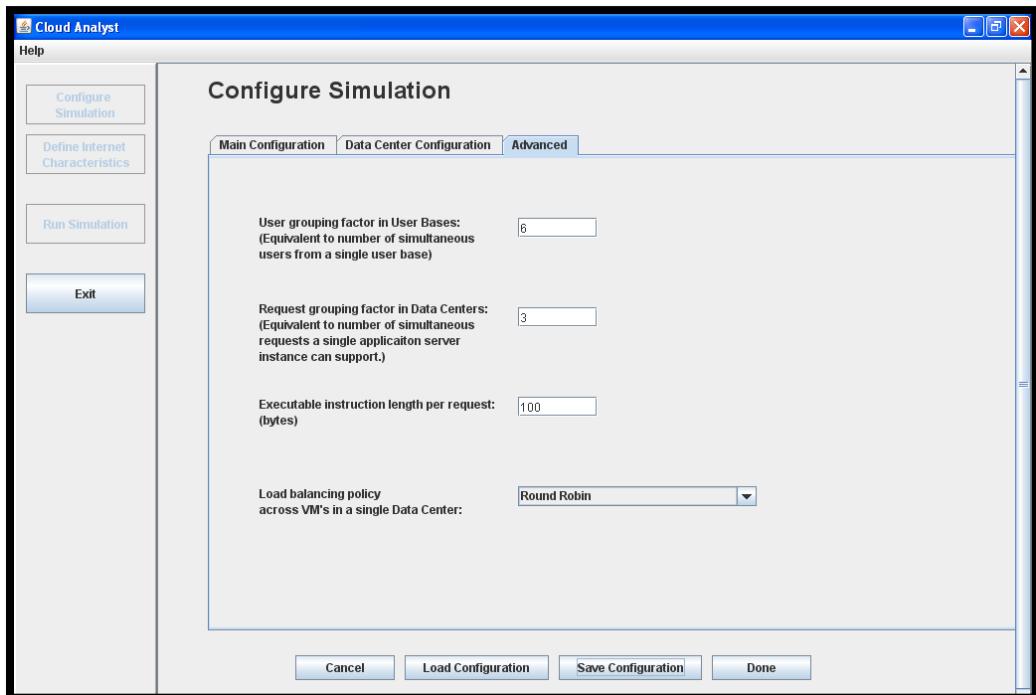


and

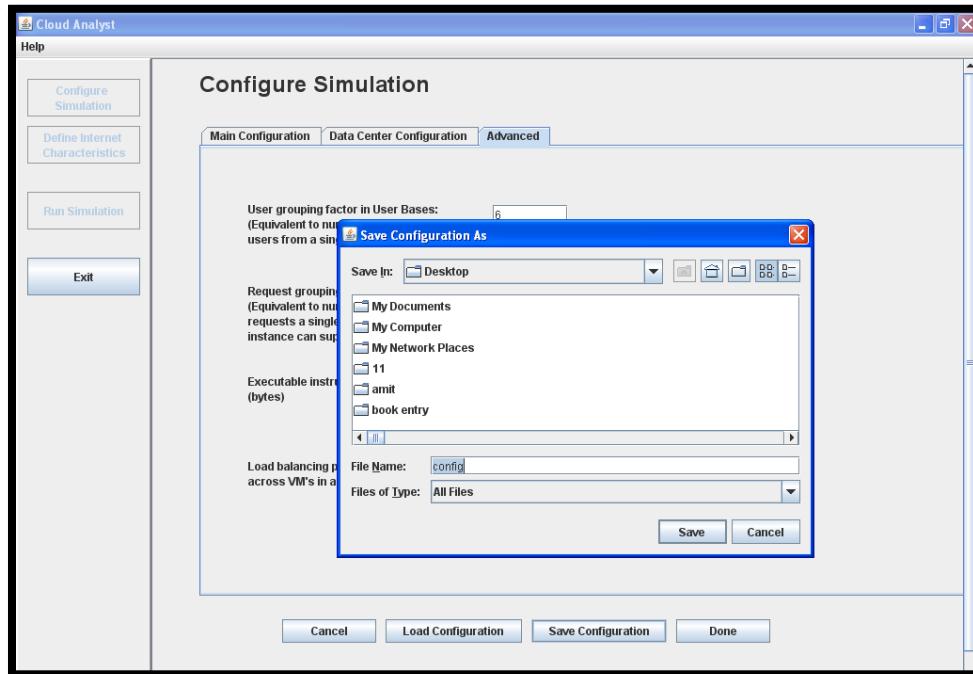
Cloud Computing - II



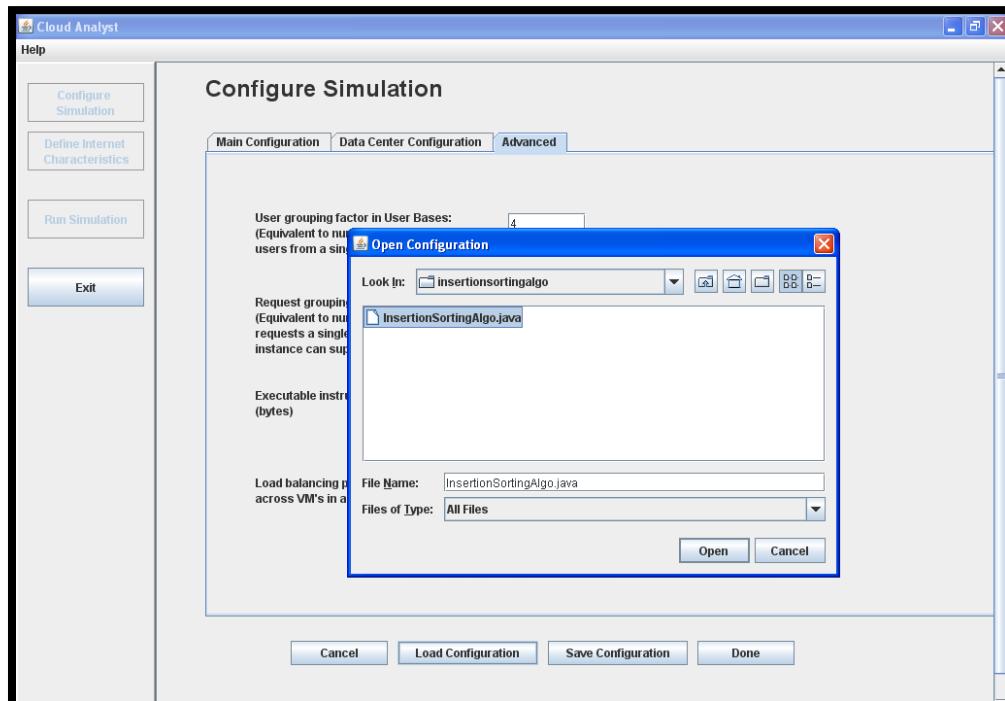
In advanced tab



Cloud Computing - II

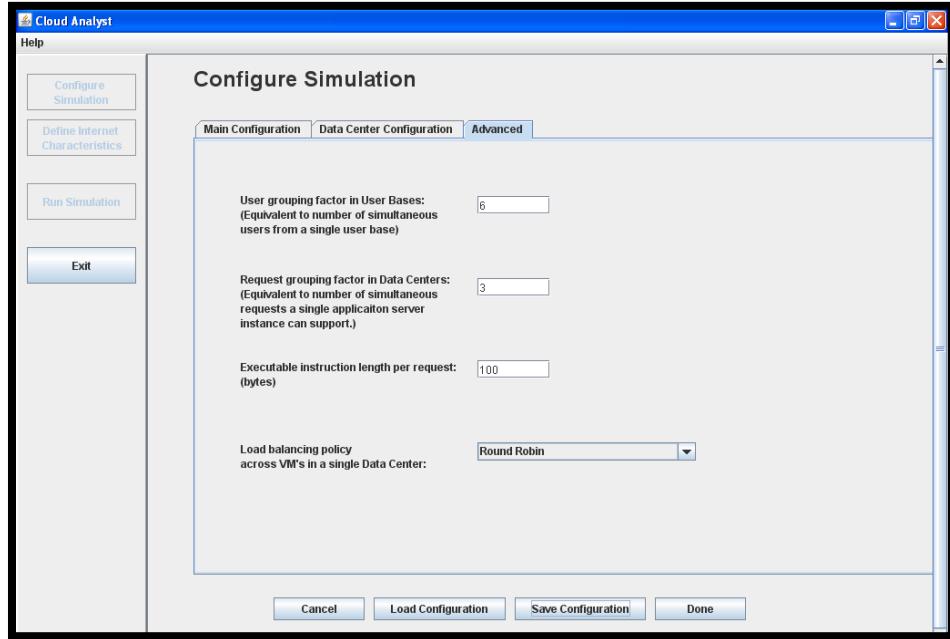


then click on load configuration button and select algorithm as



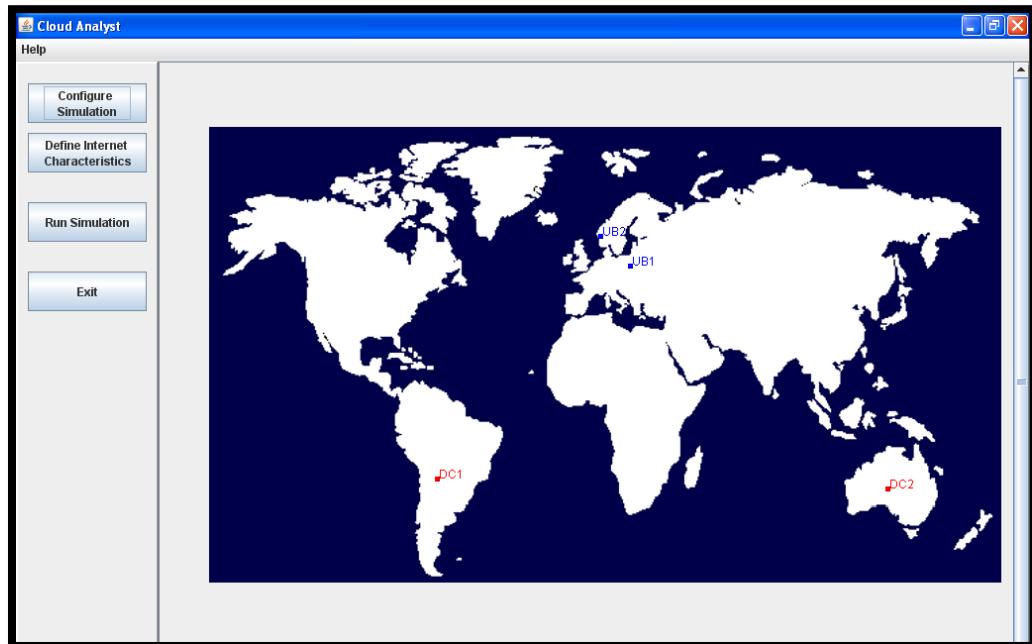
Cloud Computing - II

then save configuration as



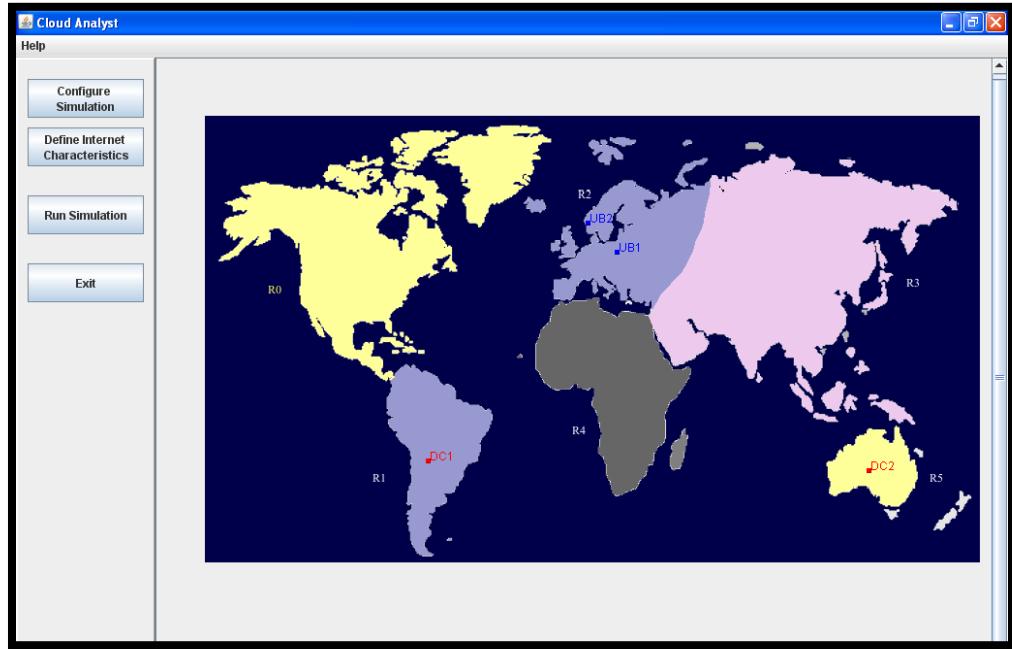
.sim file will saved to given location

then click on done

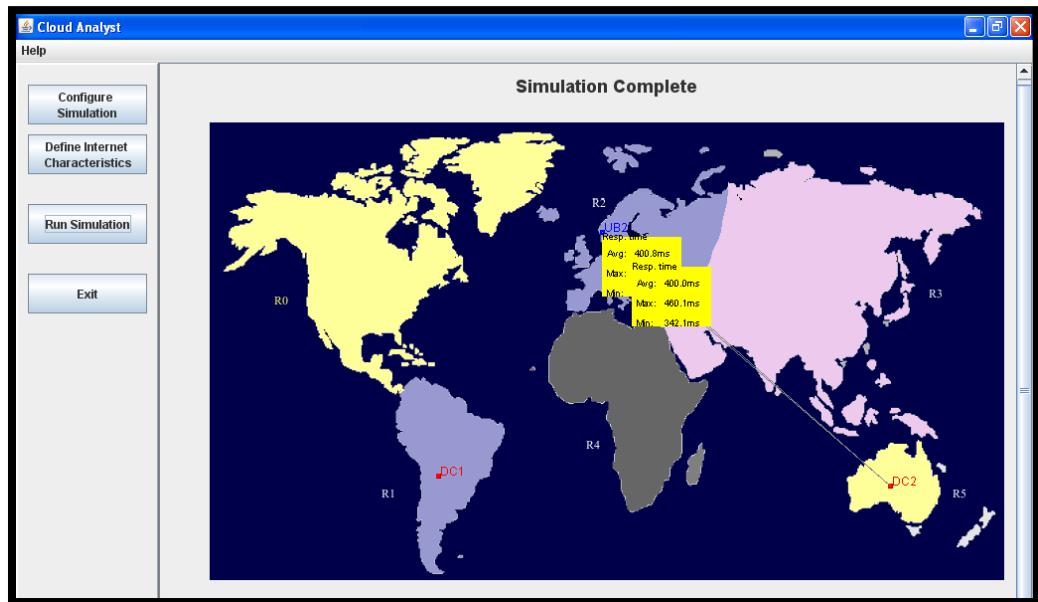


Cloud Computing - II

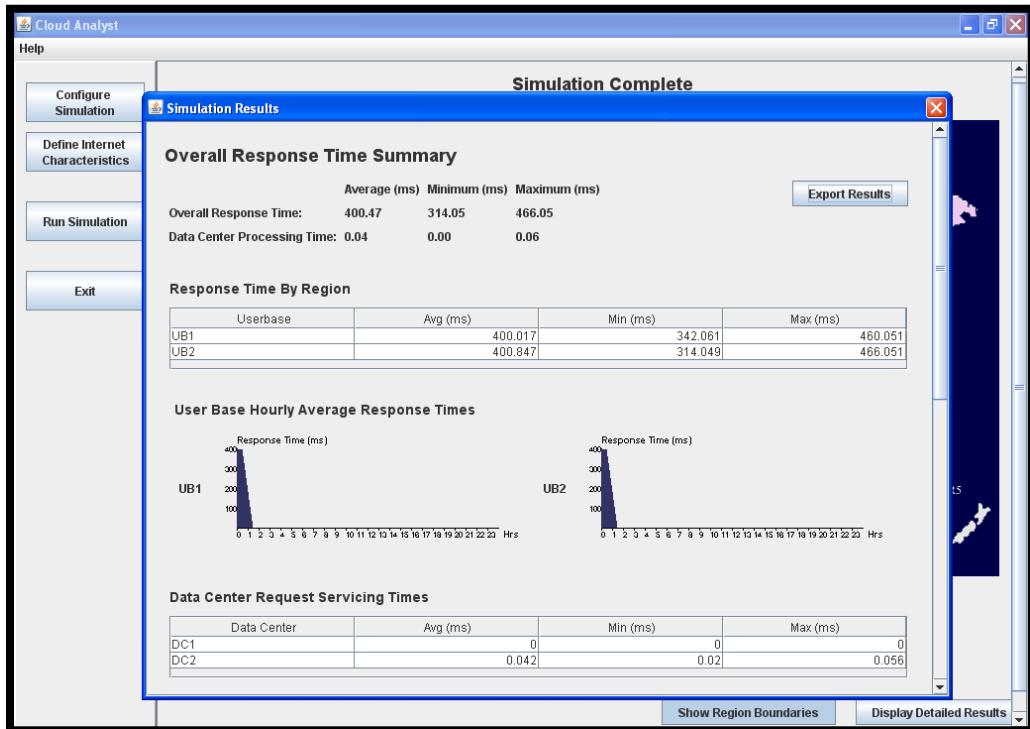
click on show region boundaries



then click run simulation



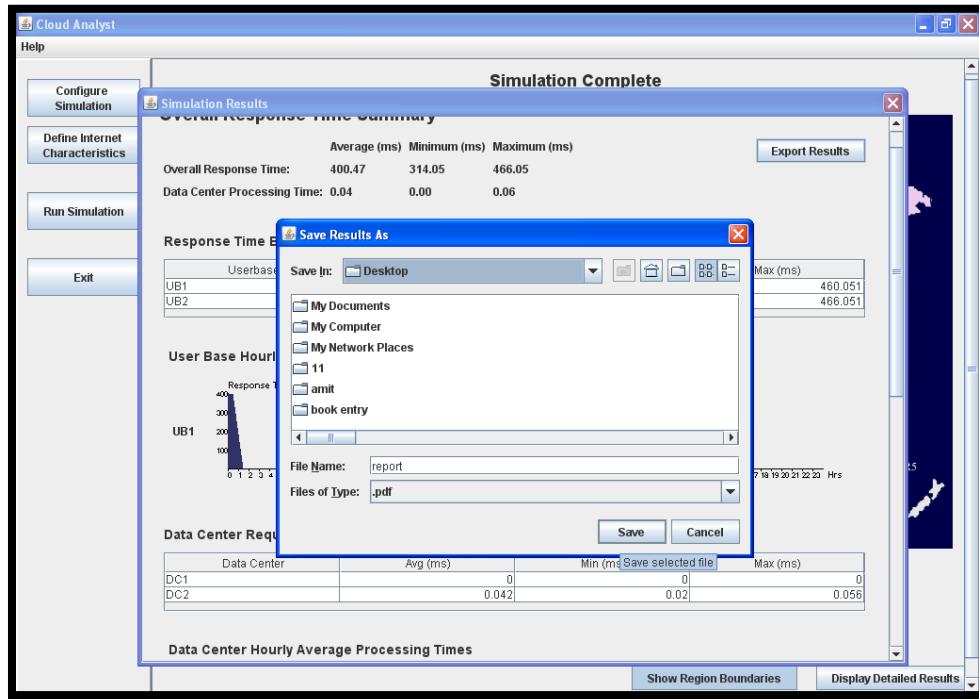
Cloud Computing - II



then click on export result

pdf will be saved to given location

Cloud Computing - II



Output:

simulation time =120000.0ms

Starting Simulation...

Initialising...

Creating new broker DC1-Broker

Creating new broker DC2-Broker

0.0 Creating new user base UB1

0.0 Creating new user base UB2

Starting GridSim version 4.2

Entities started.

Starting user base 9 UB1

Starting broker 8 name=DC2-Broker

Starting user base 11 UB2

Starting broker 6 name=DC1-Broker

Starting internet 13

5.0: DC1-Broker: Cloud Resource List received with 2 resource(s)

5.0: DC2-Broker: Cloud Resource List received with 2 resource(s)

5.0: DC2-Broker: Trying to Create VM #0

Cloud Computing - II

5.0: DC2-Broker: Trying to Create VM #1

5.0: DC2-Broker: Trying to Create VM #2

5.0: DC2-Broker: Trying to Create VM #3

5.0: DC2-Broker: Trying to Create VM #4

5.0: DC1-Broker: Trying to Create VM #0

5.0: DC1-Broker: Trying to Create VM #1

5.0: DC1-Broker: Trying to Create VM #2

5.0: DC1-Broker: Trying to Create VM #3

5.0: DC1-Broker: Trying to Create VM #4

Gathering simulation data.

DC1-Broker finalizing, submitted cloudlets=0 processing cloudlets=0 ,allRequestsProcessed=0

Got response for 900047 but it seems to be completed.

DC2-Broker finalizing, submitted cloudlets=226 processing cloudlets=0 ,allRequestsProcessed=670

UB1 finalizing. Messages sent:52, Received:52

UB1 requests sent=302 , received=302

UB2 finalizing. Messages sent:63, Received:63

UB2 requests sent=368 , received=368

Simulation completed.

***** Vm allocations in DC1

0->2

1->1

2->1

3->1

4->1

***** Vm allocations in DC2

0->92

1->92

2->92

3->91

4->91

****Datacenter: DC1****

User id	Debt
6	30128

*****Datacenter: DC2****

Cloud Computing - II

User id Debt

8 5128

Simulation finished at 145540.0

BUILD SUCCESSFUL (total time: 37 minutes 1 second)

Conclusion: Successfully executed and checked the performance of algorithm using CloudSim.

Practical No : 3

Aim : Build an application on private cloud.

Note :

1. Internet Connectivity Required
2. Preferably use Mozilla Firefox as browser
3. Install putty and puttygen and then start with practical
4. AWS Demo Account is required

Login to the aws manager account at link <https://aws.amazon.com/>

Select EC2

The screenshot shows the AWS Management Console homepage. The top navigation bar includes 'AWS Services' and 'Edit'. The user information 'rahulyagi7 @ 9440-1333-9674' and location 'Mumbai' are also present. On the left, there's a sidebar with various service icons. The 'Compute' section is expanded, showing 'EC2' as the selected service, which is highlighted in orange. Other services listed under Compute include EC2 Container Service, Elastic Beanstalk, Lambda, S3, CloudFront, Elastic File System, Glacier, Snowball, Storage Gateway, RDS, and DynamoDB. The main content area displays a grid of service icons and names. The 'EC2' service is located in the first row of the grid. To the right of the grid, there are sections for 'Resource Groups', 'Additional Resources', and links like 'Getting Started', 'AWS Console Mobile App', 'AWS Marketplace', and 'AWS re:Invent Announcements'. A 'Create a Group' button is also visible.

Click on EC2 and open it ,then select Launch INSTANCE

Cloud Computing - II

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, and Network & Security. The main area displays 'Resources' and 'Account Attributes'. Under 'Resources', it shows 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 1 Key Pairs, and 0 Placement Groups. Under 'Account Attributes', it lists Supported Platforms (VPC), Default VPC (vpc-7ada3913), and Resource ID length management. Below this is an 'Additional Information' section with links to Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us. A 'Create Instance' section with a 'Launch Instance' button is also present.

Select Linux AMI

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page. At the top, there are tabs for 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, and 7. Review. The '1. Choose AMI' tab is selected. The page title is 'Step 1: Choose an Amazon Machine Image (AMI)'. It says, 'An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.' A 'Quick Start' sidebar on the left has sections for My AMIs, AWS Marketplace, Community AMIs, and a 'Free tier only' checkbox. The main content area lists three AMIs:

- Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type - ami-fbbdd790**
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm
Select button (64-bit)
- Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-cdbdd7a2**
Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm
Select button (64-bit)
- SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type - ami-cebed4a1**
SUSE Linux Enterprise Server 12 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Select button (64-bit)

Cloud Computing - II

Chose the instance type

The screenshot shows the 'Step 2: Choose an Instance Type' page of the AWS EC2 instance creation wizard. The user is currently viewing the 't2.micro' instance type, which is highlighted in the list. The table displays various instance types categorized by family (General purpose), type (t2.nano through t2.large), vCPUs, memory, instance storage, EBS-optimized availability, and network performance. The 't2.micro' row includes a note indicating it is 'Free tier eligible'. Navigation buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

Configure the Instance

The screenshot shows the 'Step 3: Configure Instance Details' page of the AWS EC2 instance creation wizard. The user has specified 1 instance. The configuration options include:

- Purchasing option:** Request Spot instances (unchecked)
- Network:** vpc-7ada3913 (172.31.0.0/16) | DEFAULT-VPC (selected) | Create new VPC
- Subnet:** No preference (default subnet in any Availability Zone)
- Auto-assign Public IP:** Use subnet setting (Enable)
- IAM role:** None | Create new IAM role
- Shutdown behavior:** Stop
- Enable termination protection:** Protect against accidental termination (unchecked)
- Monitoring:** Enable CloudWatch detailed monitoring (unchecked)

Navigation buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage'.

Cloud Computing - II

Add the Storage type according to the Requirement

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0a1185f01a129db23	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Tag Instance

Tag Instance .key and the value

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
RJCollege	DEFAULT-VPC

Create Tag (Up to 10 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Cloud Computing - II

Configure the security group by adding some rule..

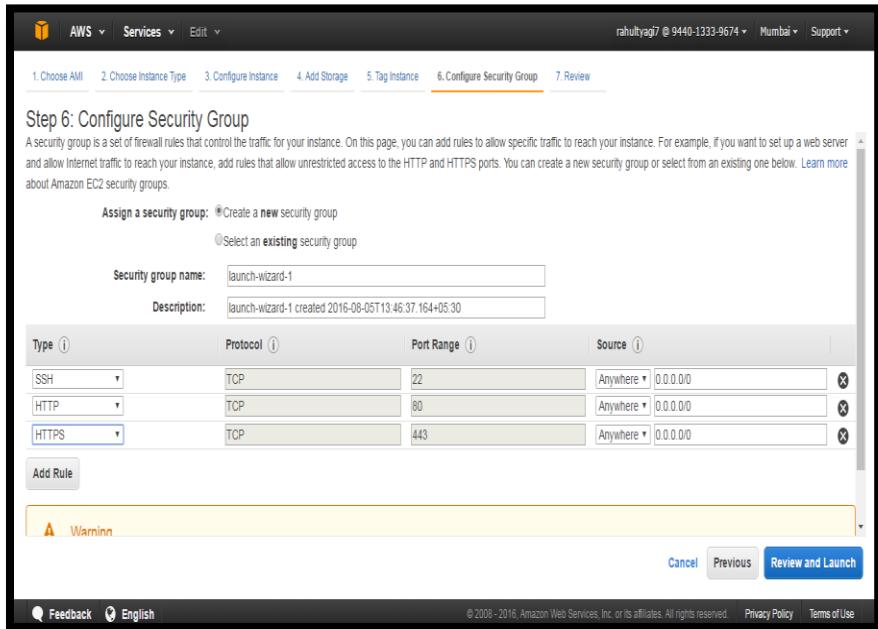
The screenshot shows the AWS EC2 Launch Wizard at Step 6: Configure Security Group. The security group name is set to 'launch-wizard-1'. A single rule is defined: Type: SSH, Protocol: TCP, Port Range: 22, Source: Anywhere (0.0.0.0/0). A warning message at the bottom states: 'Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The 'Review and Launch' button is highlighted.

Add the first rule Http.

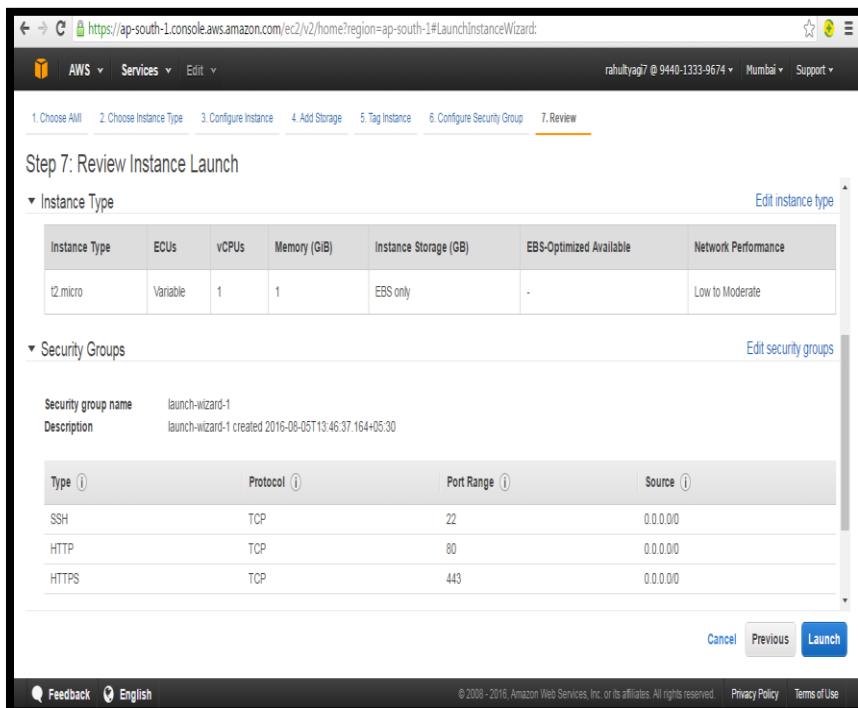
The screenshot shows the AWS EC2 Launch Wizard at Step 6: Configure Security Group. The 'HTTP' option is selected in the sidebar. The security group 'launch-wizard-1' now contains two rules: one for SSH (TCP port 22, Anywhere) and one for HTTP (TCP port 80, Custom CIDR, IP or Security Group). The warning message at the bottom is identical to the previous screenshot.

Cloud Computing - II

Add the second rule Https



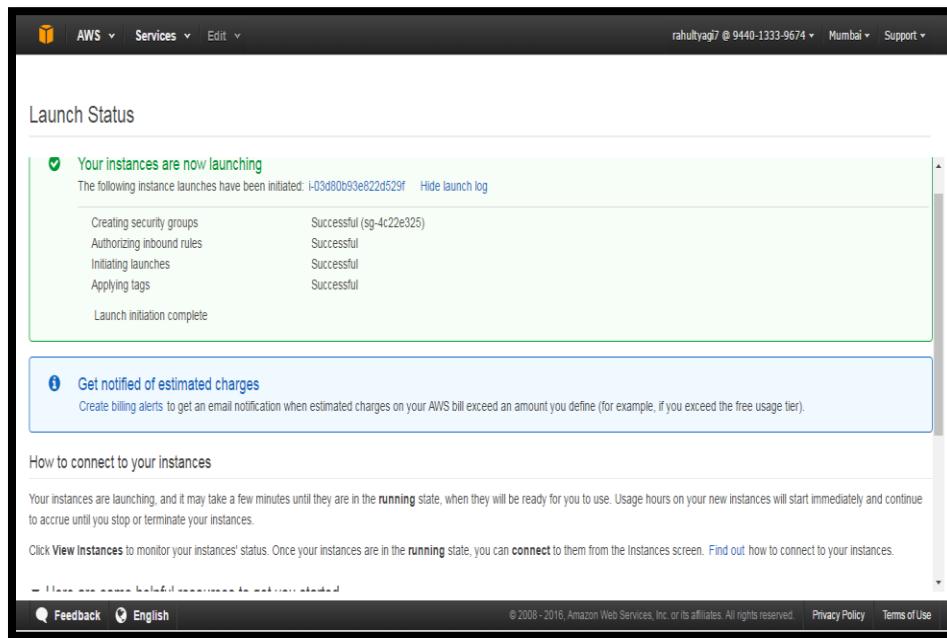
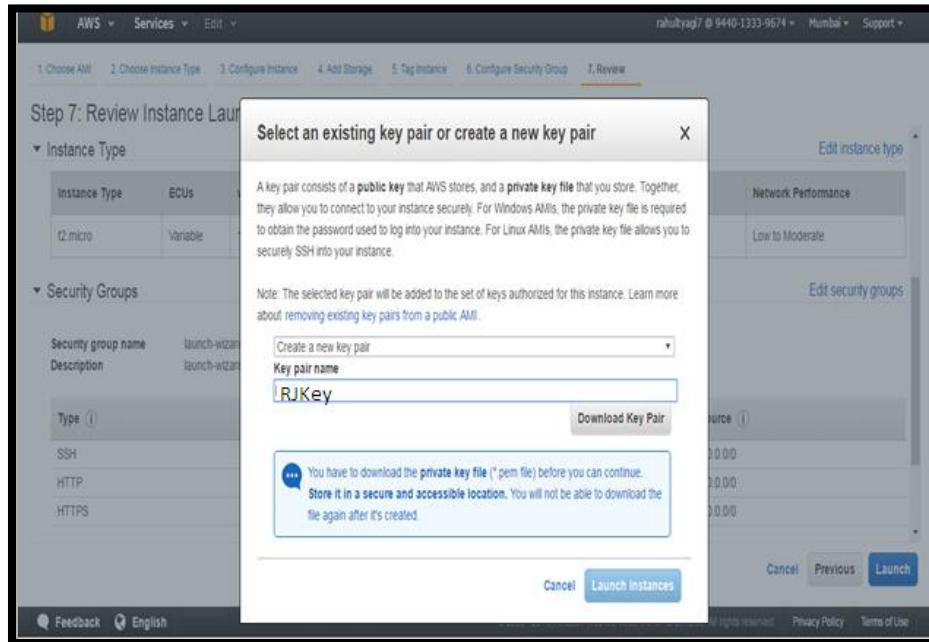
Review the Instance before launch



Once by Launching instance.the dialog box is pop up.by the message select key creating option.

Cloud Computing - II

And Download it.



Cloud Computing - II

Once you launch the instance it will take some minutes to Launch it...the Public DNS address will be launch

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under 'INSTANCES', 'Instances' is selected. In the main content area, there is a table with one row. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and Publ. The instance listed is 'i-03d80b93e822d529f' with 't2.micro' type, 'ap-south-1a' availability zone, 'running' state, and 'Initializing' status. The Public DNS is 'ec2-52-66-122-231.ap-south-1.compute.amazonaws.com'. Below the table, a detailed view for the same instance is shown with tabs for Description, Status Checks, Monitoring, and Tags. The 'Description' tab is active, displaying various details such as Instance ID, Instance state, Instance type, Private DNS, Private IPs, Secondary private IPs, VPC ID, Public DNS, Public IP, Elastic IPs, Availability zone, Security groups, Scheduled events, and AMI ID.

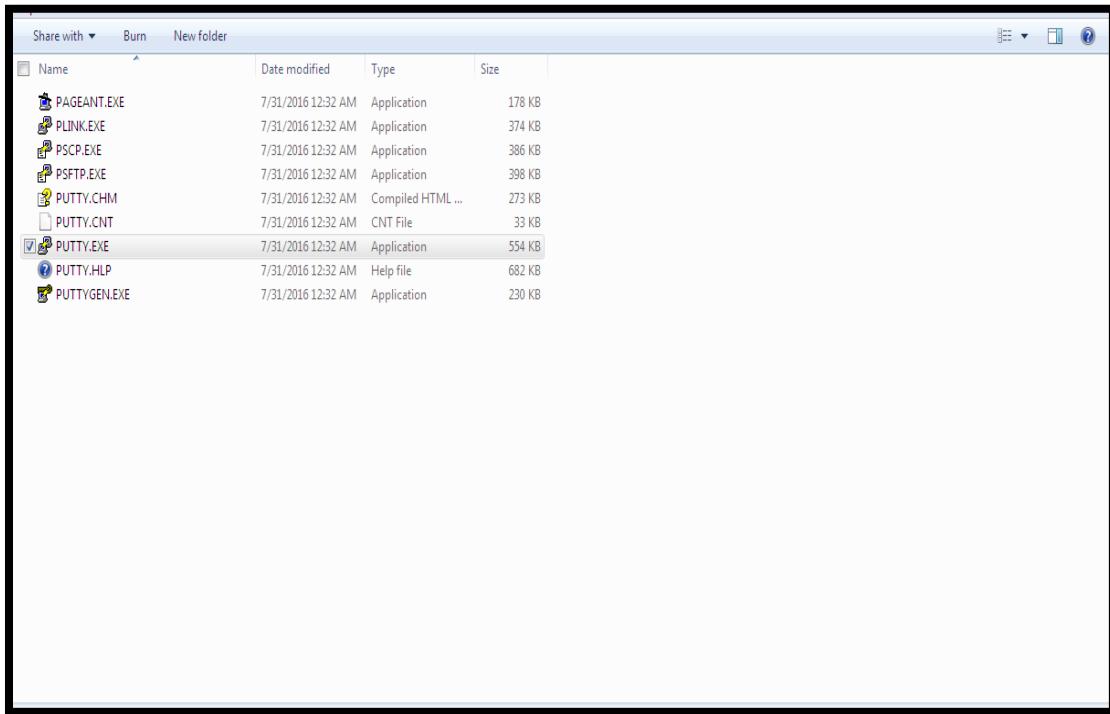
Here we have completed with instance creation.....

Now we have to connect this created instance with LAMP Server.....

Download the puttygen

and putty for the key configuration..

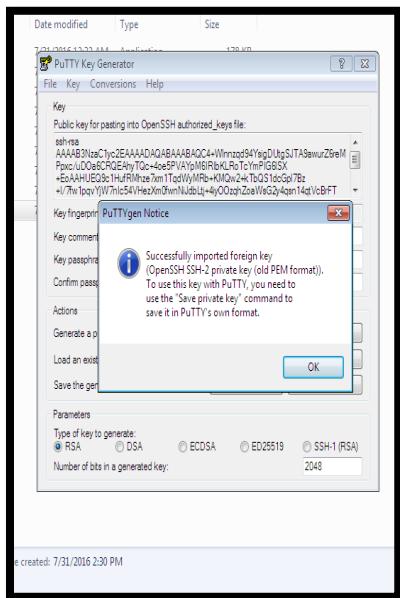
Cloud Computing - II



Open the puttygen-> load-><go to address of public key> -> open the public key you had save..

After that save the Private key which is used for verification for secure connection.

Cloud Computing - II



Save the private key.

A screenshot of the AWS EC2 Dashboard. On the left sidebar, 'Instances' is selected. In the main pane, an instance named 'i-03d80b93e822d529' is listed. A modal window titled 'PutTY Key Generator' is overlaid on the dashboard, showing the same imported key details as the previous screenshot. The AWS interface includes tabs for 'Description' and 'Status Checks', and a detailed view of the instance's public DNS, IP, and AMI information on the right.

Cloud Computing - II

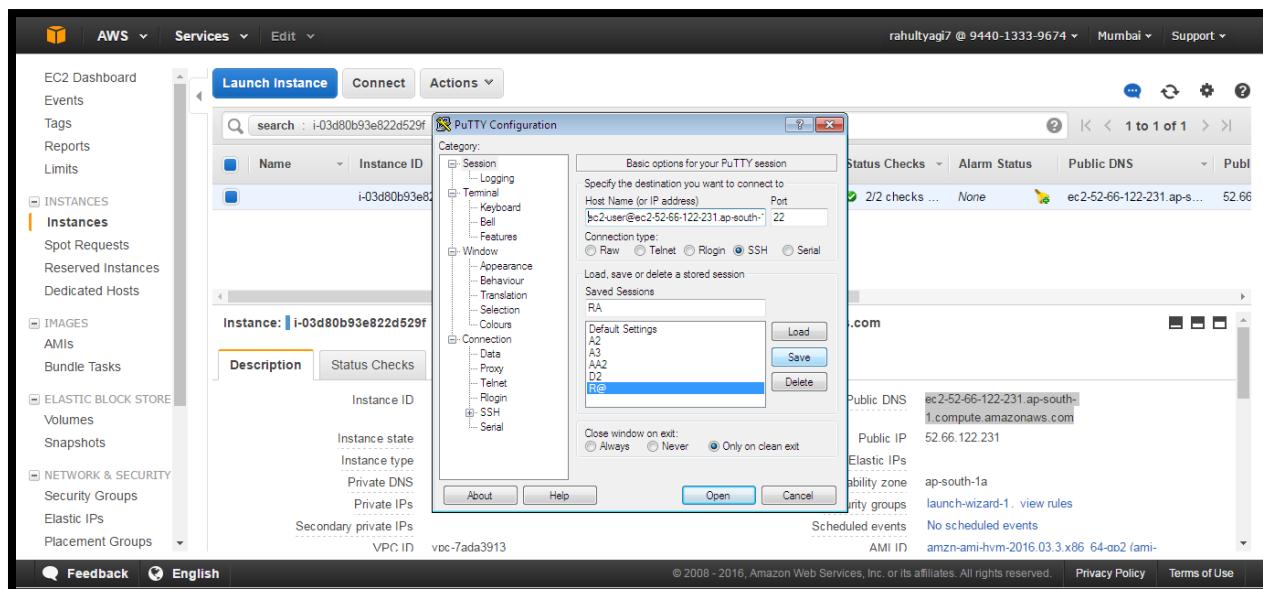
Now open the putty for the key configuration.

In Host Name(Or IP address) write

Ec2-user@public DNS.

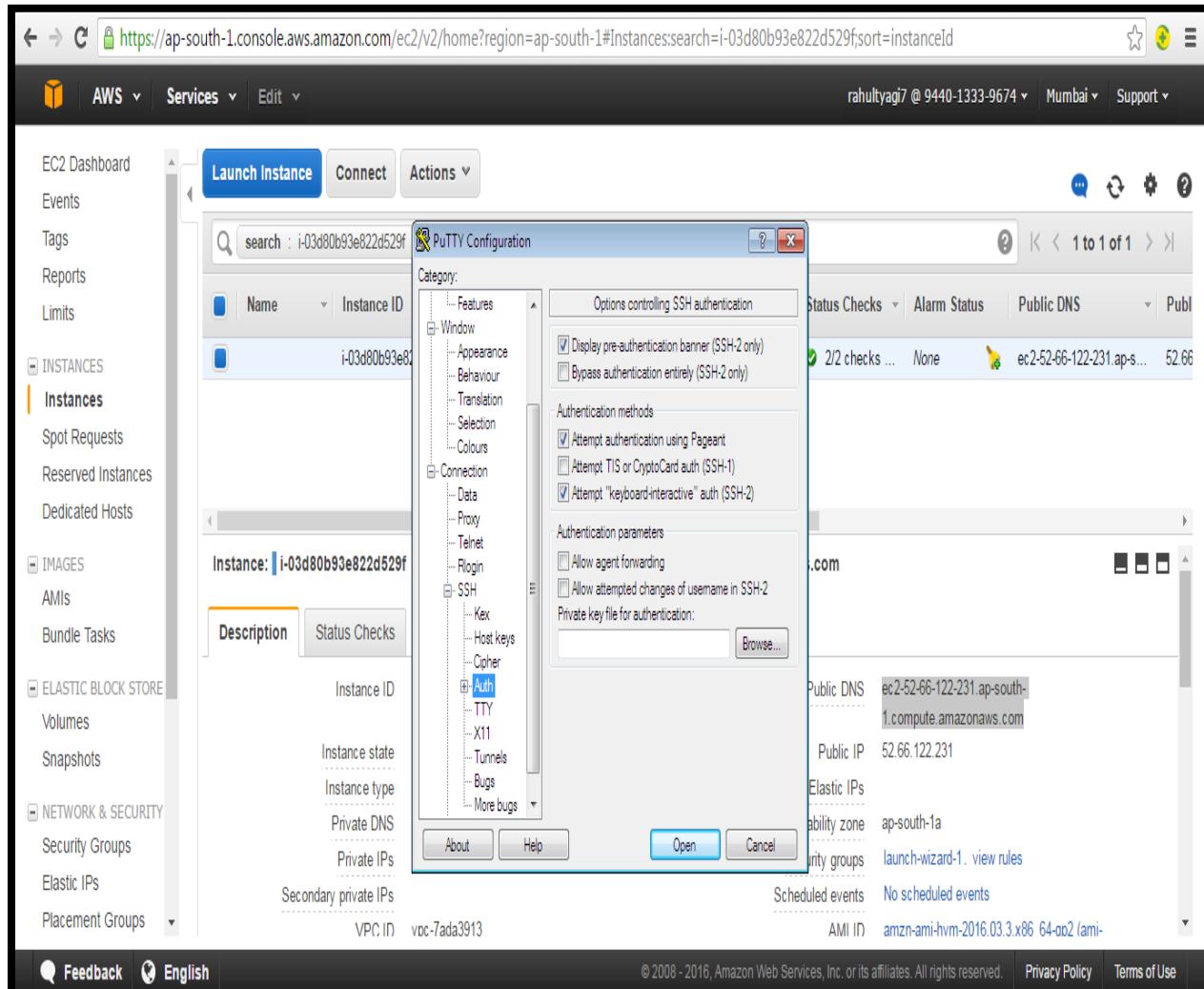
In saved Session option provide the name and save it.

Now at left side select SSH ->Auth ->Browse(Right side)->select private key from where had it saved.....



Now select the private key you have save.

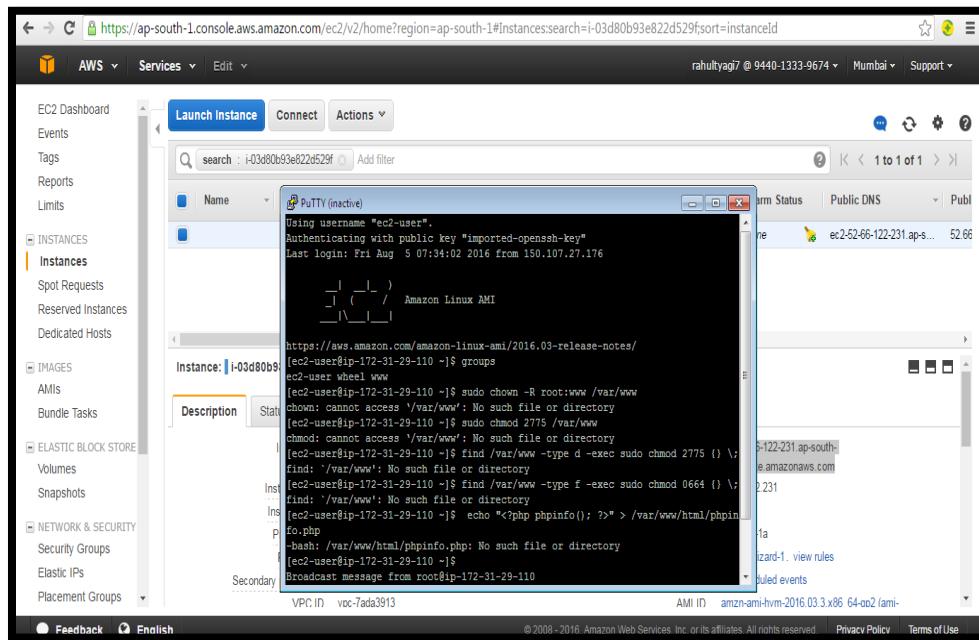
Cloud Computing - II



By clicking on the open, the Ec2 Amazon Linux will be launch...

For the configuration use the Command provided in putty.....

Cloud Computing - II



Command for launching Lamp server.

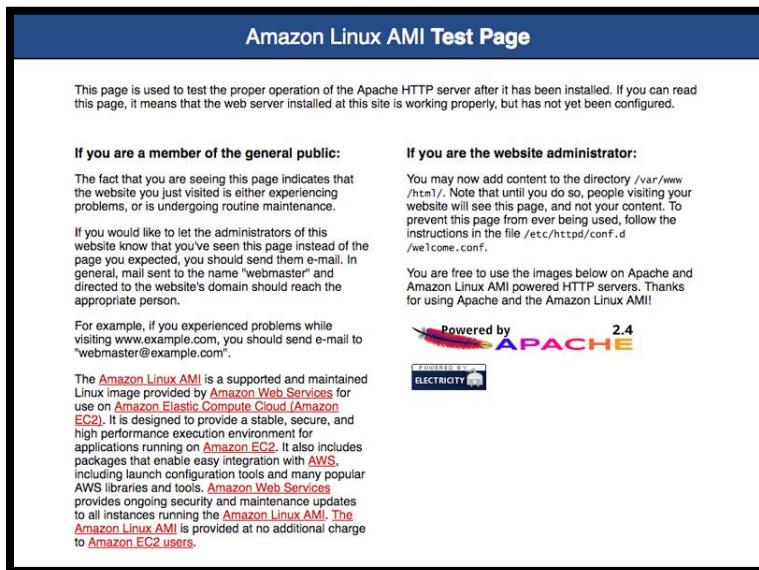
```
new1 - Notepad+
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
New [x]
1 Connect to your instance.
2 -----
3
4 software packages are up to date
5 -----
6
7 --->$ sudo yum update -y
8 -----
9
10 3 yum install
11 -----
12
13 ----> $ sudo yum install -y httpd24 php56 mysql55-server php56-mysqld
14 -----
15 4 Start the Apache web server.
16 -----
17 --->$ sudo service httpd start
18 -----
19
20 5 Use the chkconfig command to configure the Apache web
21
22 server to start at each system boot.
23 -----
24
25 --->$ sudo chkconfig httpd on
26 -----
27
28 6 verify that httpd is on
29 -----
30 --->$ chkconfig --list httpd
31 -----
```

Cloud Computing - II

Now minimize putty.....

Copy Public DNS in new tab of browser.....

If the following page appears means LAMP server is on.....



Minimize it and go to putty to Configure Apache server for PHP

```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
1 Apache httpd serves files that are kept in a directory called the
2 Apache document root. The Amazon Linux Apache
3 document root is /var/www/html, which is owned by root by
4 default.
5
6 --> $ ls -l /var/www
7
8 To set file permissions
9 =====
10 1 Add the www group to your instance.
11
12 -->$ sudo groupadd www
13
14 2 Add your user to the www group.
15
16 ---> sudo usermod -a -G www ec2-user
17
18
19 3 You need to log out and log back in to pick up the new
20 group. You can use the exit command, or close the terminal window.
21
22 1 Log out.
23
24
25
26
Normal text file length: 2102 lines: 66 Lin:16 Col:39 Sel:0|0 DosWindows UTF-8 INS
```

Cloud Computing - II

The screenshot shows a Notepad++ window with several lines of terminal commands. The commands are as follows:

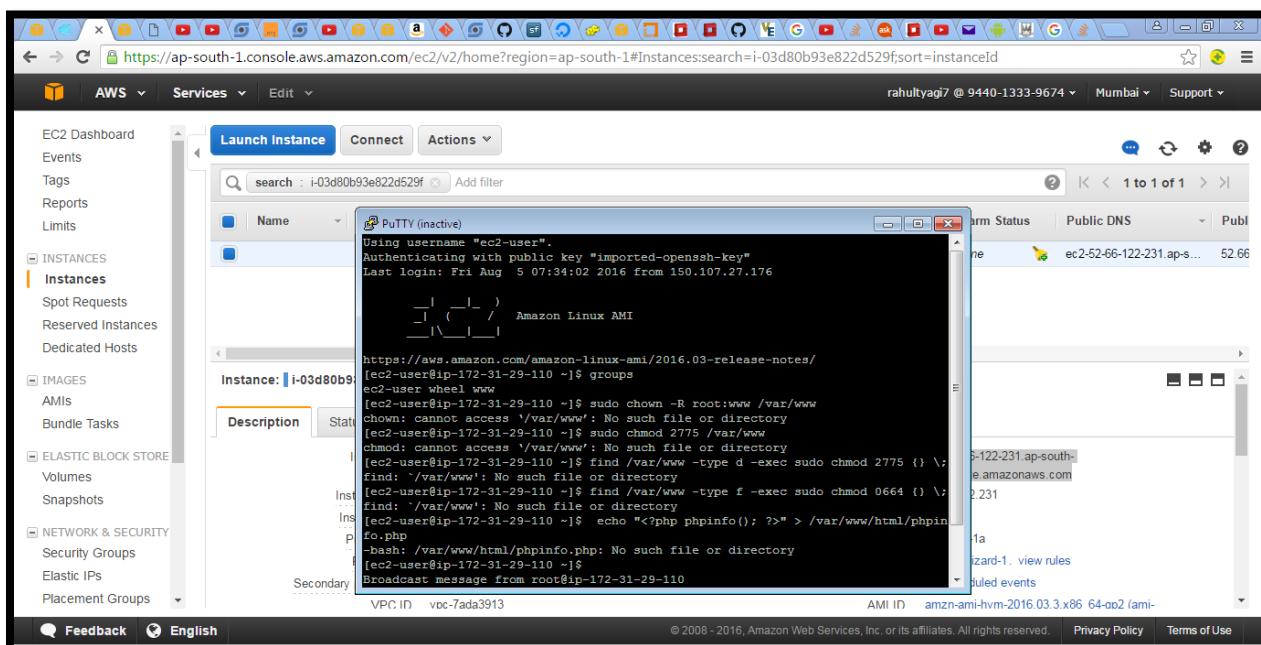
```
new - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
[File] [Edit] [Search] [View] [Encoding] [Language] [Settings] [Macro] [Run] [Plugins] [Window] ?
```

```
new1.txt
```

```
1 =====
2 Reconnect to your instance, and then run the following
3 command to verify your membership in the www group.
4 =====
5 =>$ groups
6
7 4 Change the group ownership of /var/www and its contents to the www group.
8
9 => sudo chown -R root:www /var/www
10
11 5 Change the directory permissions of /var/www and its subdirectories to add group
12 write permissions and to set the group ID on future subdirectories.
13
14 ==>$ sudo chmod 2775 /var/www
15
16 ==>$ find /var/www -type d -exec sudo chmod 2775 {} ;
17
18 =====
19 6 Recursively change the file permissions of /var/www and its subdirectories to add group write permissions.
20
21 1 find /var/www -type f -exec sudo chmod 0664 {} ;
22
23 =====
24 To test and execute php file on LAMP Server
25
26 ==> echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
27
28 =====
29 In a web browser, enter the URL of the file you just created. This URL is the public DNS address
30 of your instance followed by a forward slash and the file name. For example:
31
32 ==>http://my-public-dns.amazonaws.com/phpinfo.php
33
```

Now test php file.....

```
$ echo "<?phpphpinfo(); ?>" > /var/www/html/phpinfo.php
```



Now

Cloud Computing - II

In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

<http://my.public.dns.amazonaws.com/phpinfo.php>

PHP Version 5.6.6	
System	Linux ip-172-31-7-35.3.14.35-28.38.amzn1.x86_64 #1 SMP Wed Mar 11 22:50:37 UTC 2015 x86_64
Build Date	Mar 5 2015 23:26:53
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php-5.6.d
Additional .ini files parsed	/etc/php-5.6.d/20-bz2.ini, /etc/php-5.6.d/20-calendar.ini, /etc/php-5.6.d/20-ctype.ini, /etc/php-5.6.d/20-curl.ini, /etc/php-5.6.d/20-dom.ini, /etc/php-5.6.d/20-exif.ini, /etc/php-5.6.d/20-fileinfo.ini, /etc/php-5.6.d/20-ftp.ini, /etc/php-5.6.d/20-gettext.ini, /etc/php-5.6.d/20-iconv.ini, /etc/php-5.6.d/20-mysqlind.ini, /etc/php-5.6.d/20-pdo.ini, /etc/php-5.6.d/20-phar.ini, /etc/php-5.6.d/20-posix.ini, /etc/php-5.6.d/20-shmop.ini, /etc/php-5.6.d/20-simplexml.ini, /etc/php-5.6.d/20-sockets.ini, /etc/php-5.6.d/20-sqlite3.ini, /etc/php-5.6.d/20-sysmsg.ini, /etc/php-5.6.d/20-sysvsem.ini, /etc/php-5.6.d/20-sysvshm.ini, /etc/php-5.6.d/20-tokenizer.ini, /etc/php-5.6.d/20-xml.ini, /etc/php-5.6.d/20-xmwriter.ini, /etc/php-5.6.d/20-zip.ini, /etc/php-5.6.d/30-mysqli.ini, /etc/php-5.6.d/30-pdo_mysql.ini, /etc/php-5.6.d/30-pdo_sqlite.ini, /etc/php-5.6.d/30-wddx.ini, /etc/php-5.6.d/30-xmheader.ini, /etc/php-5.6.d/40-json.ini, /etc/php-5.6.d/40-datetime.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API20131226.NTS
PHP Extension Build	API20131226.NTS

Now execute php file as....

```
echo "<?php echo 'Hello from Zeeshan ' ; ?>" > /var/www/html/RJ.php
```

Hello from Zeeshan

Conclusion: Successfully built an application on private cloud.

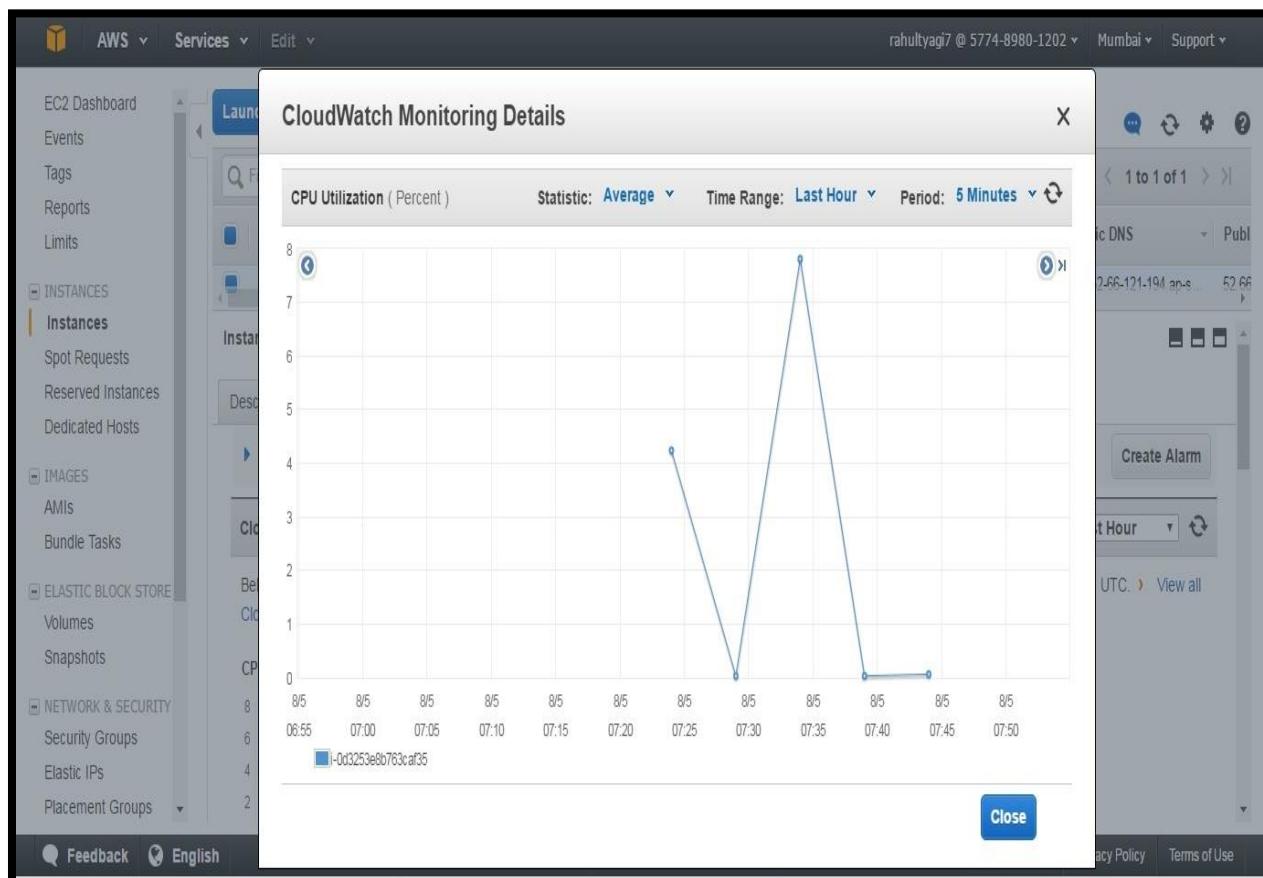
Practical No : 4

Aim : Demonstrate any Cloud Monitoring tool.

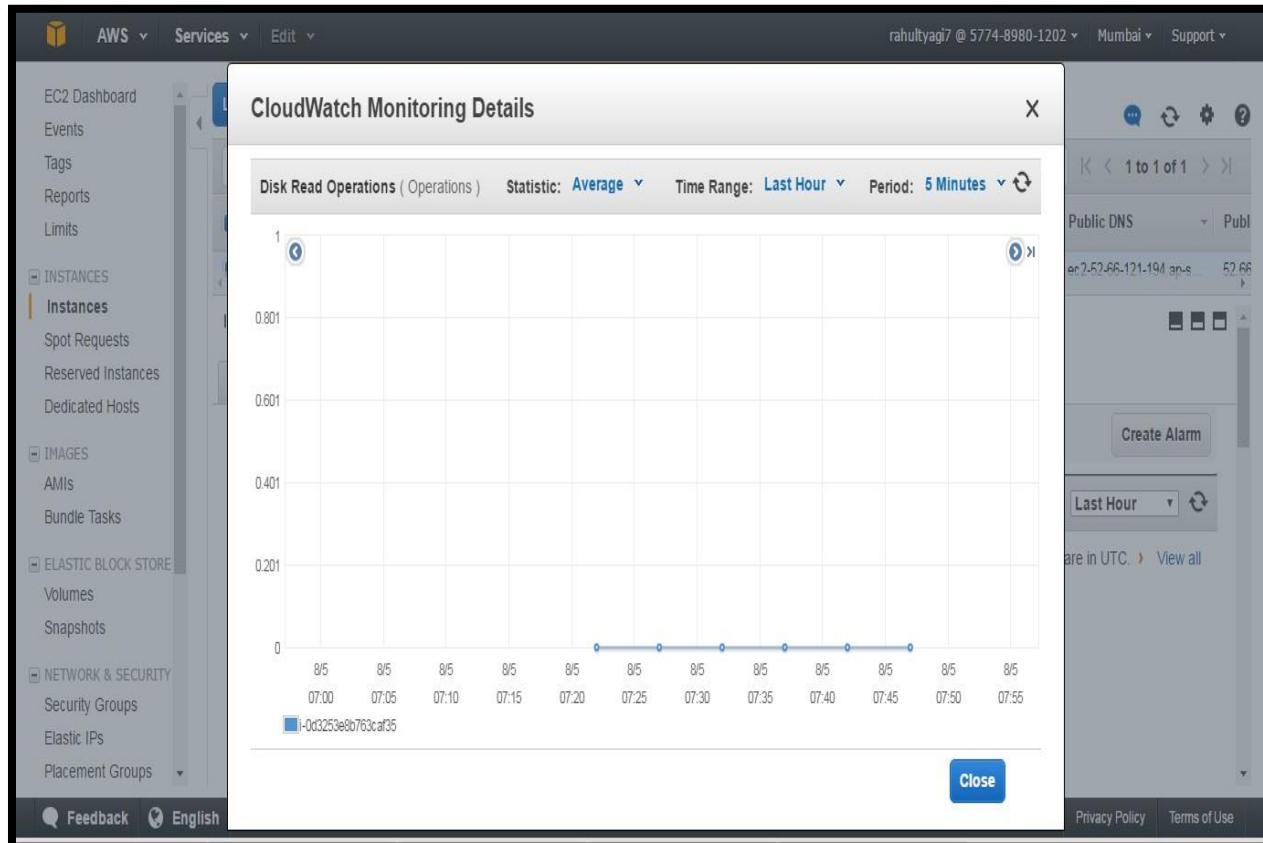
Theory:

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events. You get a unified view of operational health and gain complete visibility of your AWS resources, applications, and services running on AWS and on-premises. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

Now Monitor the graph of CPU..



Cloud Computing - II



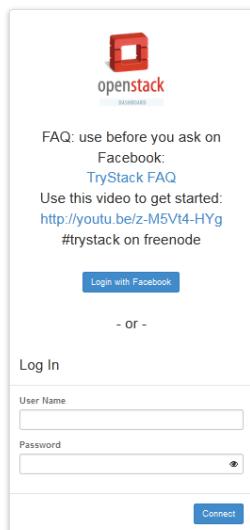
Practical No : 05

Aim : Evaluate a Private IAAS Cloud using TryStack

Note :

1. Install cygwin and then start with practical.

Go to link <https://x86.trystack.org/dashboard/auth/login/?next=/dashboard/>



Login with facebook account....

After login.....

Cloud Computing - II

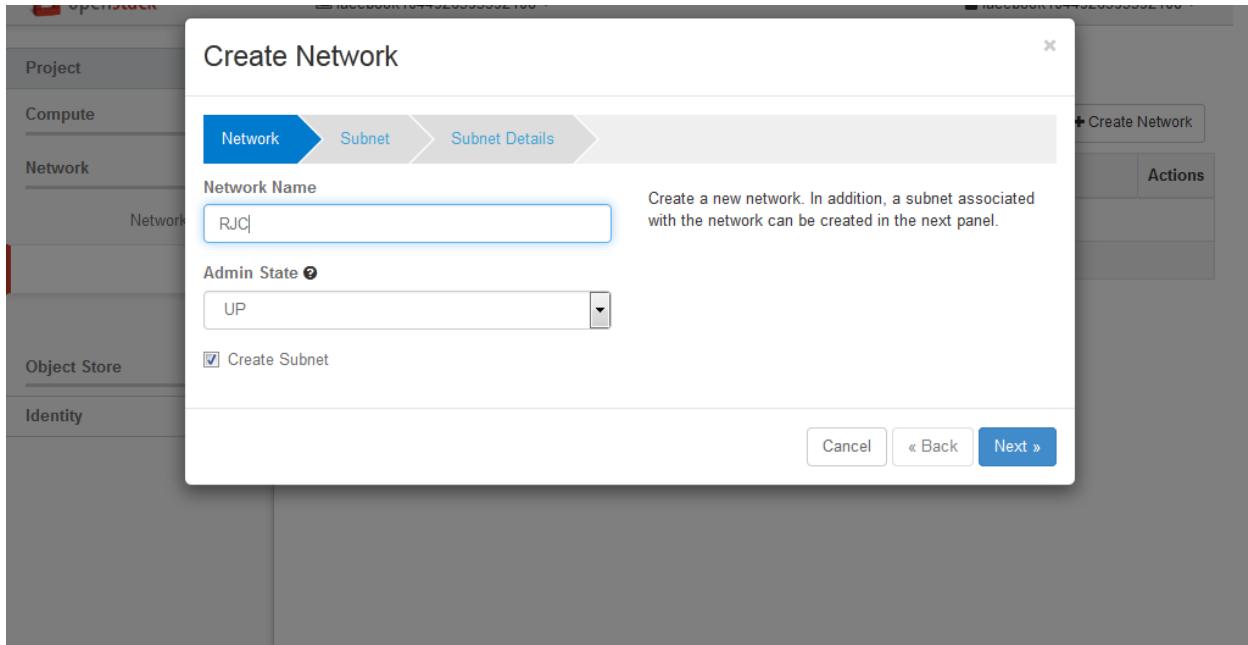
The screenshot shows the OpenStack Compute (Nova) dashboard under the 'Compute' project. The left sidebar has 'Compute' selected. The main area is titled 'Overview' and contains a 'Limit Summary' section with five pie charts showing usage ratios for Instances, VCPUs, RAM, Floating IPs, and Security Groups. Below this is a 'Usage Summary' section with a date range selector from '2018-07-20' to '2018-07-27', a 'Submit' button, and a note about date format. It also displays active instances, RAM, VCPUs, Disk, and RAM usage statistics. A 'Usage' table below shows no items displayed.

Go to Network(left side) ->

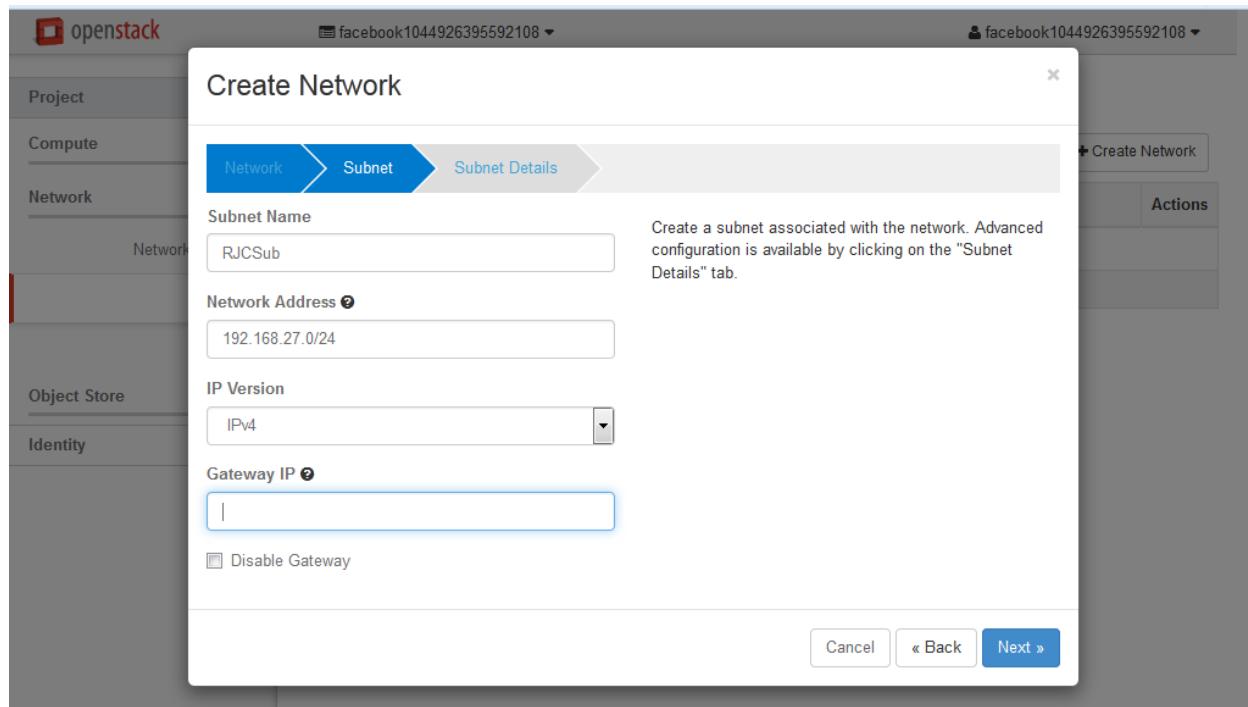
The screenshot shows the 'Networks' page under the 'Compute' project. The left sidebar has 'Network' selected. The main area is titled 'Networks' and contains a table for managing networks. The table has columns for Name, Subnets Associated, Shared, Status, Admin State, and Actions. A 'Create Network' button is visible at the top right of the table area. The message 'No items to display.' is shown below the table.

Click on Create Network(Right side tab)

Cloud Computing - II

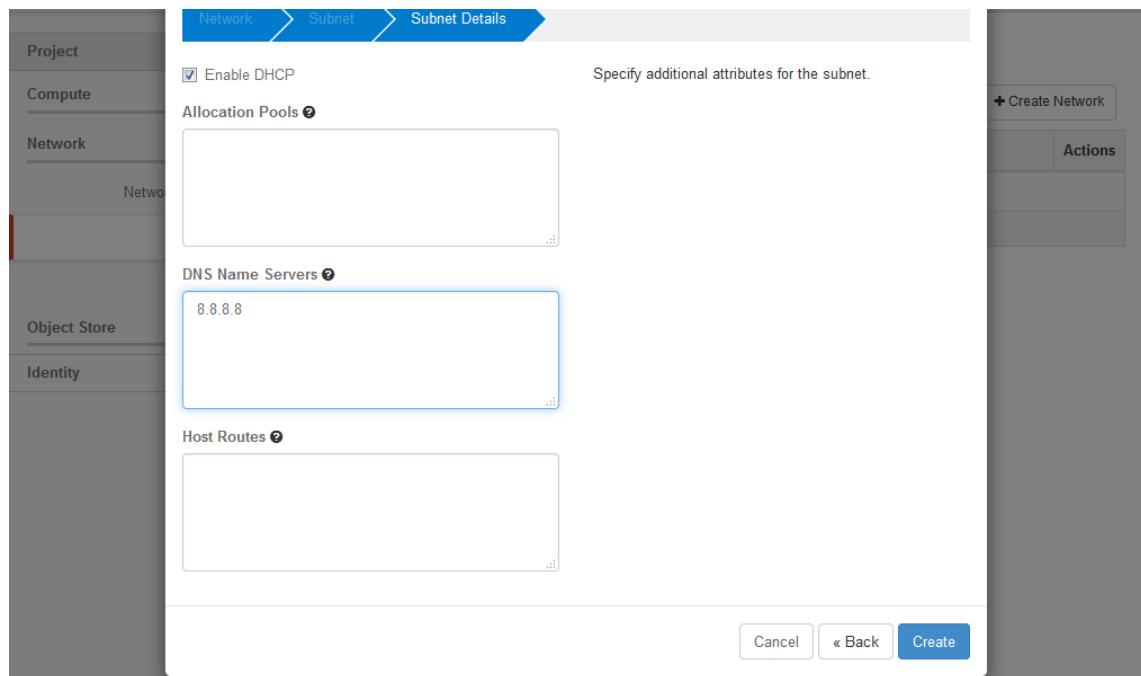


At Subnet tab



At subnet Details

Cloud Computing - II



8.8.8.8 google's DNS.....Click on create

So we have created network

openstack

facebook1044926395592108

Project Compute Network Object Store Identity

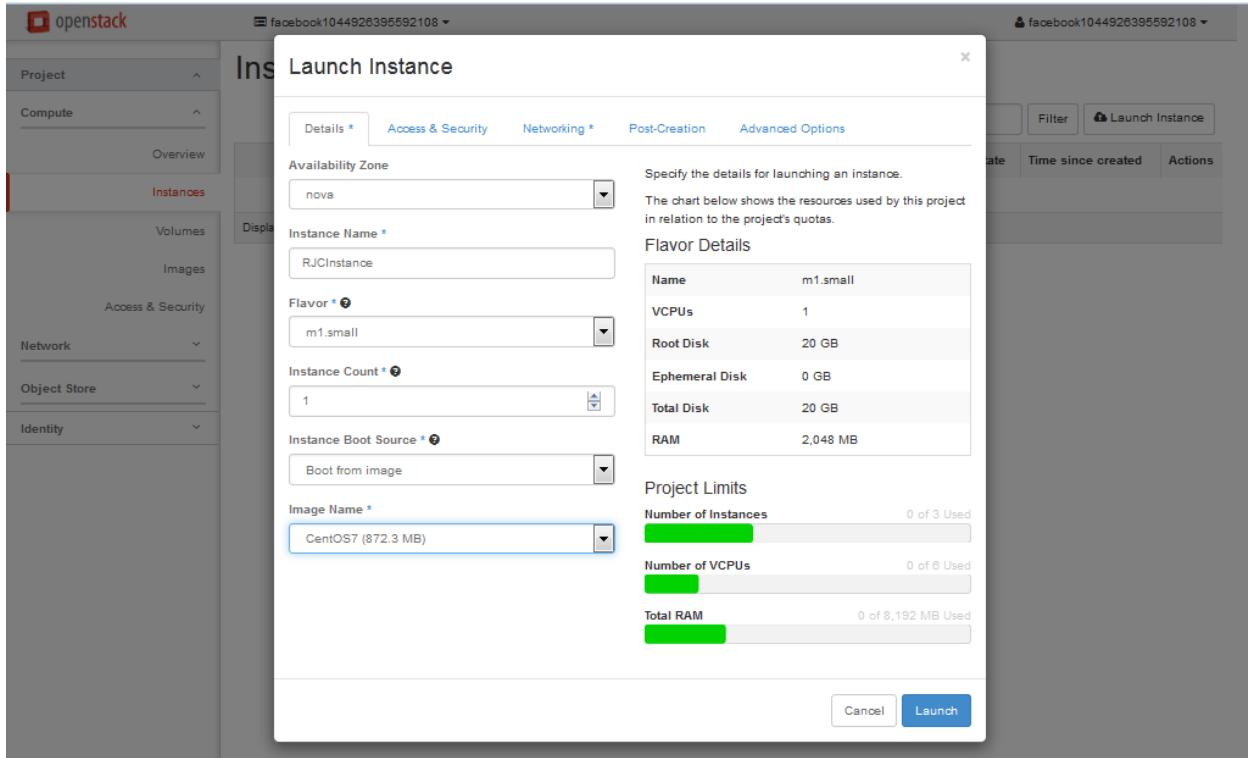
Network Topology Networks Routers

Displaying 1 item

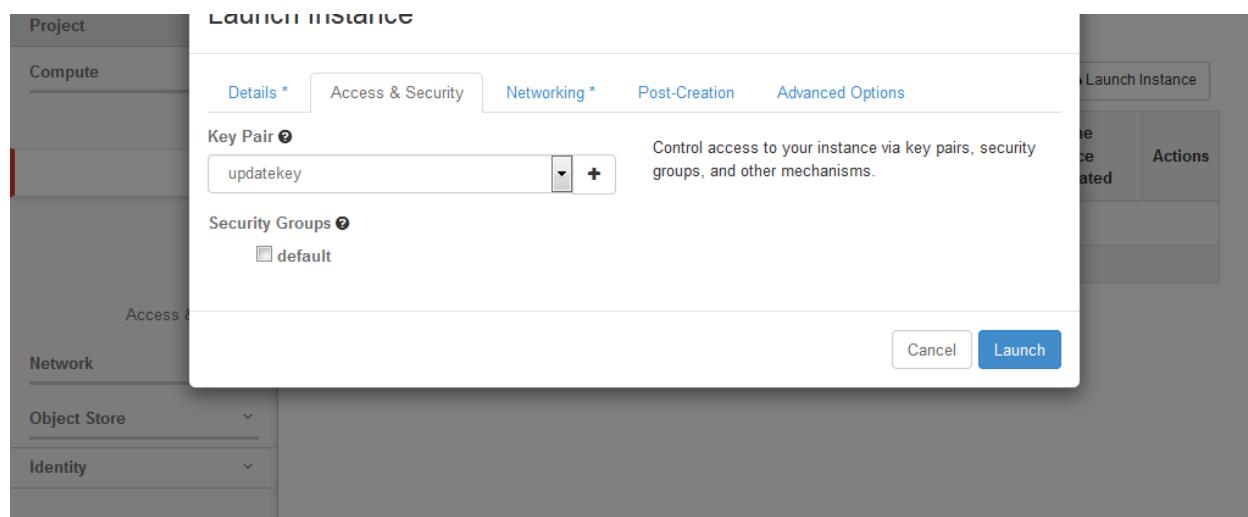
	Name	Subnets Associated	Shared	Status	Admin State	Actions
<input type="checkbox"/>	RJC	RJC Sub 192.168.27.0/24	No	Active	UP	<button>Edit Network</button>

Go to compute(left side) -> Instance-> Launch Instance(Right Side)

Cloud Computing - II

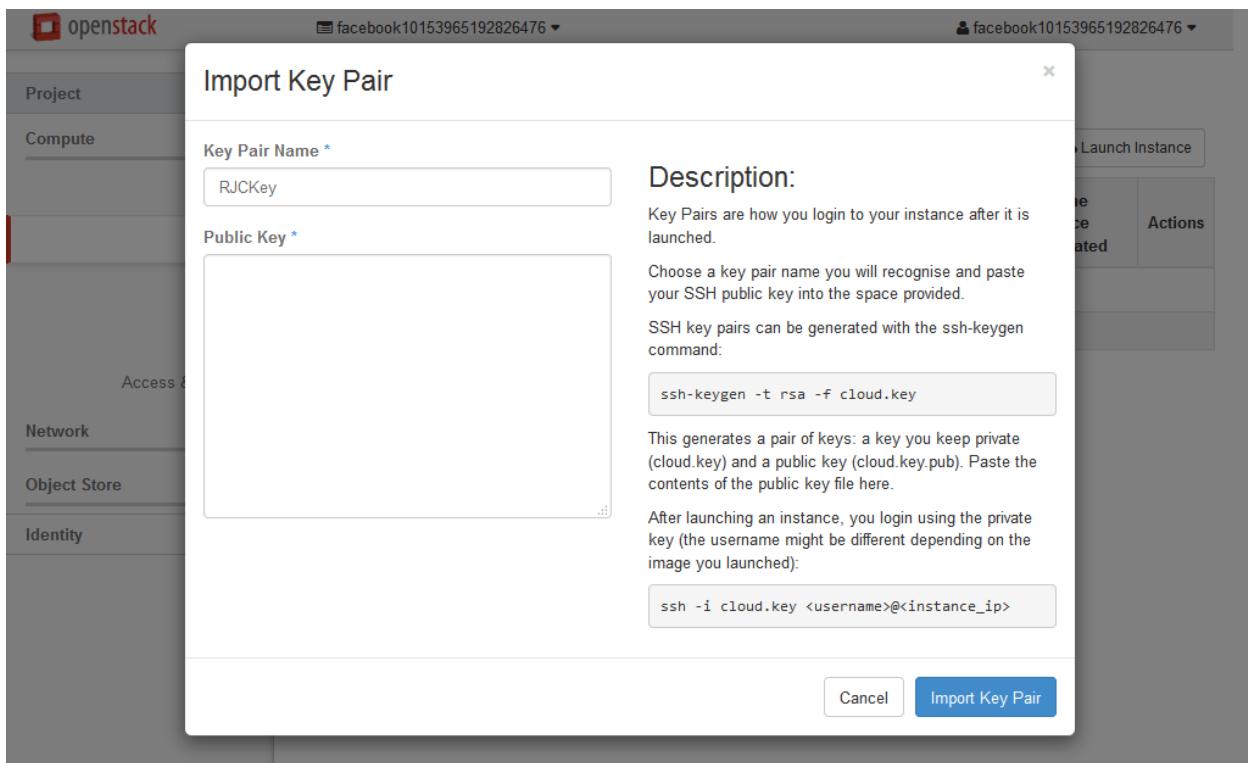


Go to Access & security



Click on “+“ Sign

Cloud Computing - II



Install SSH Package of cygwin terminal for creating SSH key

Execute given command in cygwin

Two keys will be generated (1 Public & 1 Private).....import the public key

Cloud Computing - II

```
MSC@MSC-2 ~
$ ssh-keygen -t rsa -f cloud.key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cloud.key.
Your public key has been saved in cloud.key.pub.
The key fingerprint is:
SHA256:qviaco1NhcaR9R8/Y9YkIPyprduFMB0Pfbklia1++v8 MSC@MSC-2
The key's randomart image is:
+---[RSA 2048]----+
|       o..o.
|       o...o.
|       o .+...o.
|       . o ..*o=+
|       + ..So.B^o.
|       o ..o =ooo.
|       . o . . =   .
|       + = . o     .
|       =o+       .E
+---[SHA256]-----+
MSC@MSC-2 ~
$ ls
cloud.key  cloud.key.pub

MSC@MSC-2 ~
$ cat cloud.key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCoVdIKZvpC02vPif941fbYA850ZjySYQ4r20vz2qU3Iv1IDVCuzgw
m8YNdoAIC88Pnw4TKnGRwMSyZekNeFuXpZ9RprBg2fQrpHbuTBzSpMz0p3lDyRz0S9xKpj+r+vYhZ31tsAEosMQGOTM
jYsijoAkb1DKnZqR1DYz7Paz9/I0Cse3uwJa1yKDUFLQO/H20okd2NwBcfWFYT0iv/7cejsP08j1F0pisqCCTgbCpL
ptTwo3dbOVRP5GVa4d3Wo6SAPlcw9ikASfvuHf7yEtqs2GirG/EBbWBjwx8f+Vi1bDbM2ME2zt/4gzXd5mD8cZmuXkP
czmwt15JH+OEAvvh MSC@MSC-2

MSC@MSC-2 ~
```

Select public key from terminal as shown below

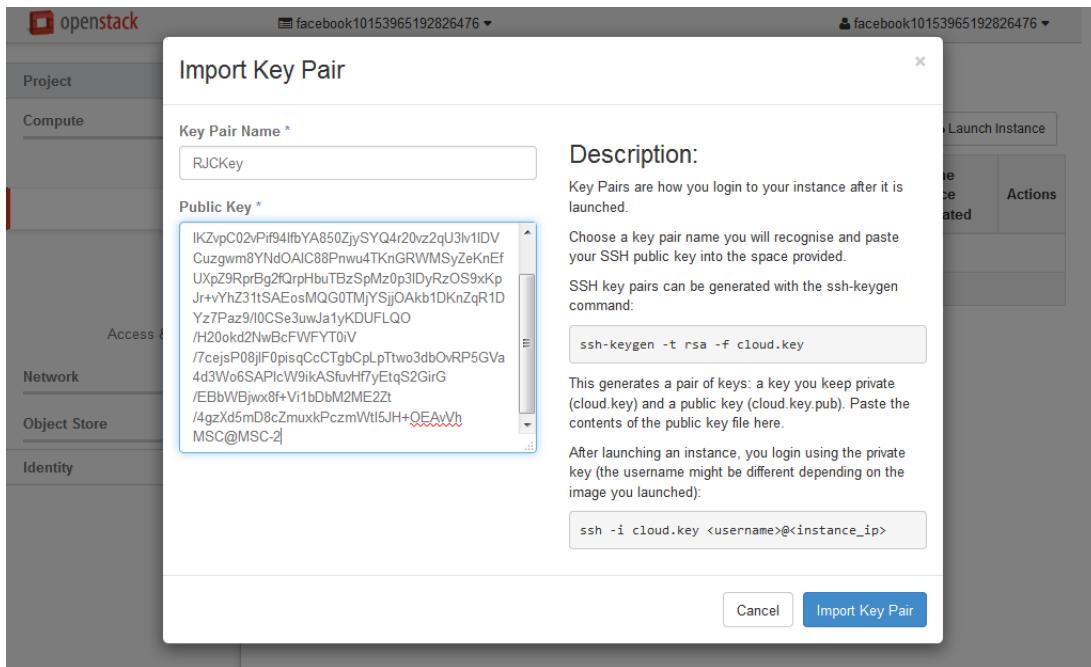
```
MSC@MSC-2 ~
$ ssh-keygen -t rsa -f cloud.key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cloud.key.
Your public key has been saved in cloud.key.pub.
The key fingerprint is:
SHA256:qviaco1NhcaR9R8/Y9YkIPyprduFMB0Pfbklia1++v8 MSC@MSC-2
The key's randomart image is:
+---[RSA 2048]----+
|       o..o.
|       o...o.
|       o .+...o.
|       . o ..*o=+
|       + ..So.B^o.
|       o ..o =ooo.
|       . o . . =   .
|       + = . o     .
|       =o+       .E
+---[SHA256]-----+
MSC@MSC-2 ~
$ ls
cloud.key  cloud.key.pub

MSC@MSC-2 ~
$ cat cloud.key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCoVdIKZvpC02vPif941fbYA850ZjySYQ4r20vz2qU3Iv1IDVCuzgw
m8YNdoAIC88Pnw4TKnGRwMSyZekNeFuXpZ9RprBg2fQrpHbuTBzSpMz0p3lDyRz0S9xKpj+r+vYhZ31tsAEosMQGOTM
jYsijoAkb1DKnZqR1DYz7Paz9/I0Cse3uwJa1yKDUFLQO/H20okd2NwBcfWFYT0iv/7cejsP08j1F0pisqCCTgbCpL
ptTwo3dbOVRP5GVa4d3Wo6SAPlcw9ikASfvuHf7yEtqs2GirG/EBbWBjwx8f+Vi1bDbM2ME2zt/4gzXd5mD8cZmuXkP
czmwt15JH+OEAvvh MSC@MSC-2

MSC@MSC-2 ~
```

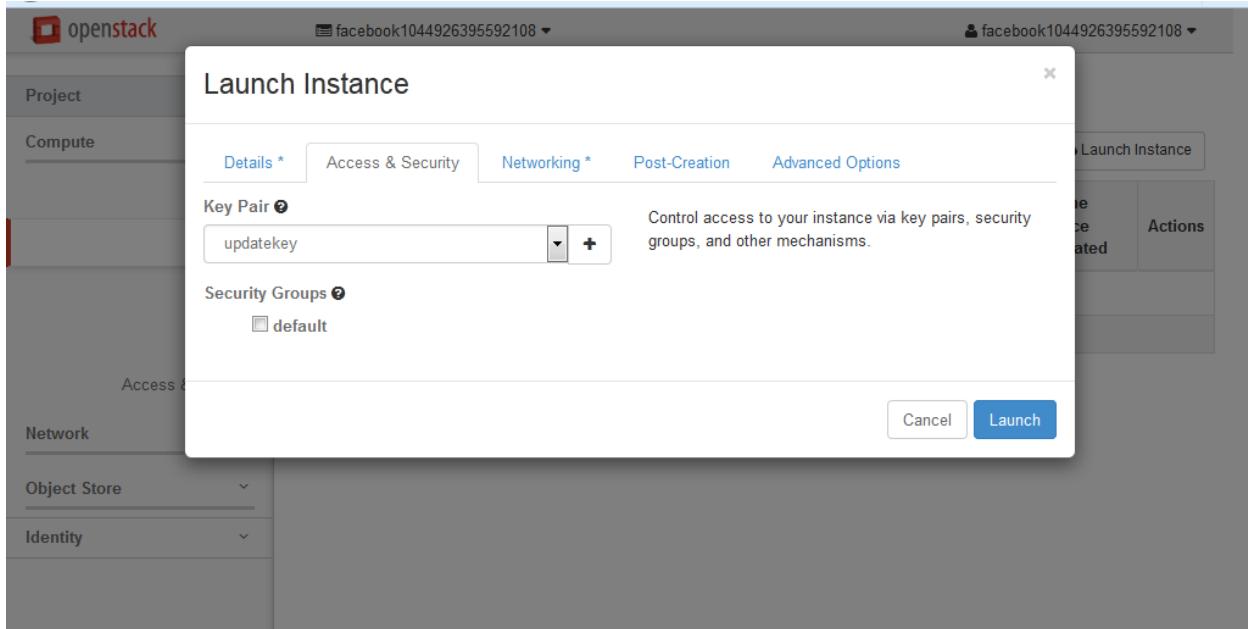
Cloud Computing - II

Now copy it as shown below



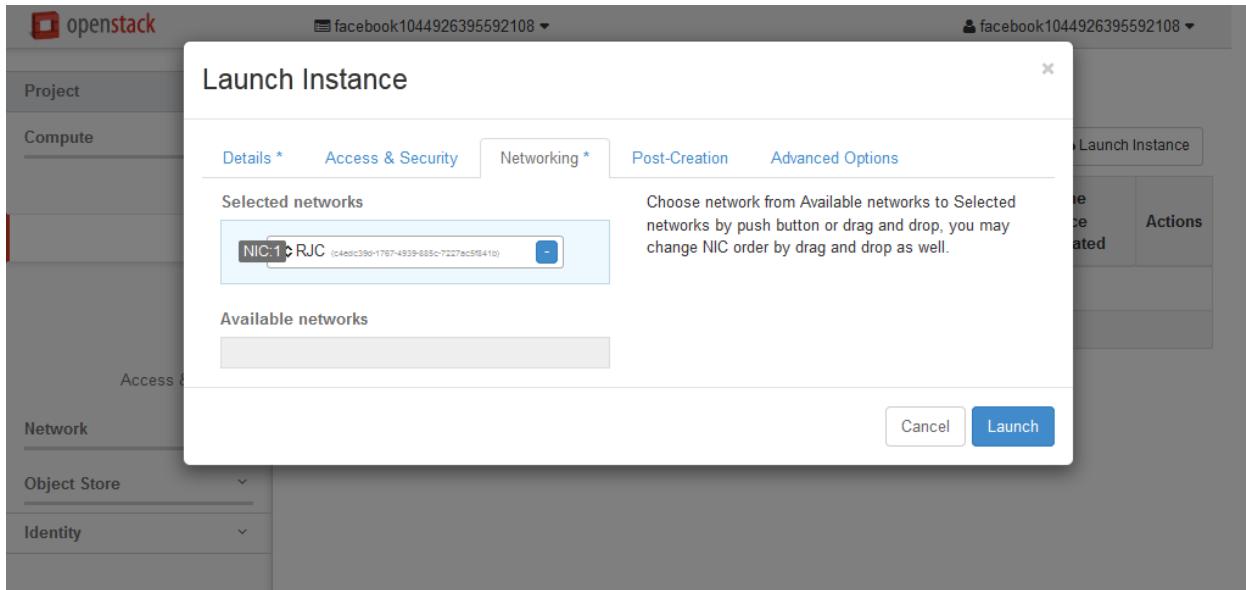
Click on Import key pair.....

Select name of the key from dropdown.....

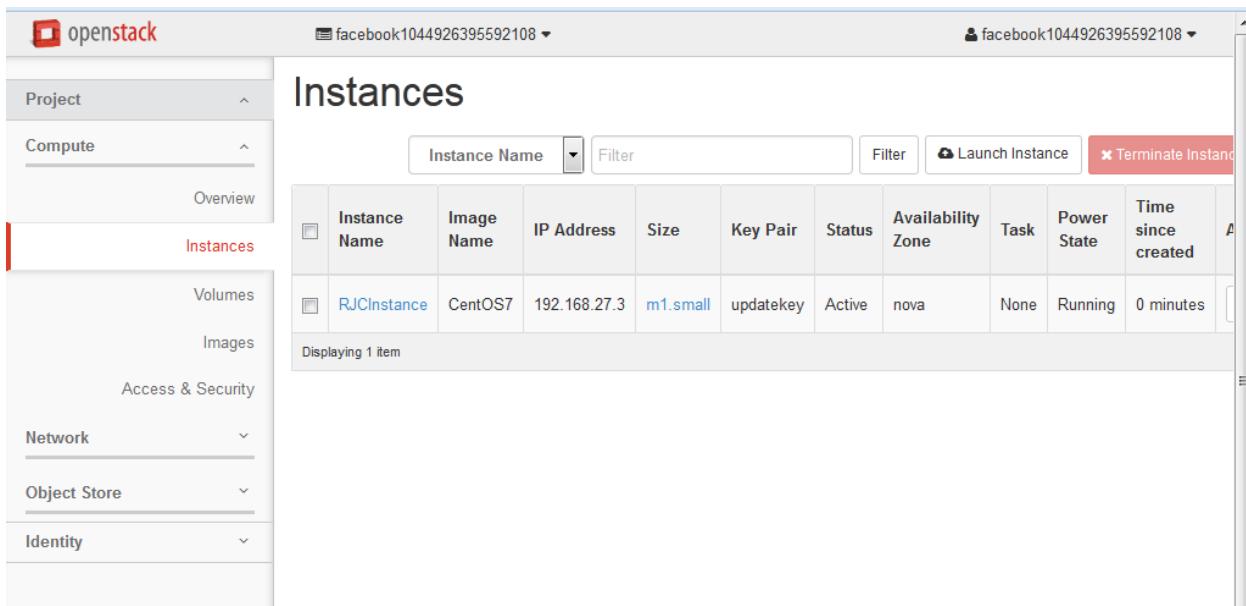


At networking tab

Cloud Computing - II



Click on Launch



Go to Network -> Routers -> Create Routers

Cloud Computing - II

The screenshot shows the OpenStack Horizon interface for creating a router. The left sidebar is visible with 'Network' selected. A modal dialog box is open with the title 'Create Router'. Inside the dialog, there are fields for 'Router Name *' (set to 'RJCRouter'), 'Admin State' (set to 'UP'), and 'External Network' (set to 'public'). To the right of the form, a 'Description:' section explains that it creates a router with specified parameters. At the bottom right of the dialog are 'Cancel' and 'Create Router' buttons.

Click on Create Router

The screenshot shows the OpenStack Horizon interface displaying a list of routers. The left sidebar has 'Routers' selected. The main area shows a table titled 'Routers' with one item listed. The table columns are 'Name', 'Status', 'External Network', 'Admin State', and 'Actions'. The single row shows 'RJCRouter' as the name, 'Active' as the status, 'public' as the external network, 'UP' as the admin state, and a red 'Clear Gateway' button under the actions column. A 'Filter' input field and a 'Delete Routers' button are also visible at the top of the table.

	Name	Status	External Network	Admin State	Actions
<input type="checkbox"/>	RJCRouter	Active	public	UP	<button>Clear Gateway</button>

Click on Name of Router (RJCRouter)

Cloud Computing - II

The screenshot shows the 'Router Details' page in the OpenStack interface. The sidebar on the left has 'Routers' selected under the 'Network' category. The main content area displays the router's name as 'RJCRouter', ID as '9349d0cb-6b1a-4a44-b46c-3b4461738b92', Project ID as '106610f9a1f242bbb62c6ac8db2a96ff', Status as 'Active', and Admin State as 'UP'. Under 'External Gateway', it shows the Network Name as 'public', Network ID as '1fd0a21e-e700-46ae-9f05-0b3164daafcc', Subnet ID as '7913ade1-01e3-44fa-9863-91867c0d23ab', and IP Address as '8.43.87.39'. The SNAT setting is listed as 'Enabled'.

Go to Interfaces->

The screenshot shows the 'Router Details' page with the 'Interfaces' tab selected. The sidebar on the left has 'Routers' selected under the 'Network' category. The main content area shows a table with columns: Name, Fixed IPs, Status, Type, Admin State, and Actions. A message at the top of the table area says 'No items to display.' Below it, another message says 'Displaying 0 items'.

Add Interface

Cloud Computing - II

Add Interface

Subnet *

RJC: 192.168.27.0/24 (RJCSUB)

IP Address (optional) (Optional)

Router Name *

RJCRouter

Router ID *

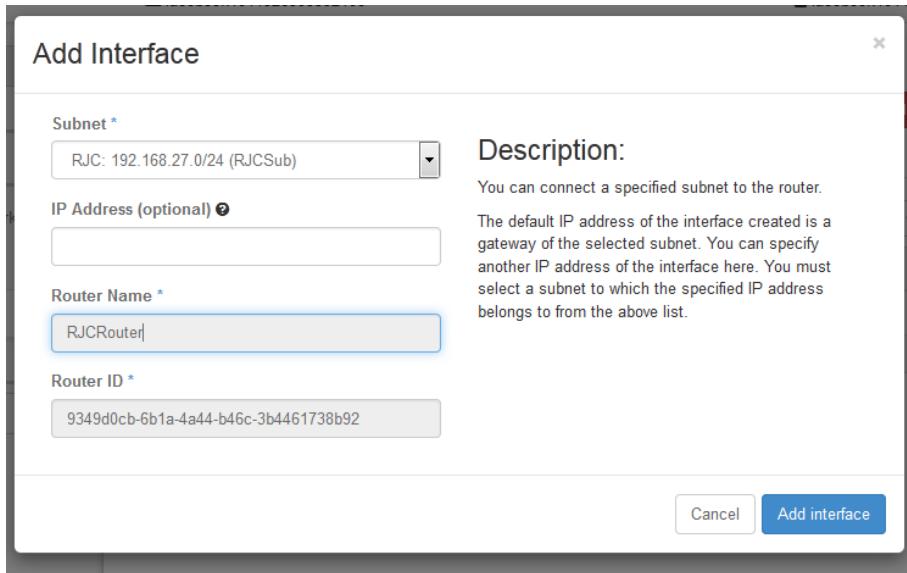
9349d0cb-6b1a-4a44-b46c-3b4461738b92

Description:

You can connect a specified subnet to the router.

The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs from the above list.

Cancel Add interface



Click on Add Interface

openstack

facebook1044926395592108 ▾

Project

Compute

Network

Network Topology

Networks

Routers

Object Store

Identity

Router Details

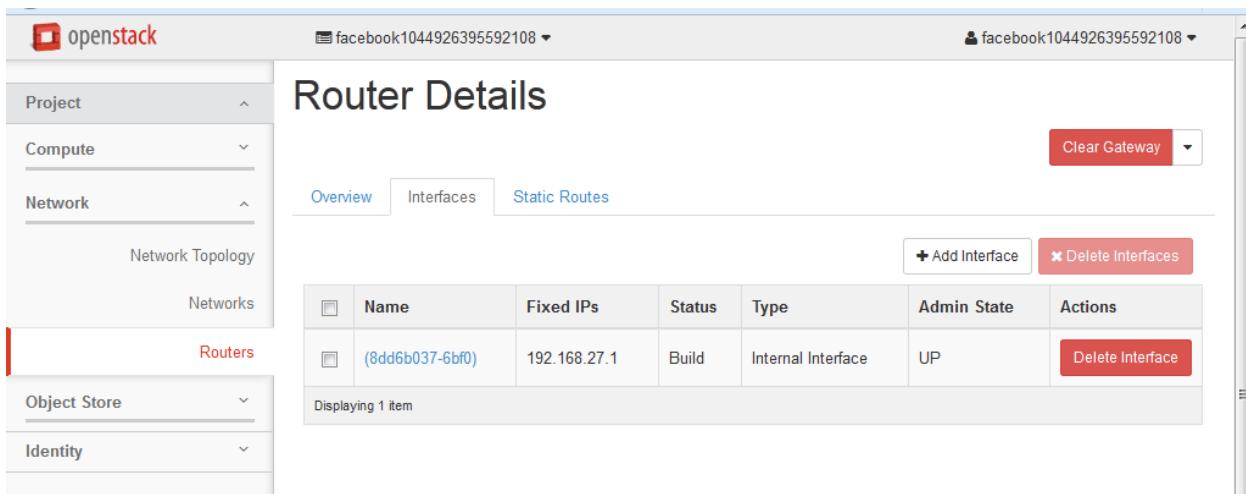
Overview Interfaces Static Routes

Clear Gateway ▾

+ Add Interface ✖ Delete Interfaces

	Name	Fixed IPs	Status	Type	Admin State	Actions
<input type="checkbox"/>	(8dd6b037-6bf0)	192.168.27.1	Build	Internal Interface	UP	<button>Delete Interface</button>

Displaying 1 item



Go to Compute -> Acess & Security

Cloud Computing - II

The screenshot shows the OpenStack Access & Security interface. On the left, there is a sidebar with navigation links: Project (Compute), Overview, Instances, Volumes, Images, and Access & Security (which is selected). Below these are Network, Object Store, and Identity options. The main content area is titled "Access & Security" and contains tabs for Security Groups, Key Pairs, Floating IPs, and API Access. The Security Groups tab is active, showing a table with one item: "default" (Description: Default security group). There are buttons for "Filter", "+ Create Security Group", and "Delete Security Groups". A message at the bottom says "Displaying 1 item".

Click on Manage Rules

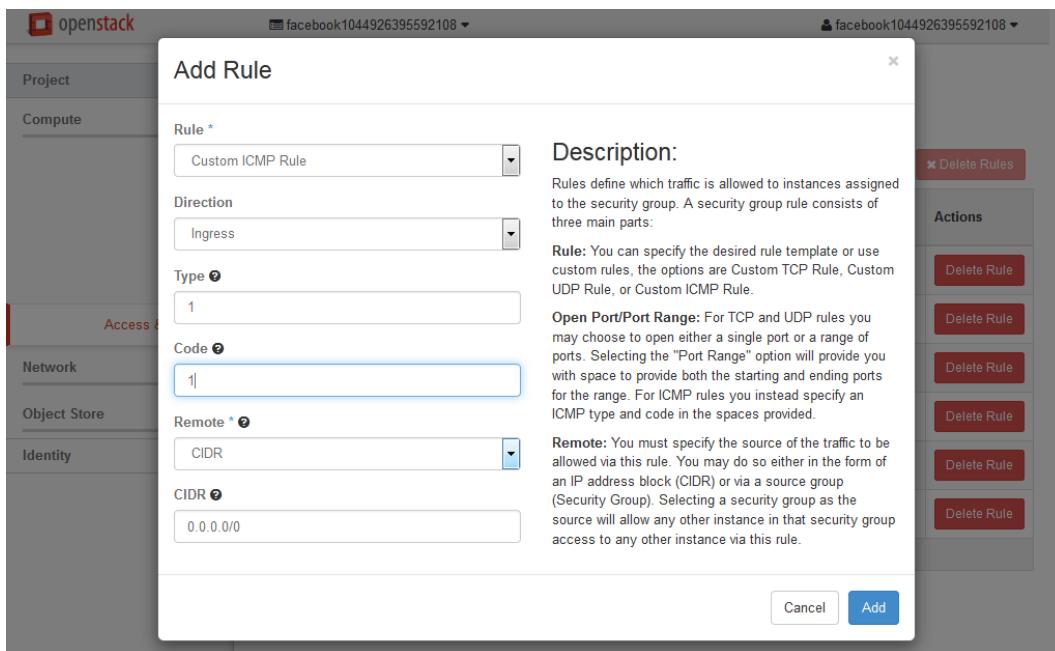
The screenshot shows the "Manage Security Group Rules" interface for the "default" security group. The top bar shows the project name "facebook1044926395592108". The main title is "Manage Security Group Rules: default (f75cb1d2-2f94-4e2b-902c-aa9b6caee607)". The interface includes a header with "+ Add Rule" and "Delete Rules" buttons. Below is a table listing six rules:

	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
1	Ingress	IPv6	Any	Any	-	default	<button>Delete Rule</button>
2	Egress	IPv6	Any	Any	::/0	-	<button>Delete Rule</button>
3	Ingress	IPv4	Any	Any	-	default	<button>Delete Rule</button>
4	Egress	IPv4	Any	Any	0.0.0.0/0	-	<button>Delete Rule</button>
5	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	<button>Delete Rule</button>
6	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	<button>Delete Rule</button>

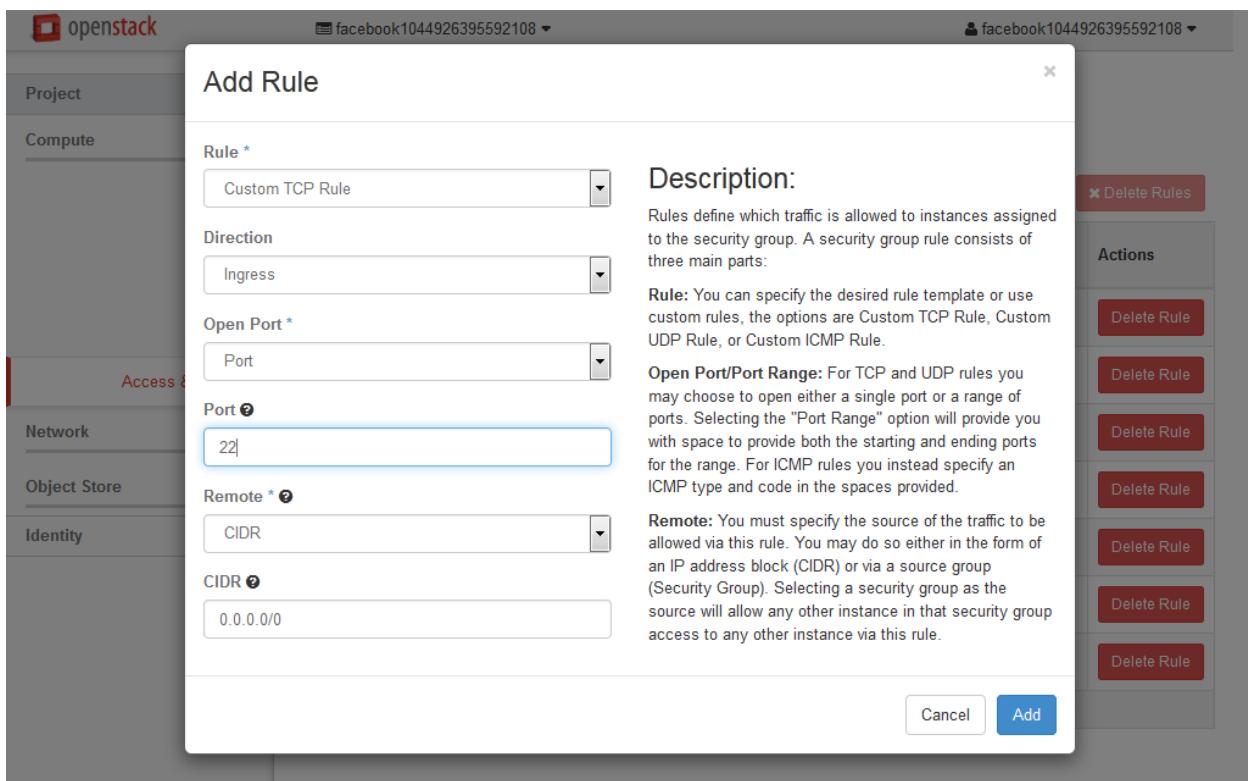
At the bottom, it says "Displaying 6 items".

Click on AddRule & select following options

Cloud Computing - II



Again click on AddRule & now insert following options



Cloud Computing - II

After adding 2 rules

The screenshot shows the OpenStack Compute interface under the 'Compute' project. The left sidebar has 'Compute' selected. The main area is titled 'Manage Security Group Rules: default (f75cb1d2-2f94-4e2b-902c-aa9b6caee607)'. It displays a table of security group rules:

	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Ingress	IPv6	Any	Any	-	default	<button>Delete Rule</button>
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv4	Any	Any	-	default	<button>Delete Rule</button>
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	<button>Delete Rule</button>

Displaying 6 items

Go to Instancedrop down create snapshot & select Associate floating IP

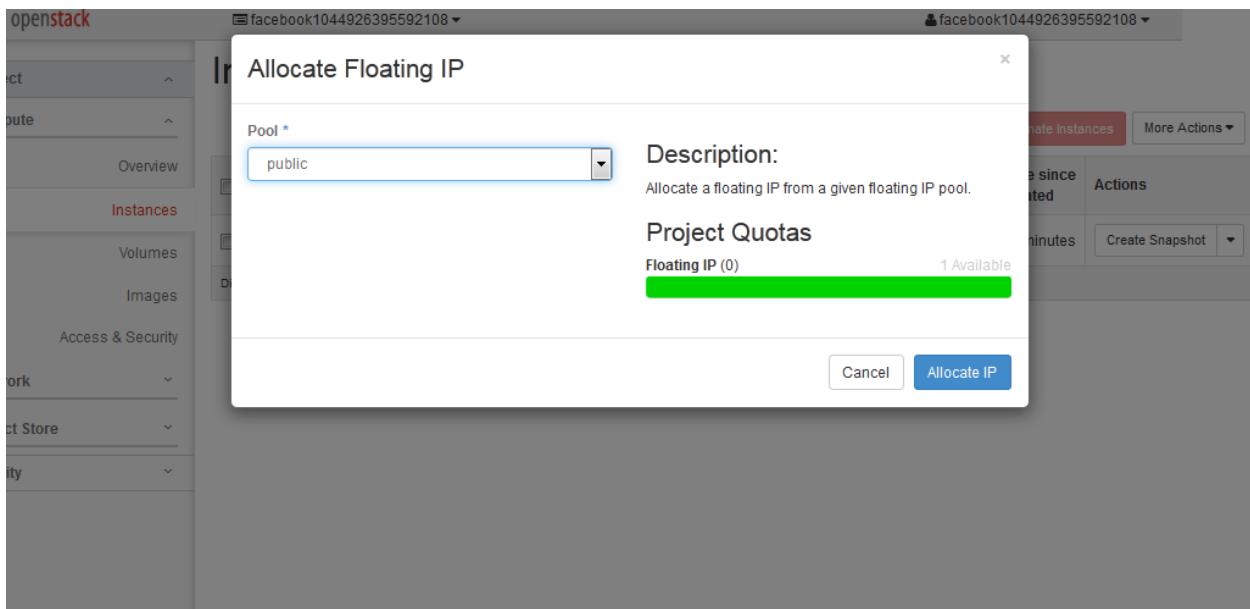
Cloud Computing - II

The screenshot shows the OpenStack Instances interface. On the left is a sidebar with links for Overview, Instances, Volumes, Images, Access & Security, Store, and Compute. The main area is titled "Instances" and displays a table with one item: "RJCInstance" (CentOS7, 192.168.27.3, m1.small, updatekey, Active, nova, None, Running, 18 minutes). A context menu is open over the "RJCInstance" row, listing options such as Associate Floating IP, Attach Interface, Detach Interface, Edit Instance, Edit Security Groups, Console, View Log, Pause Instance, Suspend Instance, Shelve Instance, Resize Instance, Lock Instance, Unlock Instance, Soft Reboot Instance, Hard Reboot Instance, Shut Off Instance, and Rebuild Instance.

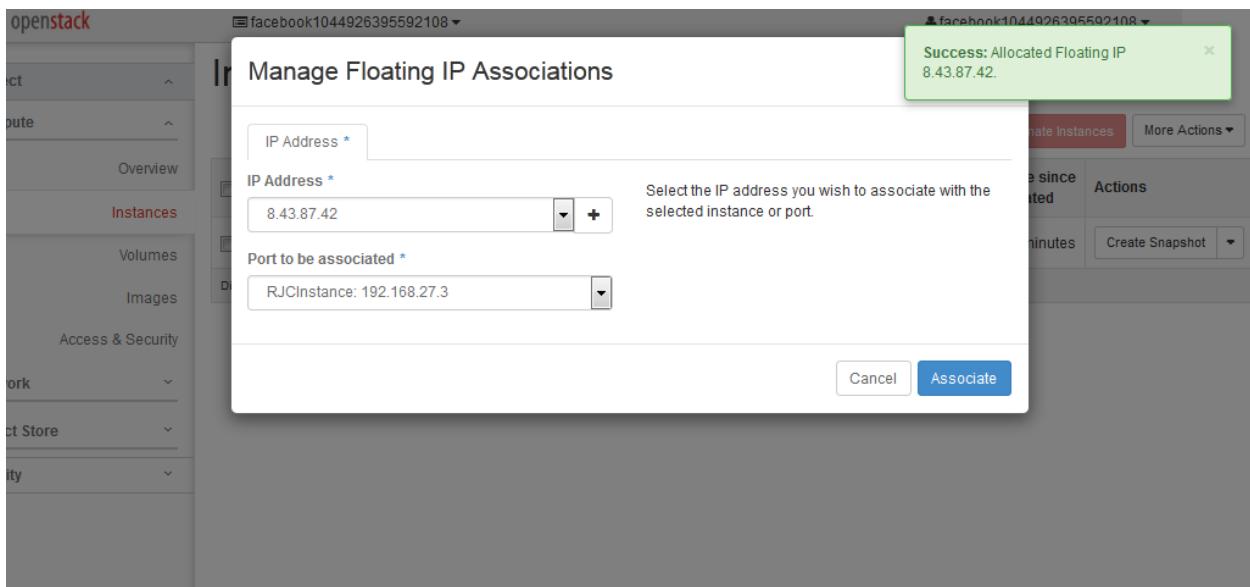
The screenshot shows the "Manage Floating IP Associations" dialog box. It has fields for "IP Address" (dropdown showing "No floating IP addresses allocated") and "Port to be associated" (dropdown showing "RJCInstance: 192.168.27.3"). A green success message at the top right says "Success: Allocated Floating IP 8.43.87.42." At the bottom are "Cancel" and "Associate" buttons.

Click on “+” sign

Cloud Computing - II



Click on Allocate IP



Click on Associate

Cloud Computing - II

The screenshot shows the OpenStack Instances dashboard. On the left, a sidebar menu includes 'Compute' (selected), 'Overview', 'Instances' (highlighted with a red border), 'Volumes', 'Images', 'Access & Security', 'Network', 'Object Store', and 'Identity'. The main area is titled 'Instances' and displays a table with one item:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
RJInstance	CentOS7	192.168.27.3 Floating IPs: 8.43.87.42	m1.small	updatekey	Active	nova	None	Running	25 minutes	<button>Create Snapshot</button>

Displaying 1 item

Go to Network -> Network Topology

The screenshot shows the OpenStack Network Topology dashboard. On the left, a sidebar menu includes 'Compute' (selected), 'Network' (highlighted with a red border), 'Network Topology' (selected), 'Networks', 'Routers', 'Object Store', and 'Identity'. The main area is titled 'Network Topology' and displays a network diagram. It features a central cloud icon connected to a monitor icon. A blue oval surrounds the cloud icon, and two arrows point from the monitor icon towards the cloud. Below the diagram are buttons: 'Toggle labels', 'Toggle Network Collapse', 'Launch Instance', 'Create Network', and 'Create Router (Quota exceeded)'.

Conclusion : Successfully evaluated a private cloud.

Cloud Computing - II

Practical No: 06

Aim: Implement FOSS-Cloud Functionality - VDI (Virtual Desktop Infrastructure)

FOSS-Cloud (FOSS-Cloud software and hardware) is an integrated and redundant server infrastructure to provide virtualization- and cloud-services, Windows or Linux based SaaS-, Terminal Server-, Virtual Desktop Infrastructure (VDI) or virtual server environments.

FOSS-Cloud covers all aspects of a virtualized IT environment. FOSS-Cloud is a pure Open Source solution, is licensed under [EUP](#) and is available on the sourceforge.net. FOSS-Cloud is the most advanced Open Source Cloud.

FOSS-Cloud is a cost-effective alternative to Citrix and VMware. We can Build our own private- or public-Cloud!

Functionality

- VDI (Virtual Desktop Infrastructure)
- VSI (Virtual Server Infrastructure)
 - Infrastructure as a Service (IaaS)
 - Platform as a Service (PaaS)
 - Software as a Service (SaaS)
- Storage Cloud

Features

- Full integration into existing Windows and Linux environments
- Cloud for server- and desktop virtualization
- Powerful virtualization for Windows and Linux 32/64bit
- Published Desktop
- Persistent virtual machines including session transfer to other devices
- Dynamic desktop with Golden Image to serve user groups
- Application streaming
- Published application support with RDS
- Video streaming (M-Jpeg)
- High resolution display
- Pools of network- and hardware-resources or virtual machines
- VDI access through Windows and Linux, PXE boot and handhelds
- Bi-directional audio and video
- Smartcard authentication (including pass through)
- USB redirection

Cloud Computing - II

- Web-based management console
- Multi-tenancy

Minimal Requirements

Your server should fulfill the following minimal requirements on each node:

- Dedicated hardware
- 64-Bit Intel with VT-Technology
- 4 Gigabyte memory
- 320 Gigabyte disk space
- Network interface

The **Demo System** is really just a playing environment and not for professional use. It is very easy to **install**. The Demo System is made to get a feeling of the possibilities of the FOSS-Cloud.

Limitation of Demo System

There is no access to the **VMs** from outside of FOSS-Cloud. That means, for example, pinging a VM is not possible.

Steps to implement FOSS Cloud Demo System

1. Create FOSS-Cloud Boot Stick or DVD

Linux

First download the appropriate iso image from sourceforge:

```
root # wget http://sourceforge.net/projects/foss-cloud/files/X.X.X/foss-cloud-installer-X.X.X.iso
```

Most modern LiveCD's, like Gentoo are already in hybrid mode. If your LiveUSB doesn't boot then you have to convert it to hybrid mode. Hybrid mode means image that can be booted from either DVD or USB drive.

```
root # isohybrid foss-cloud-installer-X.X.X.iso
```

Write the LiveCD image to the USB drive

Cloud Computing - II

root # dd if=foss-cloud-installer-X.X.X.iso of=/dev/sdb bs=4M
Check the USB device. In our case it

To create a live DVD on a Windows-PC, is /dev/sdb.

Windows

1. Do a right-click on the ISO-image and burn the DVD. To create a liveUSB-stick, download a tool like "liveUSB creator"
2. In the BIOS setup, enable VT technology for the CPU Options
3. Insert the FOSS-Cloud CD and boot the server.

Answer the questions as follows (the bold values are examples can be set through the administrator and are variable, according to the local setup):

Warning: ALL DATA WILL BE LOST!

1. Choose your keymap

1 azerty	2 be	3 bg
4 br-a	5 br-l	6 by
7 cf	8 croat	9 cz
10 de	11 dk	12 dvorak
13 es	14 et	15 fi
16 fr	17 gr	18 hu
19 il	20 is	21 it
22 jp	23 la	24 lt
25 mk	26 nl	27 no
28 pl	29 pt	30 ro
31 ru	32 se	33 sg
34 sk-y	35 sk-z	36 slovene
37 trf	39 ua	40 uk
41 us	42 wangbe	43 fr_CH
44 speakup	45 cs_CZ	46 de_CH
47 sg-lat1	48 fr-bepo	49 colemak

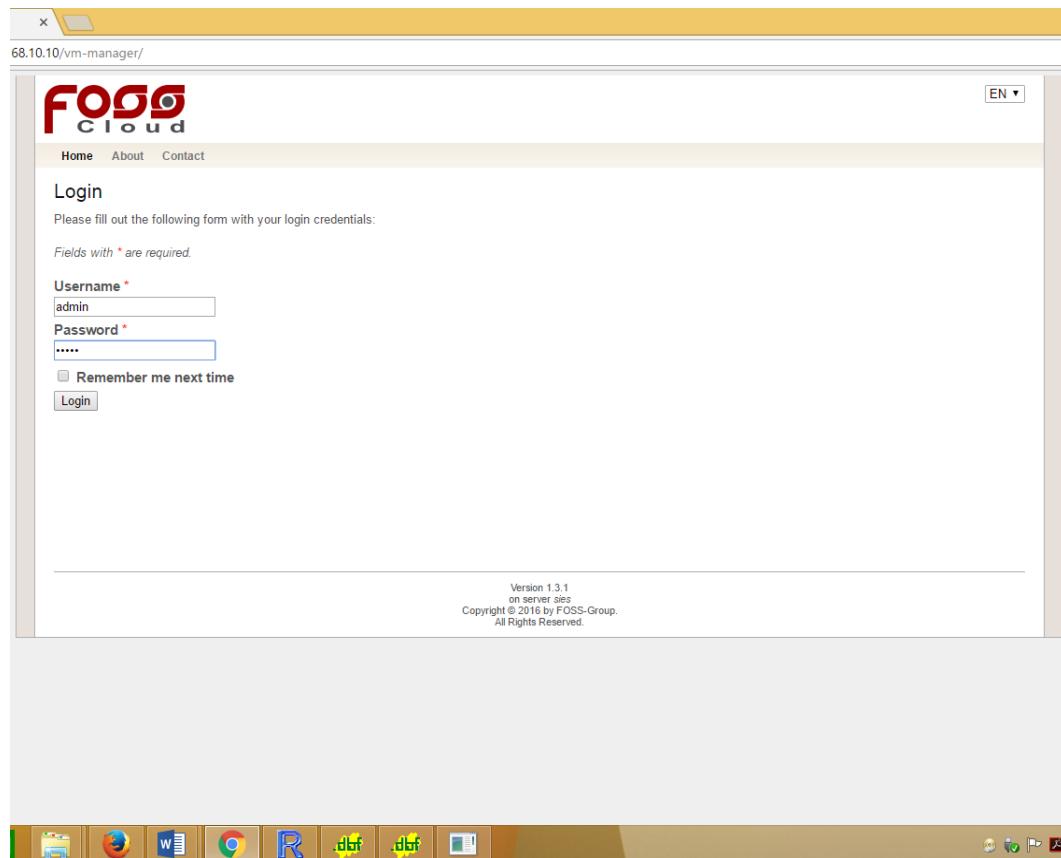
2. Confirm that you want to start: yes
3. Choose Single-Server: 1

Cloud Computing - II

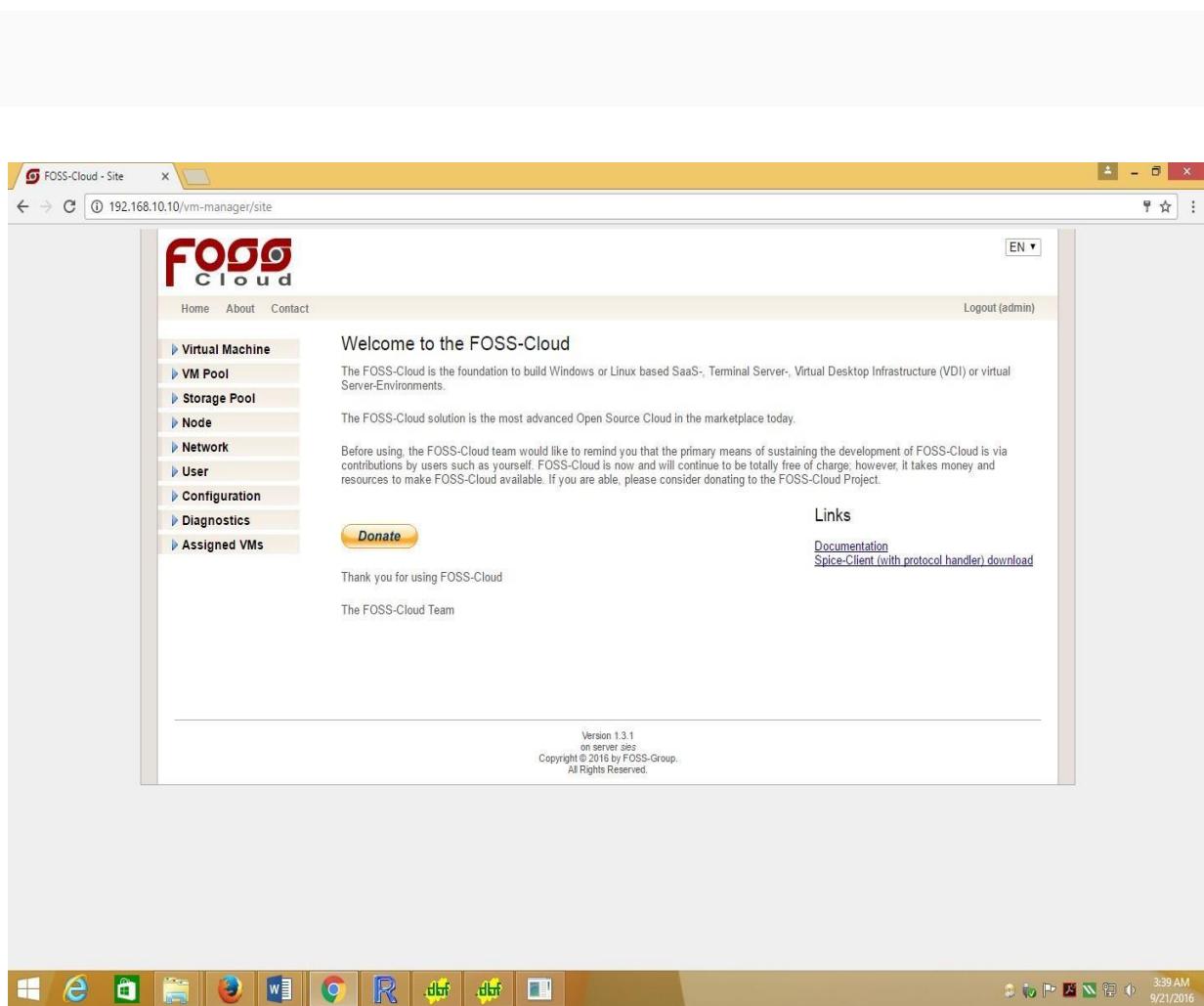
4. Choose a Block-Device: **sda**
5. Confirm that you want to continue: yes
6. Confirm that you want to continue: yes
7. Choose the network interface: **eth0**
8. Choose if you want to use automatic network configuration: no
9. Choose hostname: e.g. foss-cloud
10. Choose domain name: e.g. foss-cloud.org
11. Choose a free IP address in your home network: **192.168.10.10**
12. Choose the network mask in CIDR format: **24**
13. Choose the broadcast IP address: **192.168.1.255**
14. Confirm your entries: yes
15. Enter the IP address of your gateway: **192.168.1.1**
16. Enter the IP address of your DNS resolver: **192.168.1.1**
17. Do you like another DNS resolver: no
18. Confirm your entries: yes
19. Reboot your system: yes
20. Install [Patch: Missing roles of VM-User](#)

Access your IP **192.168.10.10** with a web browser and login with *admin* and the password *admin*

Cloud Computing - II



Home page of FOSS Cloud at Client



Uploading ISO files to foss cloud

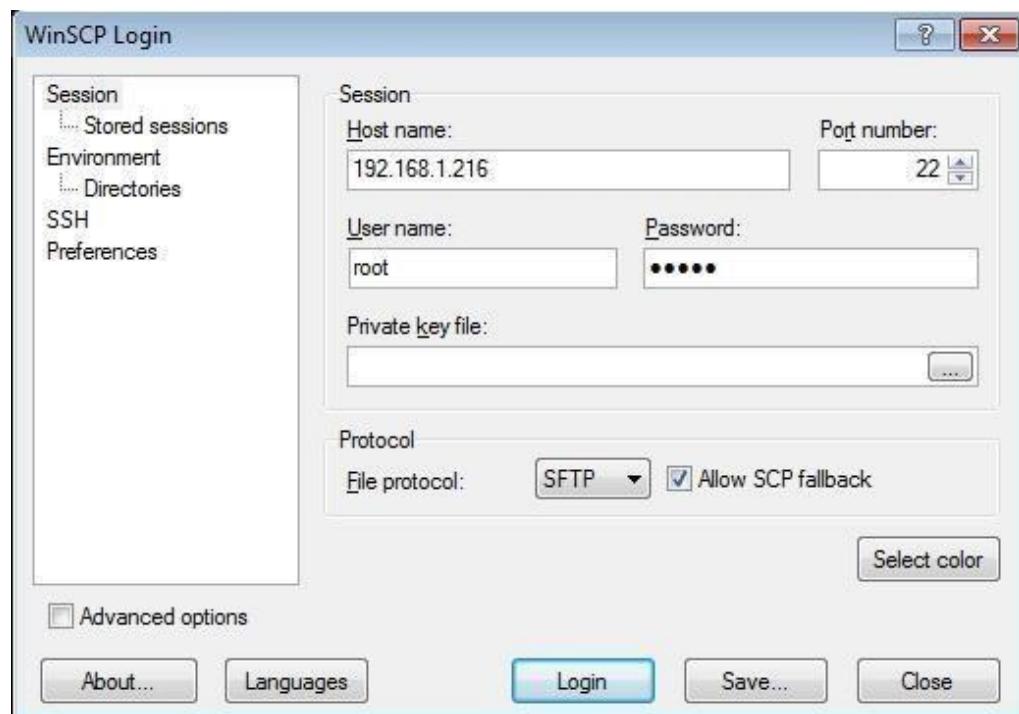
ISO files need to be uploaded manually.

Windows

- Local Windows computer with WinSCP installed ([Download WinSCP](#))
- You've downloaded a ISO-File to your computer. This example is done with: ubuntu-11.04-desktop-i386.iso
- The IP-address of the FOSS-Cloud Node. This example is done with: 192.168.1.216

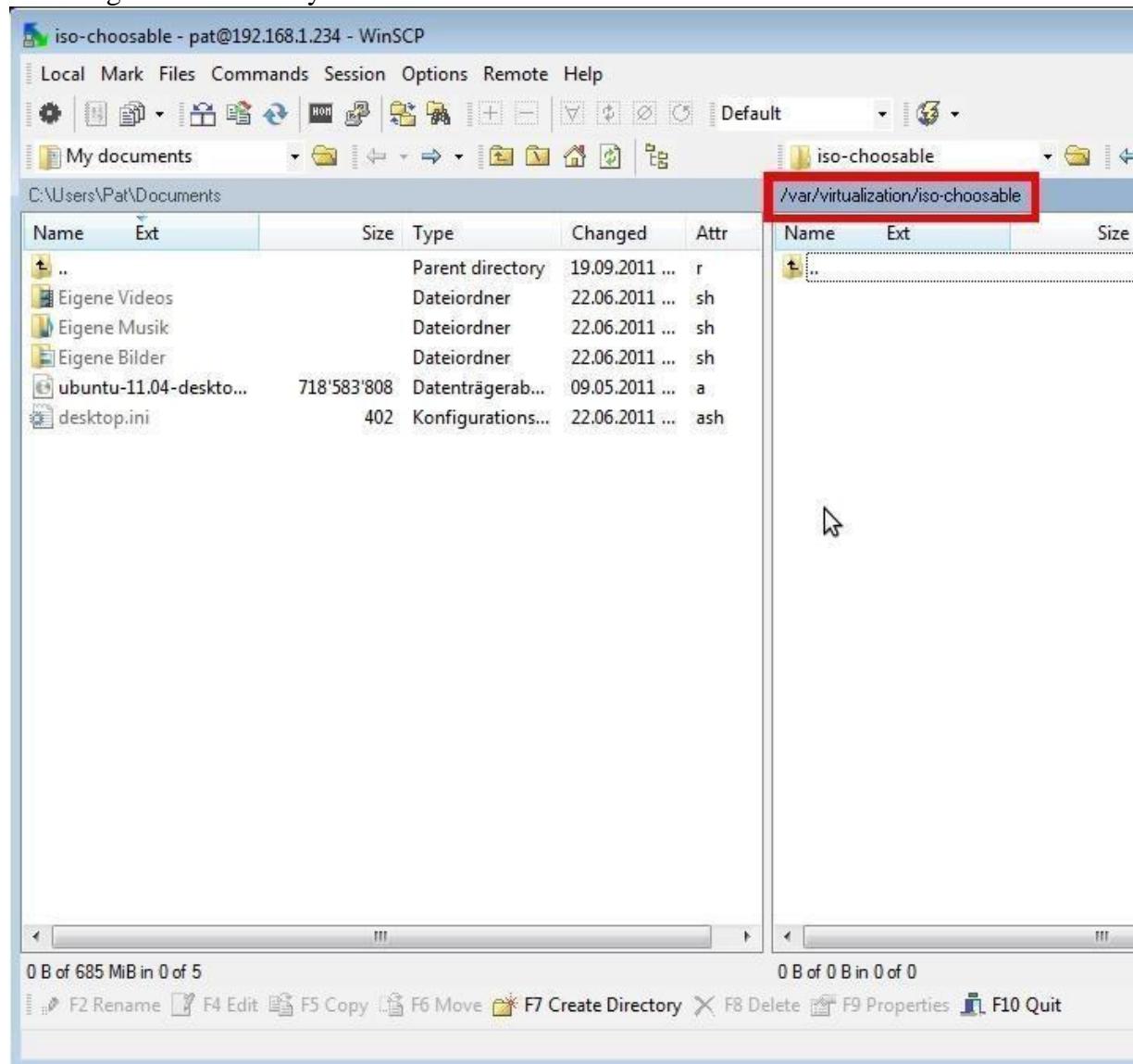
Run WinSCP and establish a connection to the FOSS-Cloud Node:

username: root password: admin:



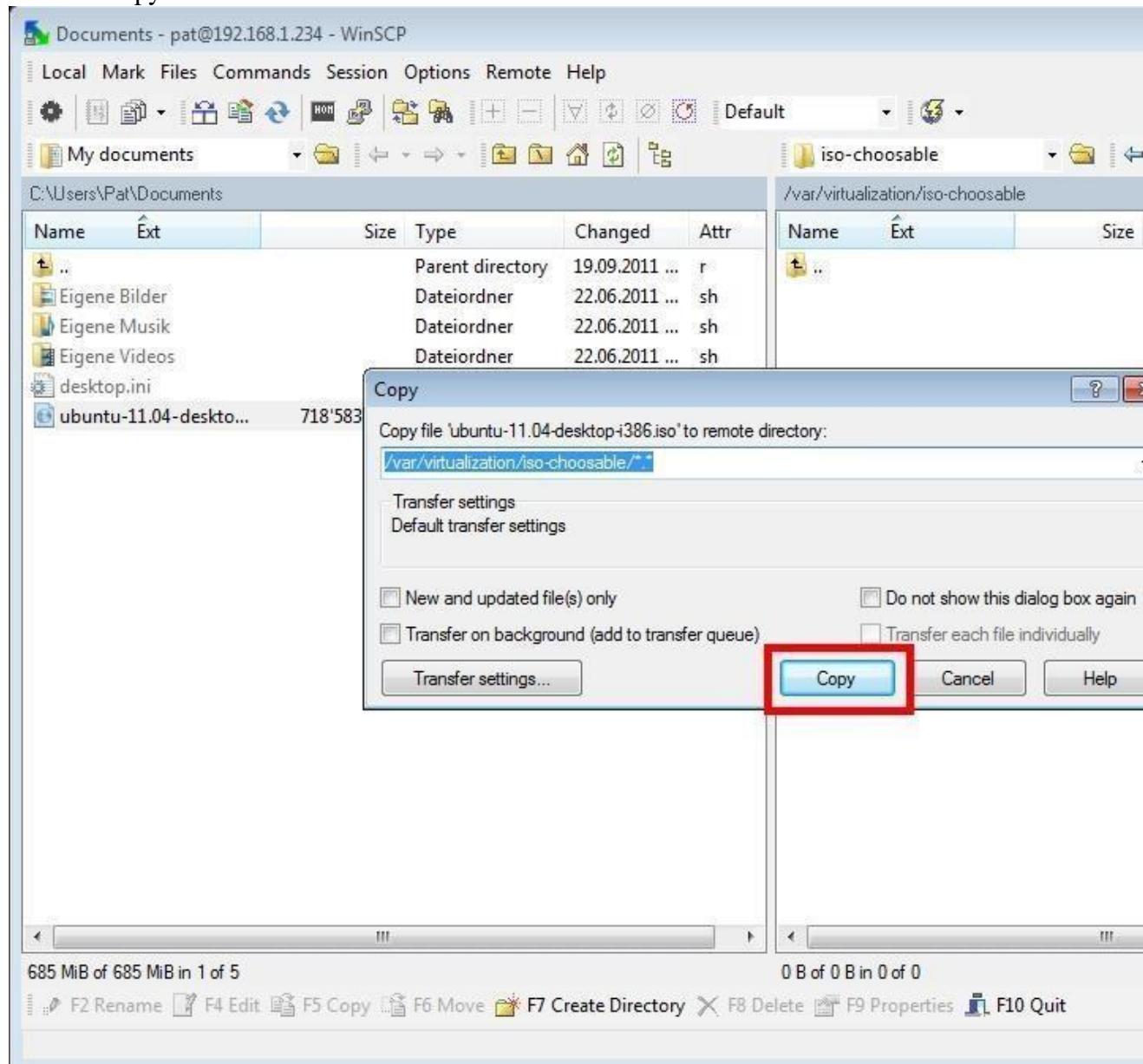
Cloud Computing - II

On the guest machine change to the '/var/virtualization/iso-choosable' directory. On the host, change to the directory the ISO-file is located:



Cloud Computing - II

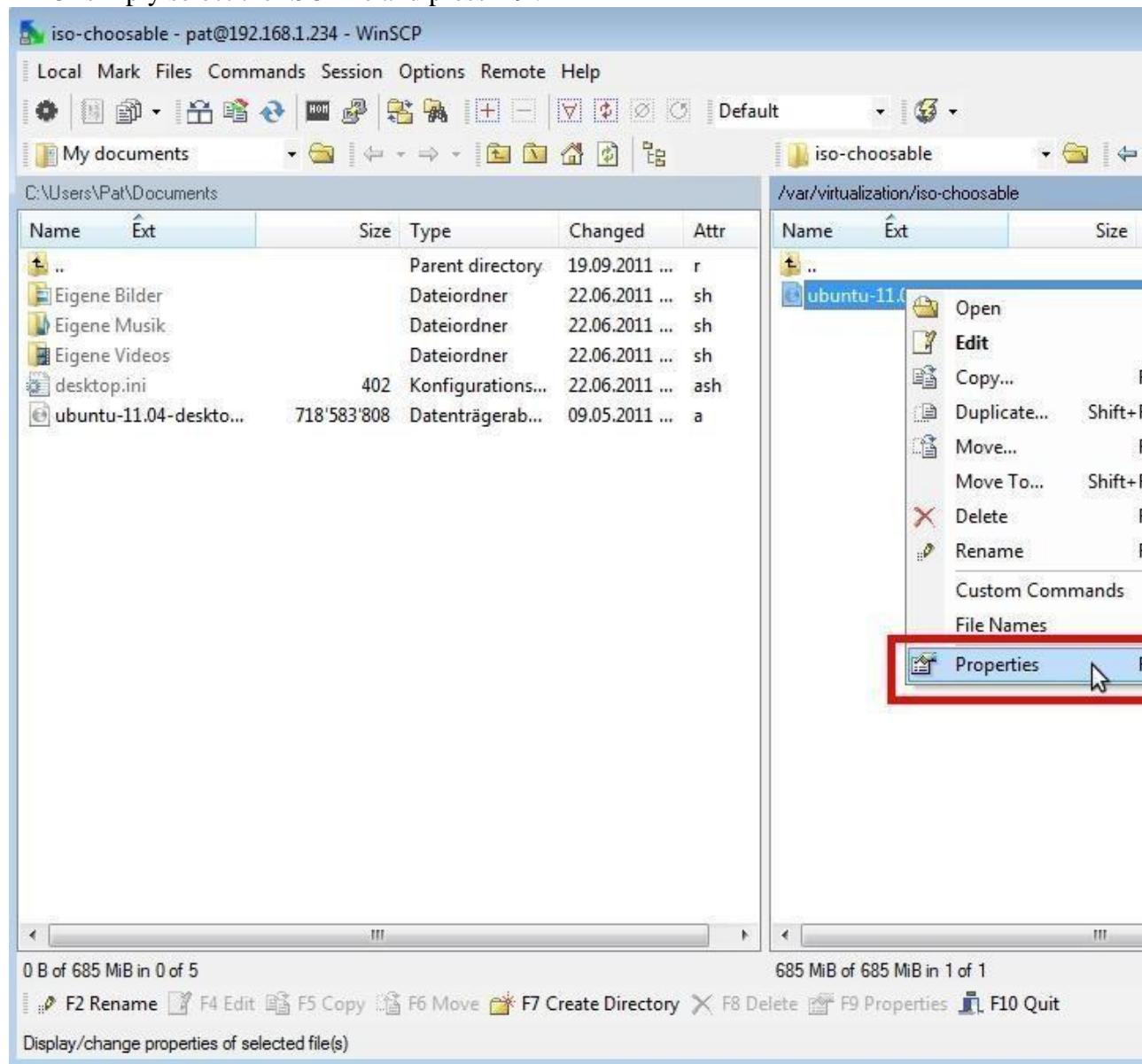
Drag (Drag & Drop) the ISO-file in the '/var/virtualization/iso-choosable' directory and commit the copy:



After the copy, right-click on the ubuntu-11.04-desktop-i386.iso file in the '/var/virtualization/iso-choosable' directory and select 'properties'.

Cloud Computing - II

Or simply select the ISO-file and press 'F9':



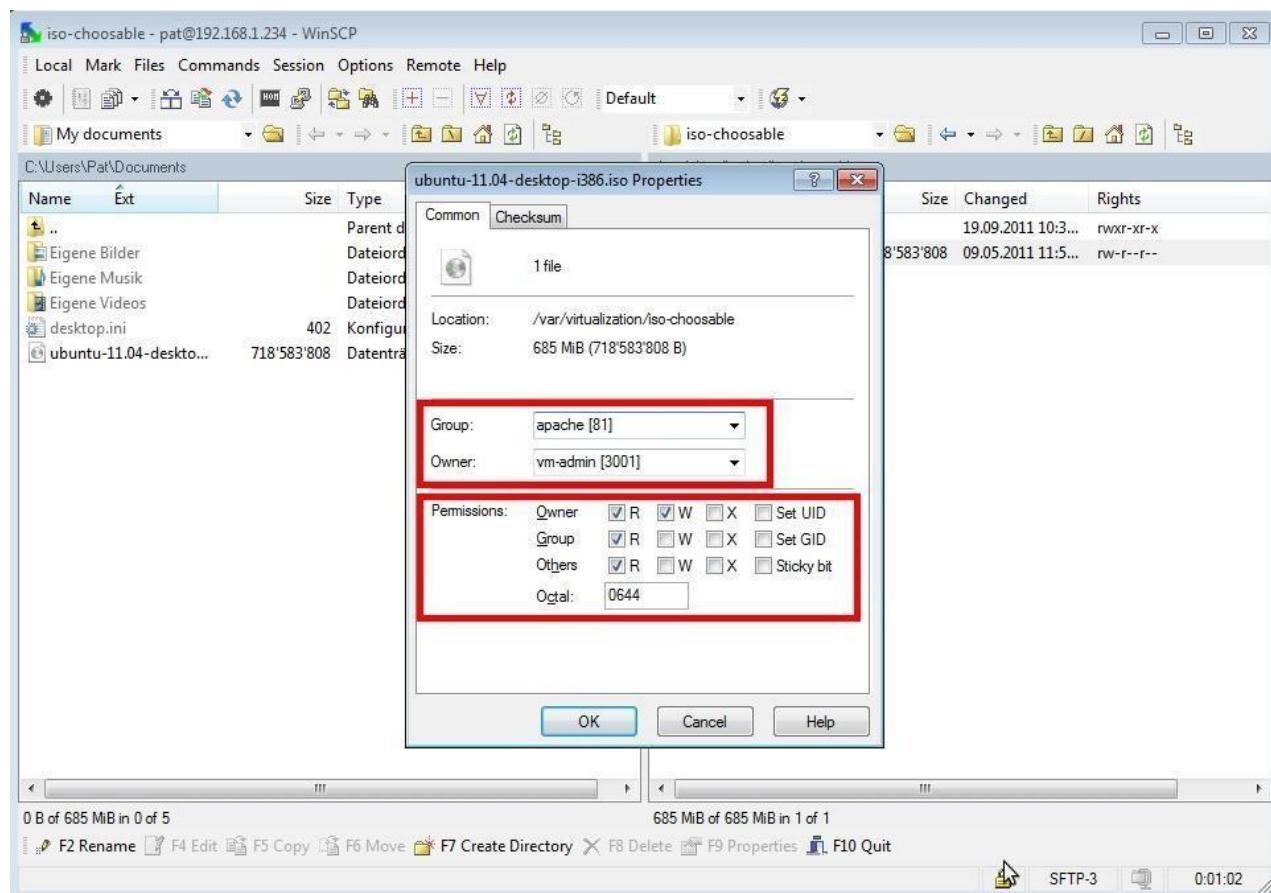
Cloud Computing - II

Set the files ownership as follows:

- Group: root
- Owner vm-storage **Caution the UID may be different to 3000**

And the rights:

- Owner: R and W
- Group: R
- Other: R Or octal: 0644



When you create a new **Virtual Machine Profile**, the uploaded ISO-file will be listed.

Cloud Computing - II

Linux

Upload ISO with scp

- Local Linux computer with shell access.
- You've downloaded a ISO-File to your computer. This example is done with openSUSE-11.3-KDE4-LiveCD-i686.iso
- The IP-address of the FOSS-Cloud Node is 192.168.1.216.

Upload the ISO-file with Secure Copy (scp). The root password is admin:

```
root # scp openSUSE-11.3-KDE4-LiveCD-i686.iso  
root@192.168.1.216:/var/virtualization/iso-choosable/.
```

Make sure, that you have the correct permissions on the newly uploaded file:

```
root # cd /var/virtualization/iso-choosable
```

```
ls -al
```

```
-rw-r--r-- 1 root vm-storage 728760320 Mär 24 17:54 openSUSE-11.4-KDE-  
LiveCD-x86_64.iso.iso
```

If not, you can set them as follows:

```
root # chown root:vm-storage openSUSE-11.4-KDE-LiveCD-x86_64.iso.iso  
root # chmod 644 openSUSE-11.4-KDE-LiveCD-x86_64.iso.iso
```

When you create a new **Virtual Machine Profile**, the uploaded ISO-file will be listed. **Upload ISO with wget**

Login to a VM node as root, then execute the following command

```
root # cd /var/virtualization/iso-choosable/  
root # wget http://<your\_download\_Link/>/your_image.iso  
root # chown apache:vm-storage <your_image.iso>  
root # chmod 644 <your_image.iso>
```

Examples:

```
root # wget http://download.fedoraproject.org/pub/fedora/linux/releases/19/Live/x86\_64/Fe\_dora-  
Live-Desktop-x86\_64-19-1.iso  
root # wget http://mirror.switch.ch/ftp/mirror/centos/6.5/isos/x86\_64/CentOS-6.5-x86\_64-  
netinstall.iso
```

Practical No: 07

Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS)

Creating Virtual Machines

Step 1.Create a Profile

The screenshot shows the 'Create VM Profile' page of the FOSS-Cloud web interface. The left sidebar has a 'Virtual Machine' section with 'Create' selected. The main area shows 'Step I: Please select a profile first!' and 'Step II: Overwrite the default values if necessary!'. Under 'BaseProfile', there is a tree view of available ISO files. The 'linux' node is expanded, showing 'default' (selected), 'i686', 'x86_64', and several language variants (multi, de-DE, de-AT, de-CH, en-US, en-GB, fr-CH, fr-FR, it-CH, it-IT). Other nodes include 'ubuntu', 'Fedora', and 'windows'. To the right, configuration fields include 'Isofile *' (set to 'ubuntu-11.04-desktop-i386.iso'), 'Name *' (empty), 'Description *' (empty), 'Memory *' (128 MB), 'Volume Capacity *' (157 GB), 'CPU *' (1), and 'Clock Offset *' (utc). A 'Create' button is at the bottom.

2. Create VM Template

The screenshot shows the 'Create VmTemplate' page of the FOSS-Cloud interface. The left sidebar has a 'Virtual Machine' section with 'Create' selected, and other options like 'Persistent VMs', 'Dynamic VMs', 'VM Templates', 'Profiles', 'Storage Pool', 'Node', 'Network', 'User', 'Configuration', 'Diagnostics', and 'Assigned VMs'. The main area has two steps: Step I (Please select a profile first!) and Step II (Please choose a node and overwrite the default values if necessary!). Step II is active, showing a 'Profile' dropdown menu with 'linux' expanded, showing 'ubuntu', 'i686', 'multi' (selected), 'Fedora', and 'windows'. It also shows 'Vmpool *' (dropdown), 'Node *' (dropdown), 'Name *' (text input 'ubuntu'), 'Description *' (text input 'Ubuntu workstation'), 'Memory *' (slider from 128 MB to 2 GB at 2 GB), 'Volume Capacity *' (slider from 10 GB to 2048 GB at 15 GB), 'CPU *' (dropdown '1'), 'Clock Offset *' (dropdown 'utc'), and 'Number of displays' (dropdown '1'). The bottom navigation bar includes icons for Windows, Internet Explorer, File Explorer, Mozilla Firefox, Google Chrome, R, .dbf, .pdf, and Notepad, along with system status icons and a date/time stamp '3:46 AM 9/21/2016'.

3. Manage VM Templates

The screenshot shows the FOSS-Cloud VM Template management interface. The main title bar reads "FOSS-Cloud - VmTempla x". The address bar shows the URL "192.168.10.10/vm-manager/vmTemplate/index.html". The top right corner has a language dropdown set to "EN" and a "Logout (admin)" link. The left sidebar menu includes "Virtual Machine" (selected), "Persistent VMs", "Dynamic VMs", "VM Templates" (selected), "Create", "Profiles", "Create", "Upload ISO File", "VM Pool", "Storage Pool", "Node", "Network", "User", "Configuration", "Diagnostics", and "Assigned VMs". The "Links" section at the bottom left has a "Download Spice Client" link. The main content area is titled "Manage VMTemplates" and shows a table for "Vm Pool: vm-template-virtual-machine-pool-01". The table has columns: No., DisplayName, Status, Run Action, Memory, Node, and Action. It lists two entries: "ubuntu" (No. 1) and "Fedora" (No. 2), both in a "stopped" state. The "Action" column for each entry contains icons for "Start", "Stop", "Reset", and "Delete". Below the table is a pagination control with "Page 1 of 1" and "10" items per page, along with a "Refresh" button. At the bottom of the page, there is a footer note: "Version 1.3.1 on server sies Copyright © 2016 by FOSS-Group. All Rights Reserved." The taskbar at the bottom of the screen shows various application icons and the date/time "3:46 AM 9/21/2016".

No.	DisplayName	Status	Run Action	Memory	Node	Action
1	ubuntu	stopped	Start Stop Reset Delete		sies.sies.org	Start Stop Reset Delete
2	Fedora	stopped	Start Stop Reset Delete		sies.sies.org	Start Stop Reset Delete

Implement FOSS-Cloud Functionality – VDI (Virtual Desktop Infrastructure)

Today, Linux distributions like Ubuntu, Suse, Fedora etc. contains the spice protocol in their distribution. The client is named virt- or remote-viewer. FOSS-Cloud needs the remote viewer which is part of the virt-viewer package. In general the client is working out of the box.

The screenshot shows the FOSS-Cloud web-based management interface for virtual machine templates. The title bar reads "FOSS-Cloud - VmTemplate" and the address bar shows "192.168.10.10/vm-manager/vmTemplate/index.html". The main content area is titled "Manage VMTemplates" and displays a table of VM Templates. The table has columns: No., DisplayName, Status, Run Action, Memory, Node, and Action. There are two entries:

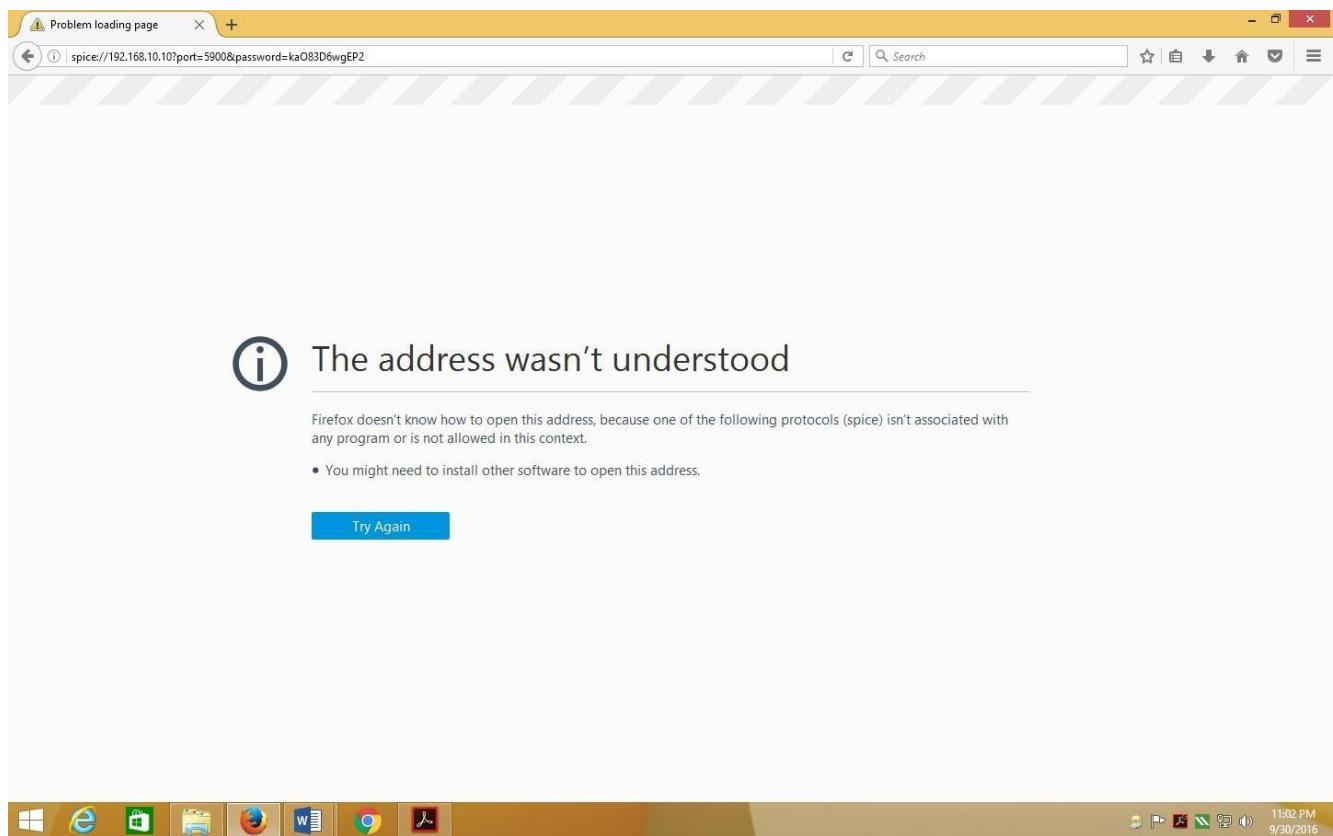
No.	DisplayName	Status	Run Action	Memory	Node	Action
1	Fedora	running	⟳ ⏪ ⏴ ⏵	2 GB / 2 GB	sies.sies.org	
2	ubuntu	stopped	⟳ ⏪ ⏴ ⏵	...	sies.sies.org	use VM Template

The left sidebar contains navigation links under "Virtual Machine" such as Persistent VMs, Dynamic VMs, VM Templates (which is selected), Create, Profiles, Create, and Upload ISO File. Other links include VM Pool, Storage Pool, Node, Network, User, Configuration, Diagnostics, Assigned VMs, and Links (Download Spice Client). The bottom right corner shows system icons and the date/time: 4:01 AM 9/22/2016.

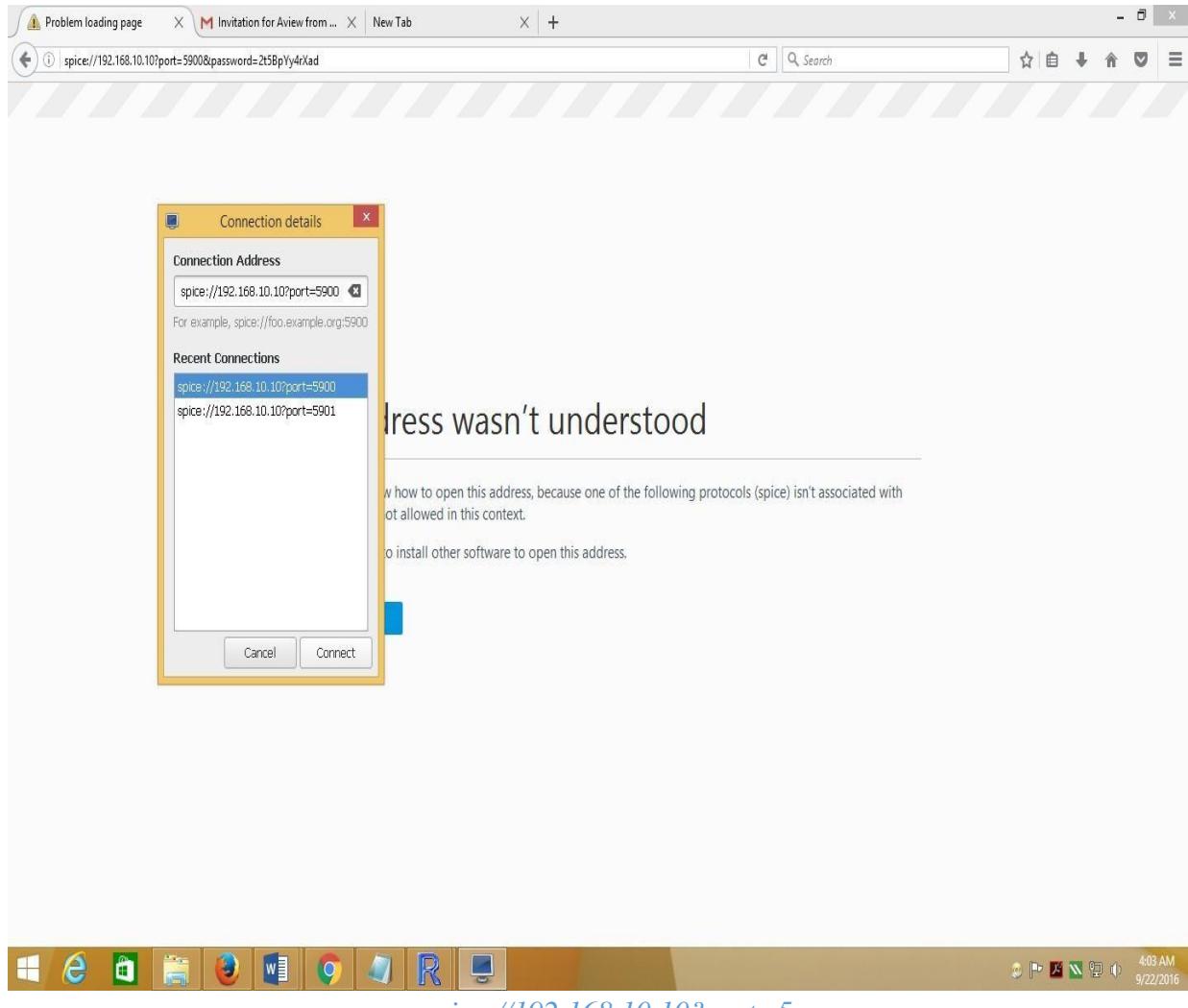
Cloud Computing - II

The Browser window opens up with following content in the address bar

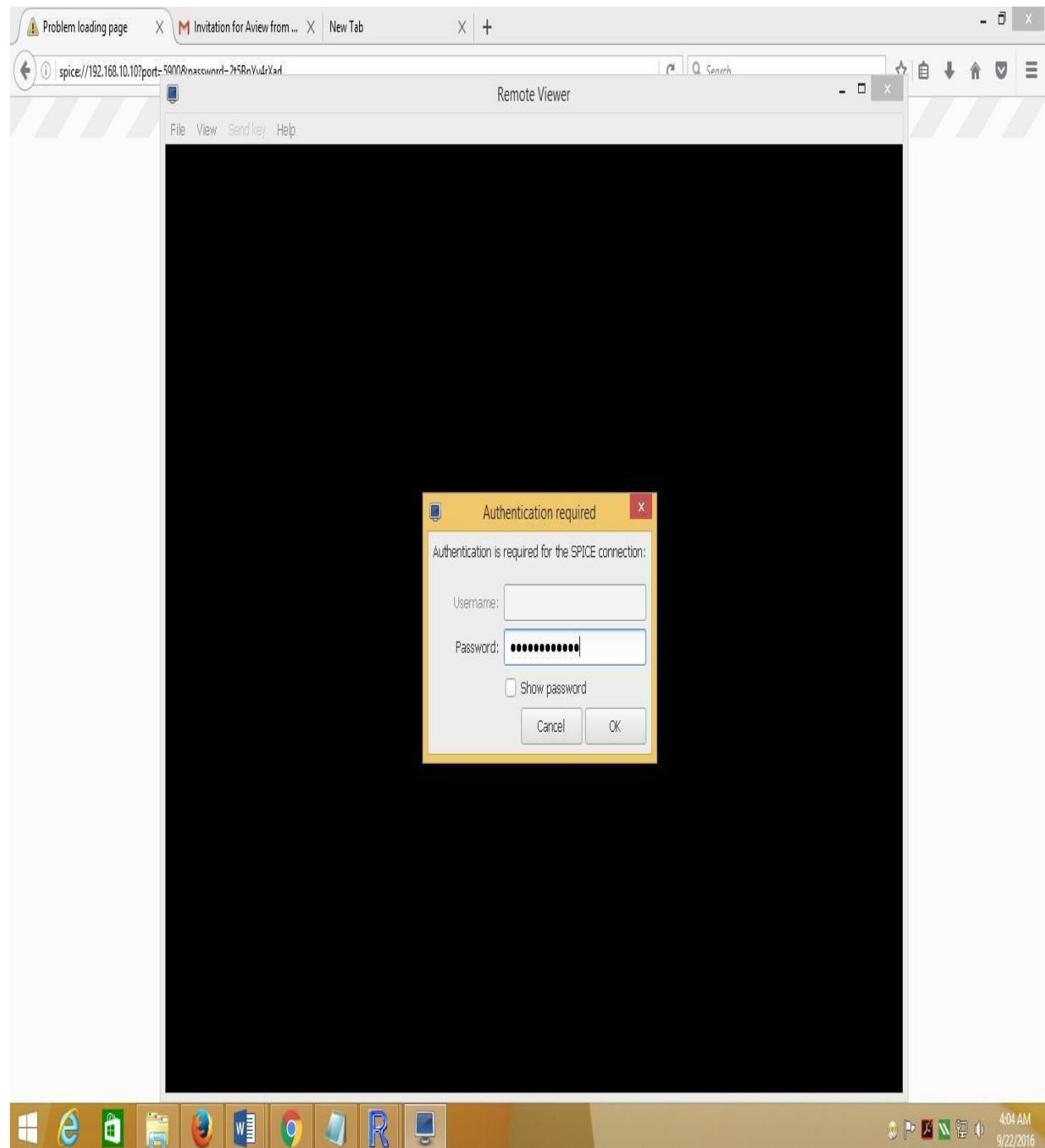
`spice://192.168.10.10?port=5900&password=kaO83D6wgEP2`



Open the Remote Viewer put the spice address in Connection Address as



Cloud Computing - II



Cloud Computing - II

The Virtual Desktop Interface opens up for that Virtual Machine as follows:

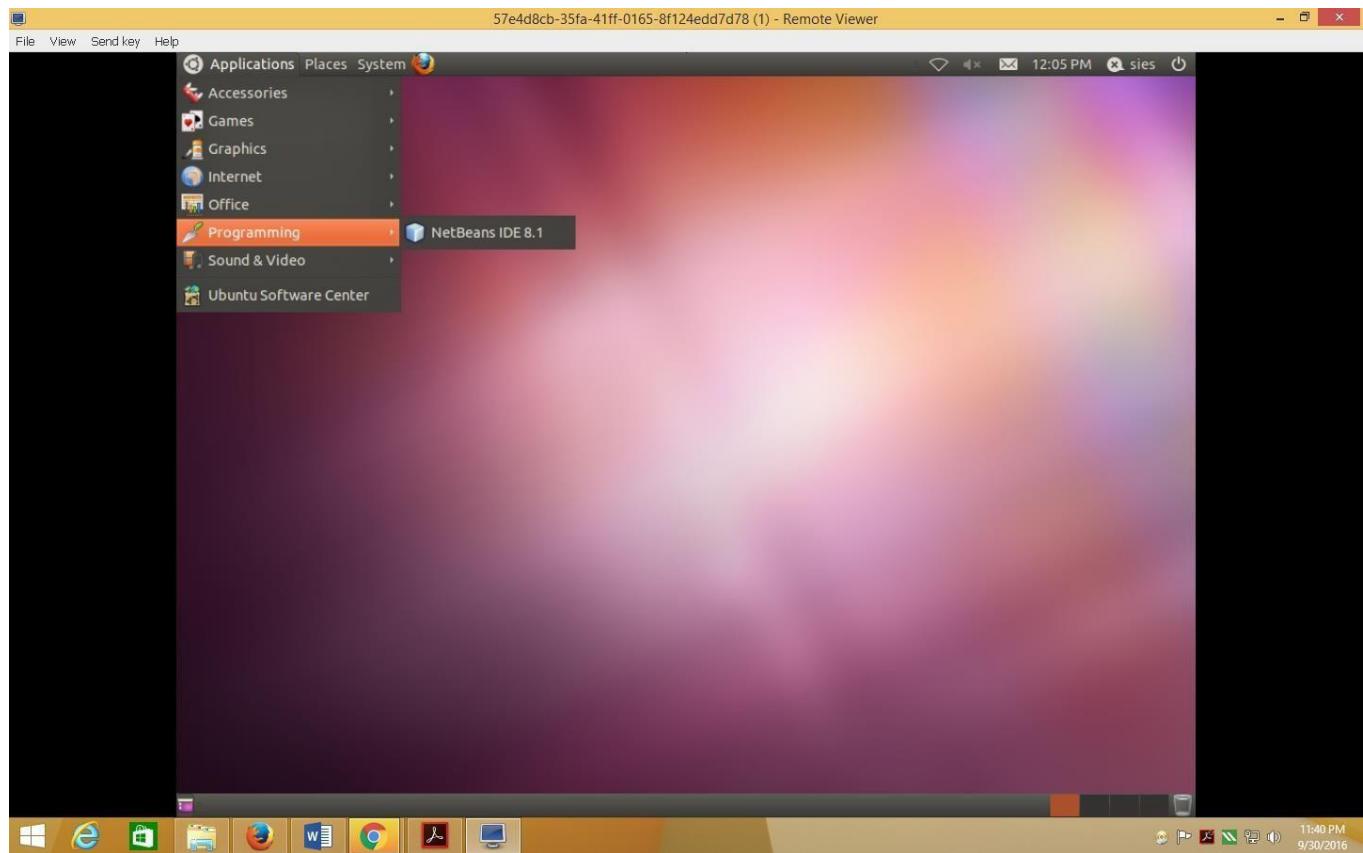


Practical No: 08

Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Platform as a Service (PaaS)

Software development Kits can be made available in the Virtual Machines that can be implemented as Platform as a Service.

Installation of Netbeans, Eclipse, Visual Studio and DBMS can be done in the appropriate Virtual Machines.



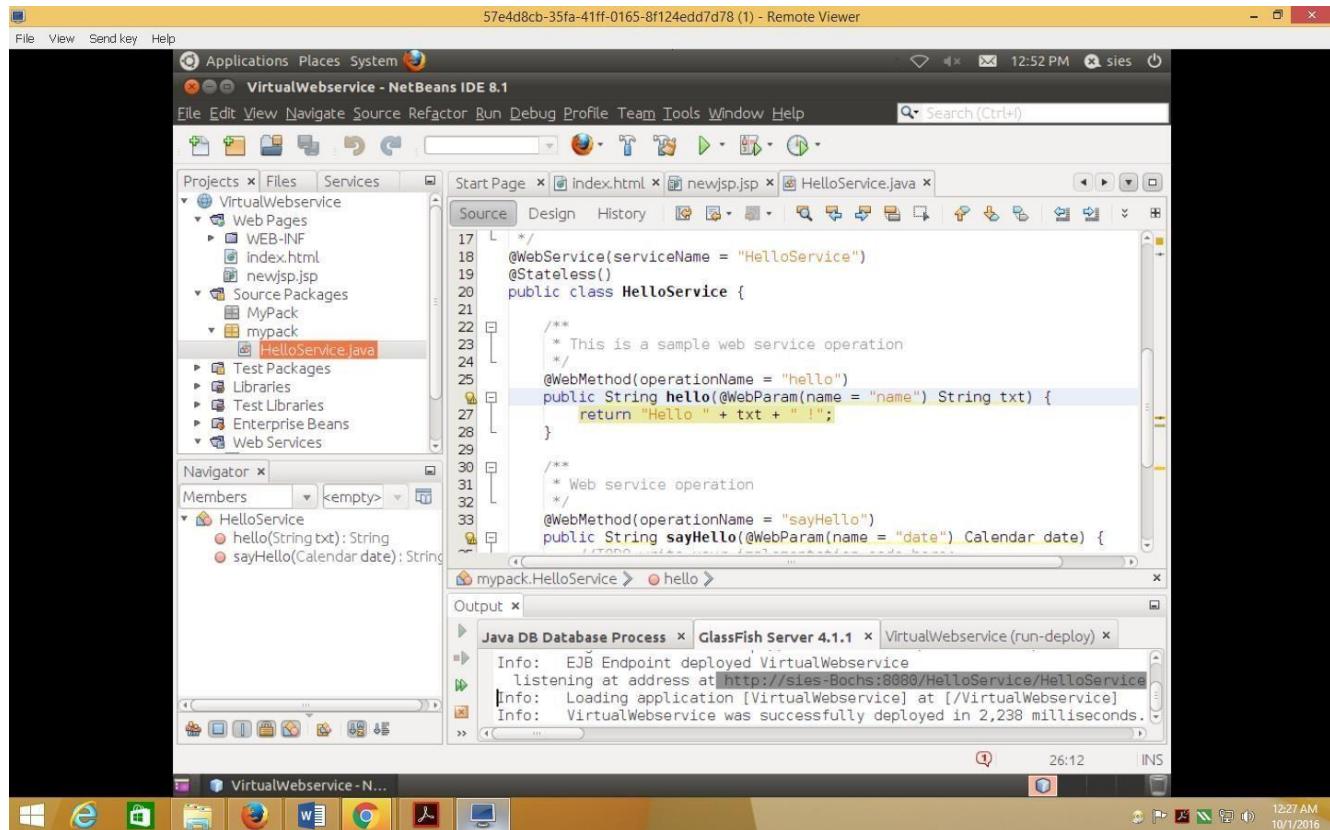
Practical No : 09

Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure)

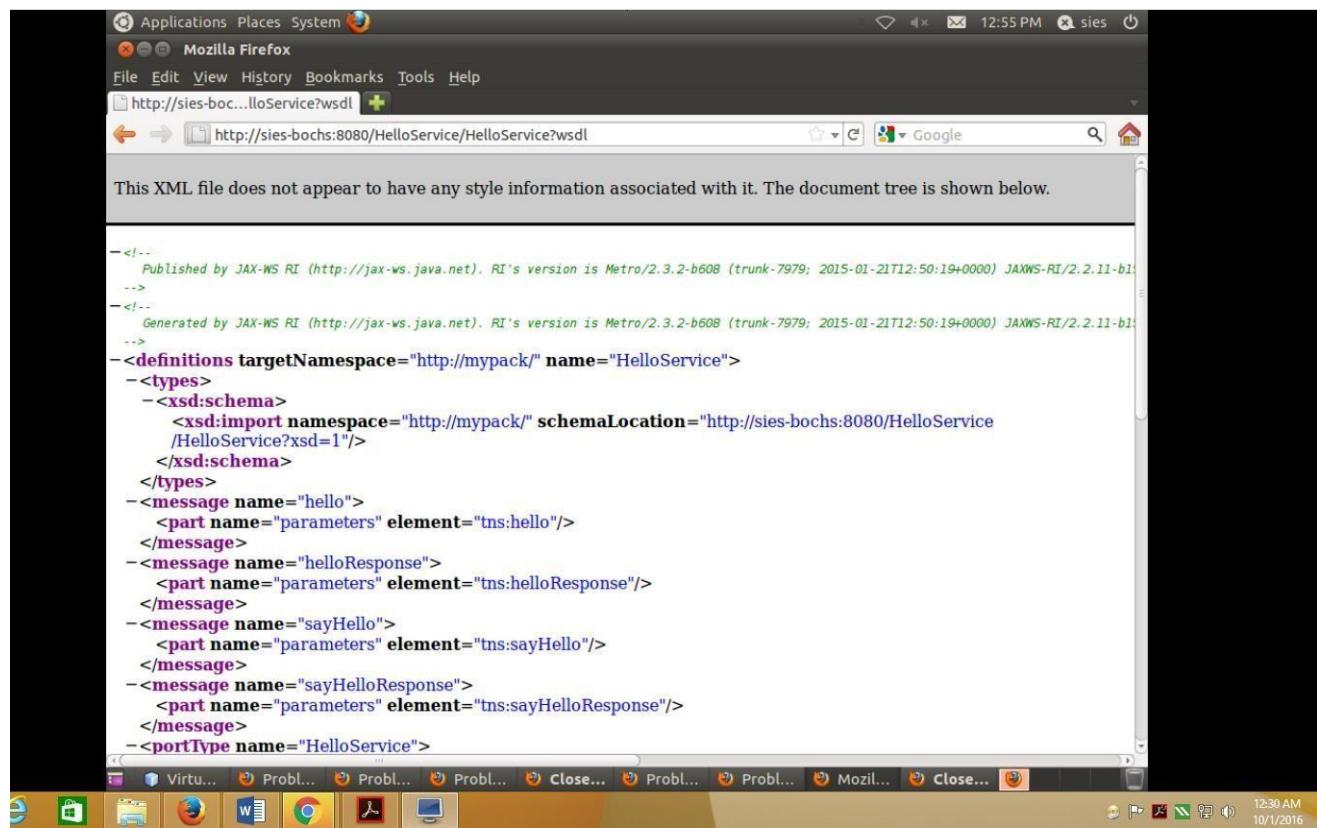
Software as a Service (SaaS)

Applications created and deployed in the virtual machines can be accessed by outside world.

This can be implemented as Software as Service.



Cloud Computing - II



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b1
-->
<!--
  Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b1
-->
<definitions targetNamespace="http://mypack/" name="HelloService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://mypack/" schemaLocation="http://sies-bochs:8080/HelloService
>HelloService?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="hello">
    <part name="parameters" element="tns:hello"/>
  </message>
  <message name="helloResponse">
    <part name="parameters" element="tns:helloResponse"/>
  </message>
  <message name="sayHello">
    <part name="parameters" element="tns:sayHello"/>
  </message>
  <message name="sayHelloResponse">
    <part name="parameters" element="tns:sayHelloResponse"/>
  </message>
  <portType name="HelloService">
```

Cloud Computing - II

Practical 10

Aim: Explore GCP / AWS / Azure

Theory:

What is DynamoDB?

DynamoDB is a hosted NoSQL database offered by Amazon Web Services (AWS). It offers:

reliable performance even as it scales;

a managed experience, so you won't be SSH-ing into servers to upgrade the crypto libraries;

a small, simple API allowing for simple key-value access as well as more advanced query patterns.

DynamoDB is a particularly good fit for the following use cases:

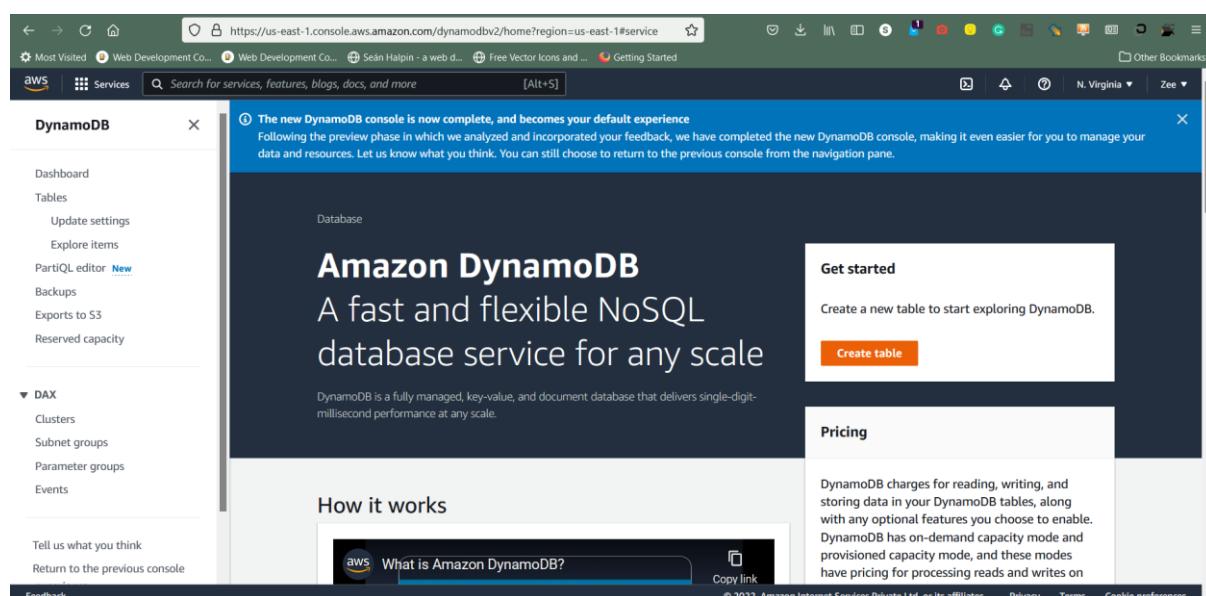
Applications with large amounts of data and strict latency requirements. As your amount of data scales, JOINs and advanced SQL operations can slow down your queries. With DynamoDB, your queries have predictable latency up to any size, including over 100 TBs!

Serverless applications using AWS Lambda. AWS Lambda provides auto-scaling, stateless, ephemeral compute in response to event triggers. DynamoDB is accessible via an HTTP API and performs authentication & authorization via IAM roles, making it a perfect fit for building Serverless applications.

Data sets with simple, known access patterns. If you're generating recommendations and serving them to users, DynamoDB's simple key-value access patterns make it a fast, reliable choice.

Steps:

1) Create table



Cloud Computing - II

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table details' step is active. A table named 'Zeehan-Practical-8' is being created with a partition key 'Roll-no' of type Number. A sort key is optional.

The screenshot shows the 'Create table' wizard continuing through configuration steps. It includes sections for Read/write capacity (using provisioned mode at 5 units each with auto scaling), Secondary indexes (none created), Key management (AWS owned key), Table class (Standard table class), and Tags (none added). Finally, it shows the 'Create table' button.

The screenshot shows the 'Tables' page in the AWS DynamoDB console. The table 'Zeehan-practical-8' is listed with a status of 'Creating'. It has a partition key 'Roll-no' and no sort key. The table uses provisioned capacity mode with 5 units each for reads and writes.

Cloud Computing - II

The screenshot shows the AWS DynamoDB console. On the left, the navigation bar includes 'Dashboard', 'Tables', 'Update settings' (which is highlighted in orange), 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', and 'Reserved capacity'. Below this is a section for 'DAX' with options for 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. At the bottom, there are links to 'Tell us what you think' and 'Return to the previous console experience'. The main content area is titled 'Zeeshan-Practical-8' and shows 'General information' for the table. It lists the partition key as 'Roll-no (Number)', sort key as '-', capacity mode as 'Provisioned', and table status as 'Active' with 'No active alarms'. Below this is an 'Items summary' table with columns for Item count (0), Table size (0 bytes), and Average item size (0 bytes). A button labeled 'Get live item count' is also present.

This screenshot shows the same AWS DynamoDB interface as above, but with the 'Actions' dropdown menu open. The menu includes options such as 'Edit capacity', 'Update table class', 'Delete table', 'Create item' (which is highlighted in blue), 'Create index', 'Create replica', 'Export to S3', 'Enable TTL', 'Manage tags', and 'Create access control policy'. The rest of the interface remains the same, showing the table's general information and items summary.

This screenshot shows the 'Scan/Query items' section of the AWS DynamoDB console. It features a 'Scan' or 'Query' selector, a 'Table or index' dropdown set to 'Zeeshan-Practical-8', and a 'Filters' section with a single filter entry: 'Attribute name' set to 'Enter attribute', 'Type' set to 'String', 'Condition' set to 'Equal to', and 'Value' set to 'Enter attribute'. Buttons for 'Run' and 'Reset' are at the bottom. The left sidebar and overall layout are identical to the previous screenshots.

2) Create Item

Cloud Computing - II

The screenshot shows the AWS DynamoDB console interface. A new item is being created in a table named 'Zeeshan-Practical-8'. The 'Roll-no' attribute is designated as the Partition Key with a value of '99'. There are two other attributes: 'Name' (String type) and 'Phone-No' (Number type). The 'Create item' button is visible at the bottom right.

Attribute name	Value	Type
Roll-no - Partition key	99	Number
Name	Zeeshan	String
Phone-No	6969696969	Number

3) Scan Items.

The screenshot shows the results of a scan operation. Three items were found, each with a Roll-no attribute and associated Name and Phone-No attributes.

	Roll-no	Name	Phone-No
<input type="checkbox"/>	99	Goliath	6969696969
<input type="checkbox"/>	1	John	
<input type="checkbox"/>	4	Zeeshan	

Cloud Computing - II

▼ Scan/Query items

Table or index
Zeeshan-Practical-8

▼ Filters

Attribute name	Type	Condition	Value	Remove
Roll-no	Number	Equal to	4	<input type="button" value="Remove"/>

⌚ Completed Read capacity units consumed: 0.5

Items returned (1)

	Roll-no	Name
<input type="checkbox"/>	4	Zeeshan

4) Query Items.

▼ Scan/Query items

Table or index
Zeeshan-Practical-8

Roll-no (Partition key)
99

▼ Filters

Add a filter to get started.

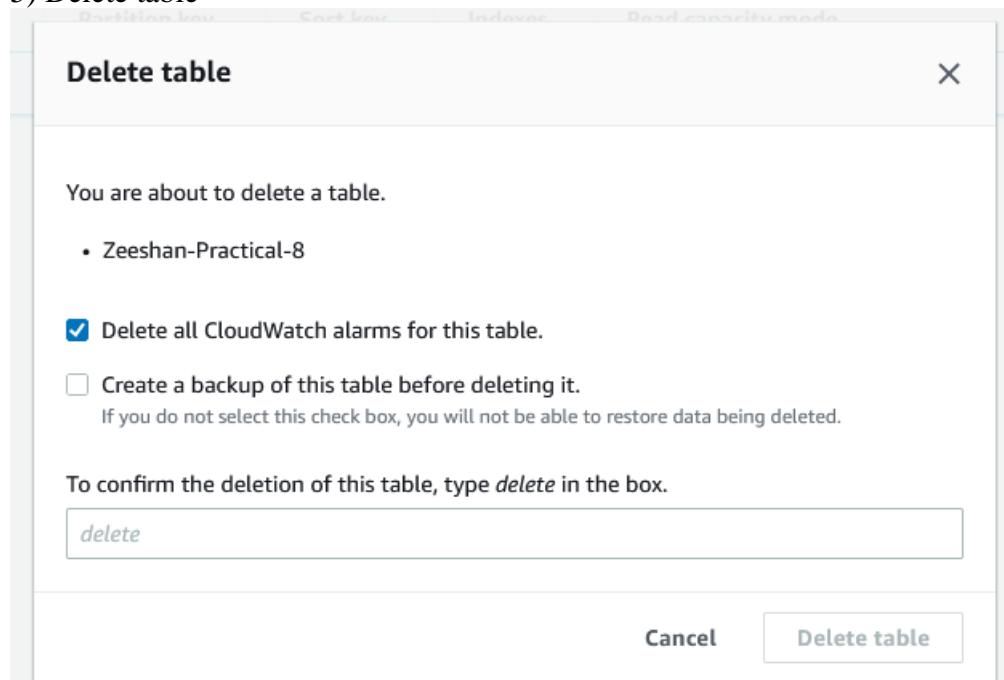
⌚ Completed Read capacity units consumed: 0.5

Items returned (1)

	Roll-no	Name	Phone-No
<input type="checkbox"/>	99	Goliath	6969696969

Cloud Computing - II

5) Delete table



The screenshot shows the AWS DynamoDB "Tables" page. The left sidebar has sections for "Dashboard", "Tables" (selected), "Partitions editor", "Backups", "Exports to S3", and "Reserved capacity". Under "Tables", there is a "Tables (1) Info" section with a table:

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Size	Table class
Zeeshan-Practical-8	Deleting	-	-	0	Provisioned (1)	Provisioned (1)	0 bytes	DynamoDB Standard

A green banner at the top of the main content area says: "The request to delete the 'Zeeshan-Practical-8' table has been submitted successfully."

Conclusion: Add, Update, Scan, Query and Delete table operation successfully implemented on DynamoDB.