

## Index

<b>Sr. No.</b>	<b>Practical Name</b>	<b>Page No</b>	<b>Date</b>
<b>1</b>	Execute & check the performance of existing algorithms using CloudSim.	3-17	09 <sup>th</sup> June 2022
<b>2</b>	Install a Cloud Analyst and Integrate with Eclipse/Netbeans. Monitor the performance of an Existing Algorithms.	18-28	16 <sup>th</sup> June 2022
<b>3</b>	Build an application on private cloud.	28-36	14 <sup>th</sup> July 2022
<b>4</b>	Demonstrate any Cloud Monitoring tool.	37-48	21 <sup>st</sup> July 2022
<b>5</b>	Evaluate a Private IAAS Cloud .	49-55	04 <sup>th</sup> Aug 2022
<b>6</b>	Implement FOSS-Cloud Functionality - VDI (Virtual Desktop Infrastructure)	56-61	11 <sup>th</sup> Aug 2022
<b>7</b>	Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS)	61-65	18 <sup>th</sup> Aug 2022
<b>8</b>	Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS)	65-67	25 <sup>th</sup> Aug 2022
<b>9</b>	Implement FOSS-Cloud Functionality - VSI Software as a Service (SaaS)	68-70	01 <sup>st</sup> Sep 2022
<b>10</b>	Explore GCP / AWS / Azure	71-72	15 <sup>th</sup> Sep 2022



## CERTIFICATE

This is here to certify that Mr. **Zeeshan Ali Sayyad**, Seat Number **051** M.Sc. II Computer Science, has satisfactorily completed the required number of experiments prescribed by the syllabus during the academic year 2022 – 2023.

Date:

Place: Mumbai

Teacher In-Charge

Head of Department

External Examiner

## **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.

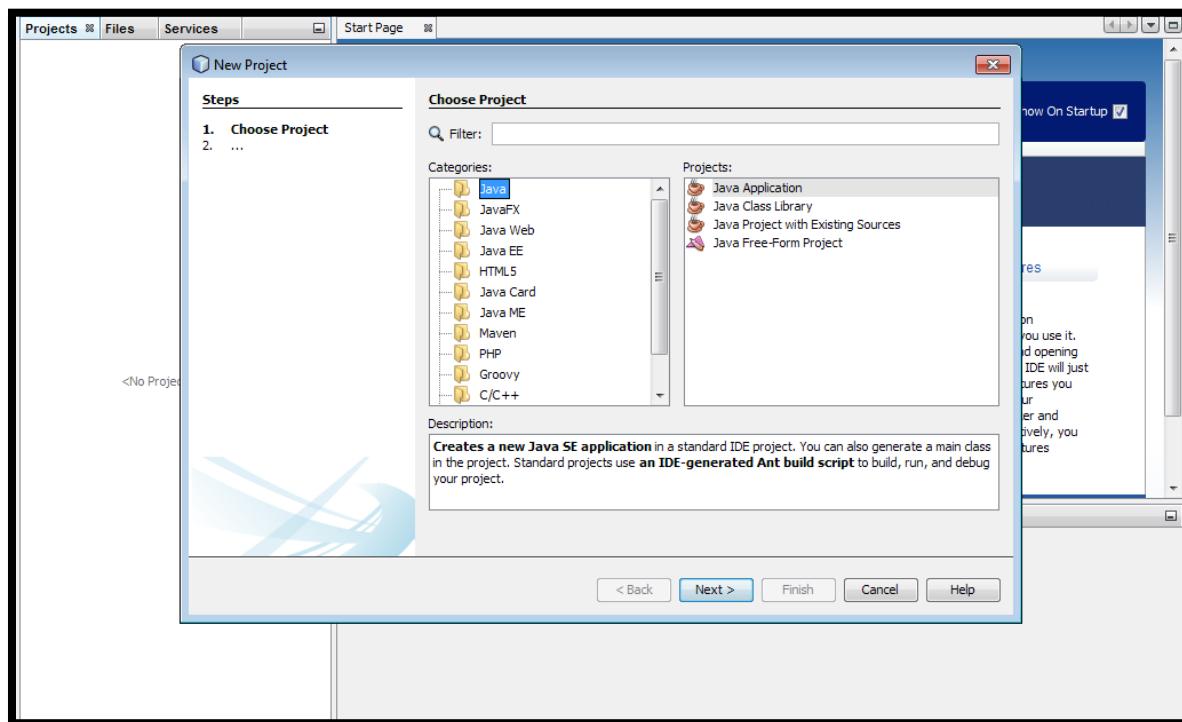
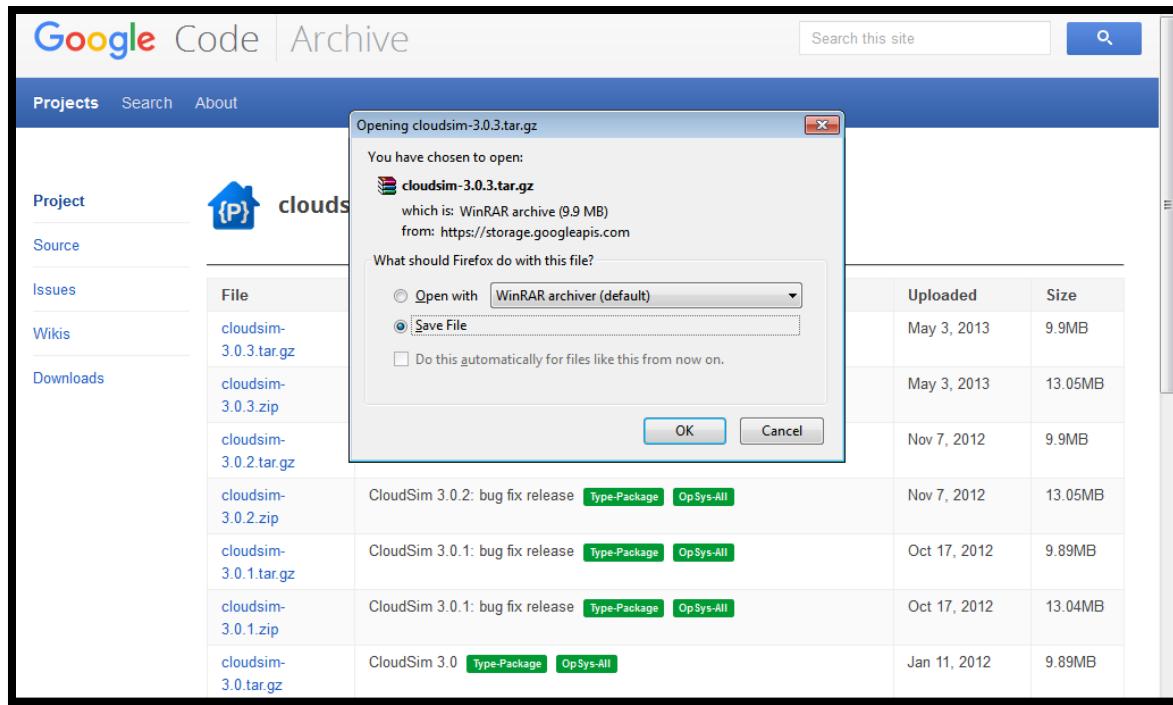
Download CloudSim:

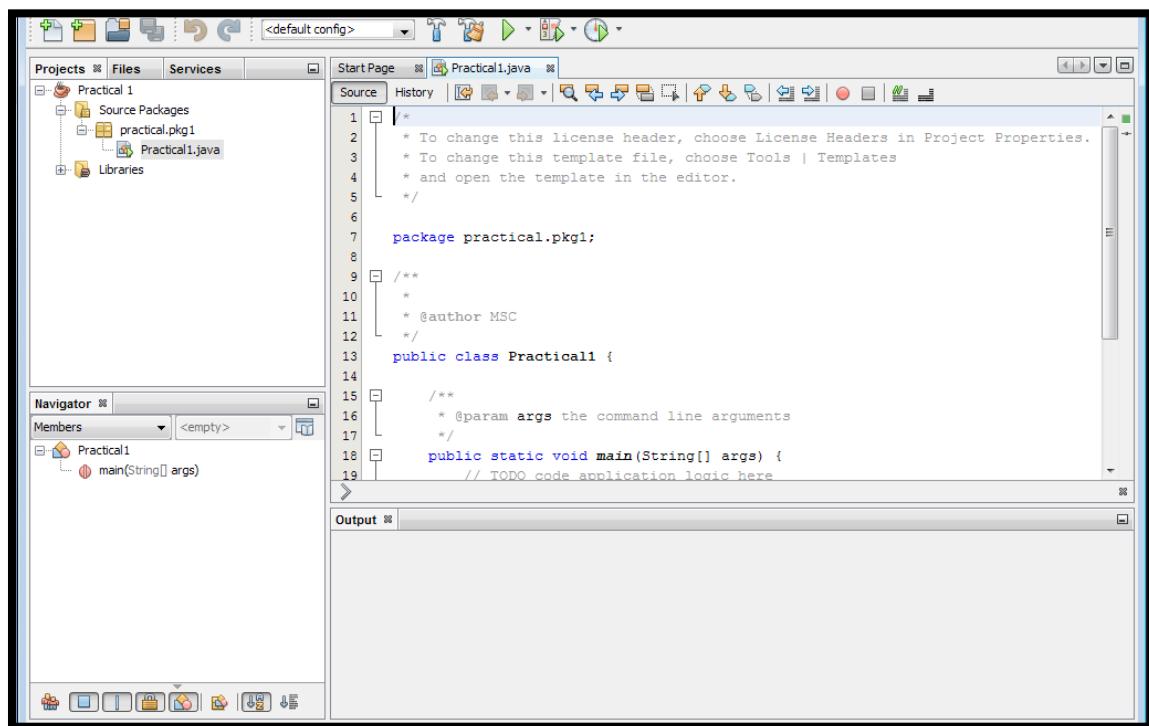
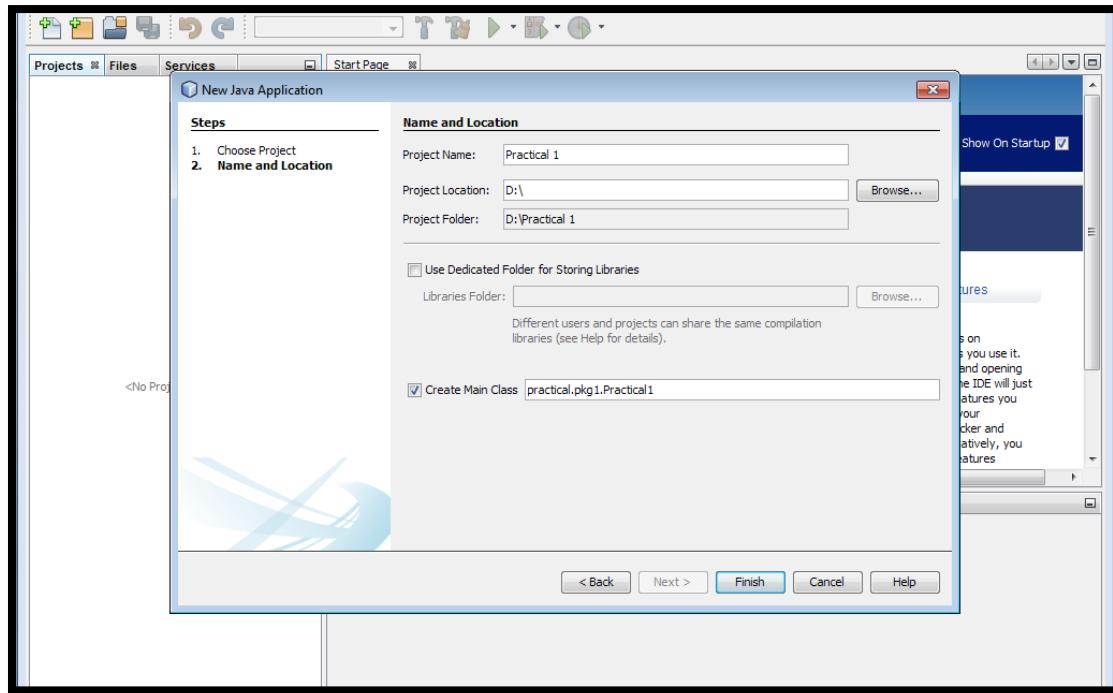
<http://code.google.com/p/cloudsim/downloads>

The screenshot shows the Google Code Archive interface for the 'cloudsim' project. The left sidebar has links for Project, Source, Issues, Wikis, and Downloads. The main area shows a table of download files. The first file, 'cloudsim-3.0.3.tar.gz', is highlighted with a red border around its row. Other files listed include 'cloudsim-3.0.3.zip', 'cloudsim-3.0.2.tar.gz', 'cloudsim-3.0.2.zip', 'cloudsim-3.0.1.tar.gz', 'cloudsim-3.0.1.zip', and 'cloudsim-3.0.tar.gz'. Each file entry includes a summary, upload date, and size.

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

Select 1<sup>st</sup> option



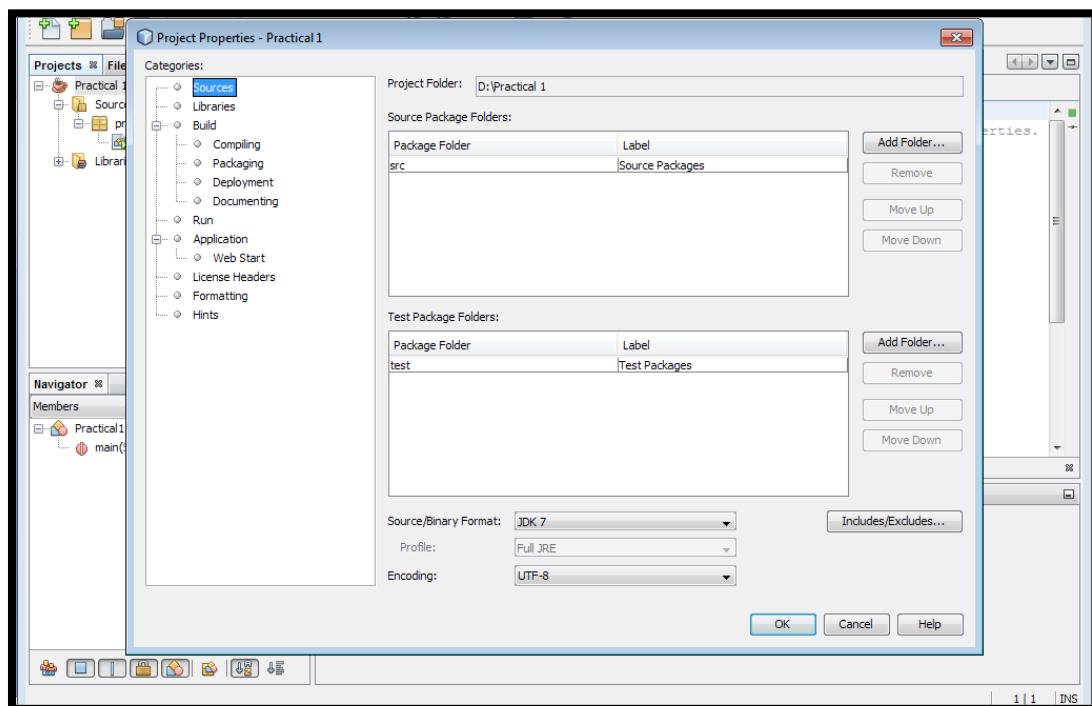


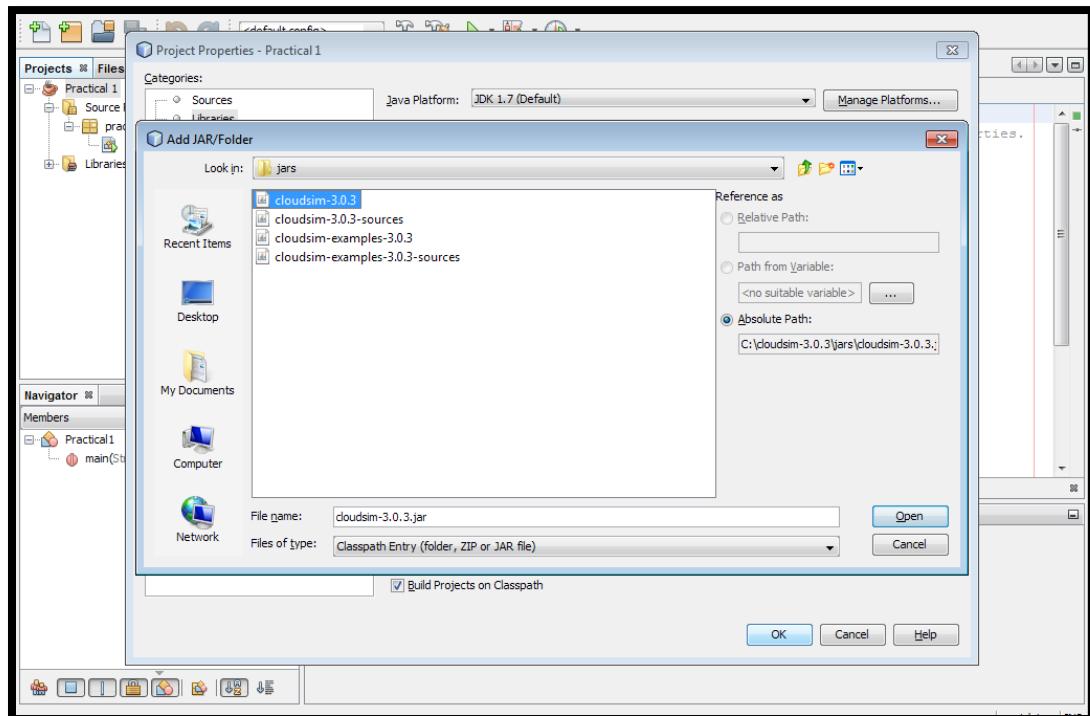
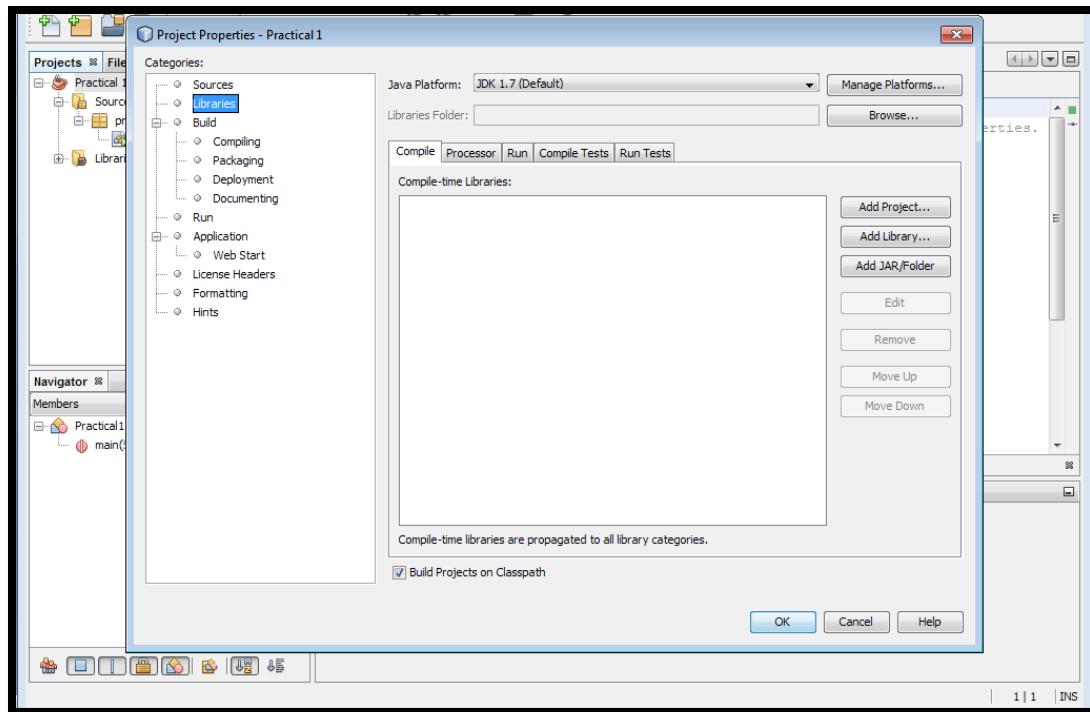
The screenshot shows the NetBeans IDE interface. The left sidebar displays the 'Projects' view with a single project named 'Practical1'. The main area is the code editor for 'Practical1.java', which contains the following Java code:

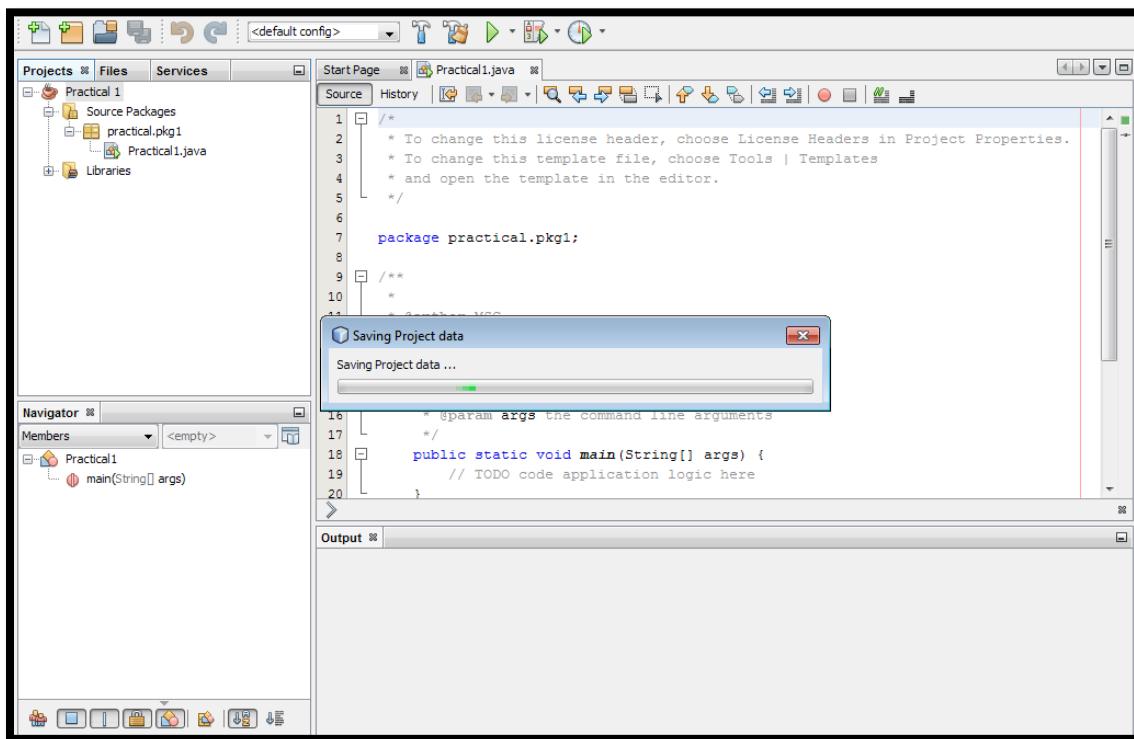
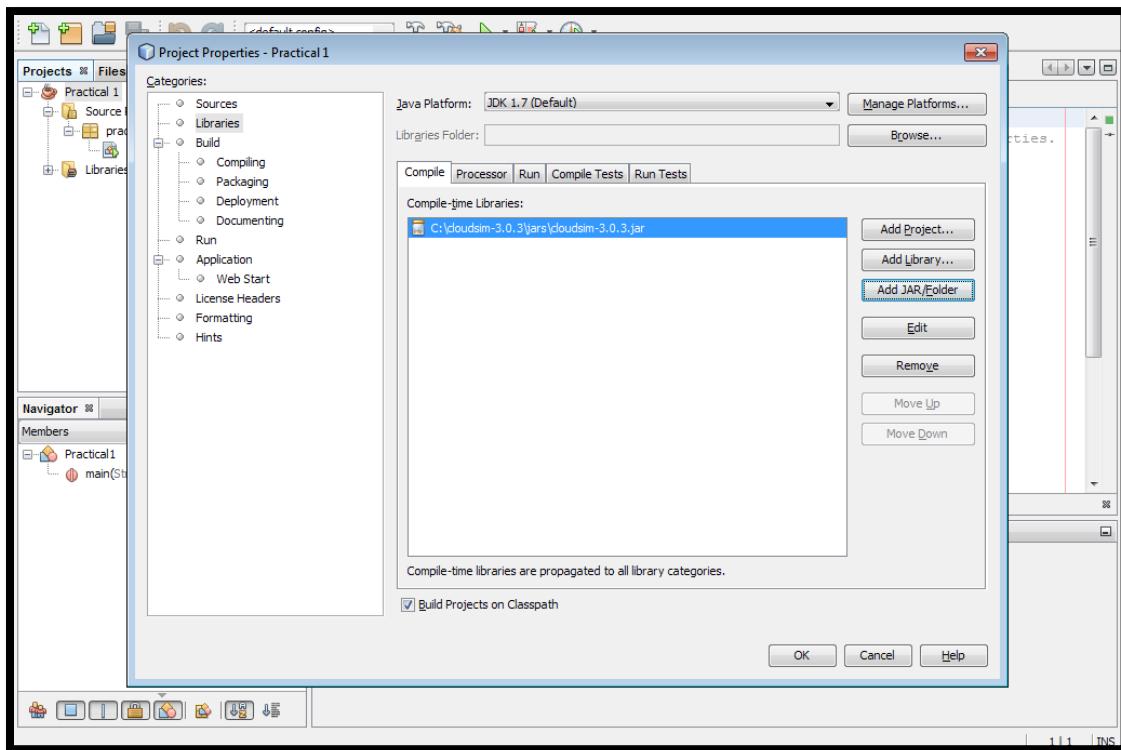
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package practical.pkg1;

/**
 * @author MSC
 */
public class Practical1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```







## 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;

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
            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;

long fileSize = 300;

long outputStreamSize = 300;

UtilizationModel utilizationModel = new UtilizationModelFull();

Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputStreamSize,
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.printLine("CloudSimExample1 finished!");

    } catch (Exception e) {
        e.printStackTrace();
        Log.printLine("Unwanted errors happen");
    }
}

/**
 * Creates the datacenter.
 *
 * @param name the name
 *
 * @return the datacenter
 */
private static Datacenter createDatacenter(String name) {

    // 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

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");
    } 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.printLine();

    Log.printLine("===== OUTPUT =====");

    Log.printLine("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.printLine(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()));

}

}

```

## Output:

```

Output - CloudSim1 (run) x
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

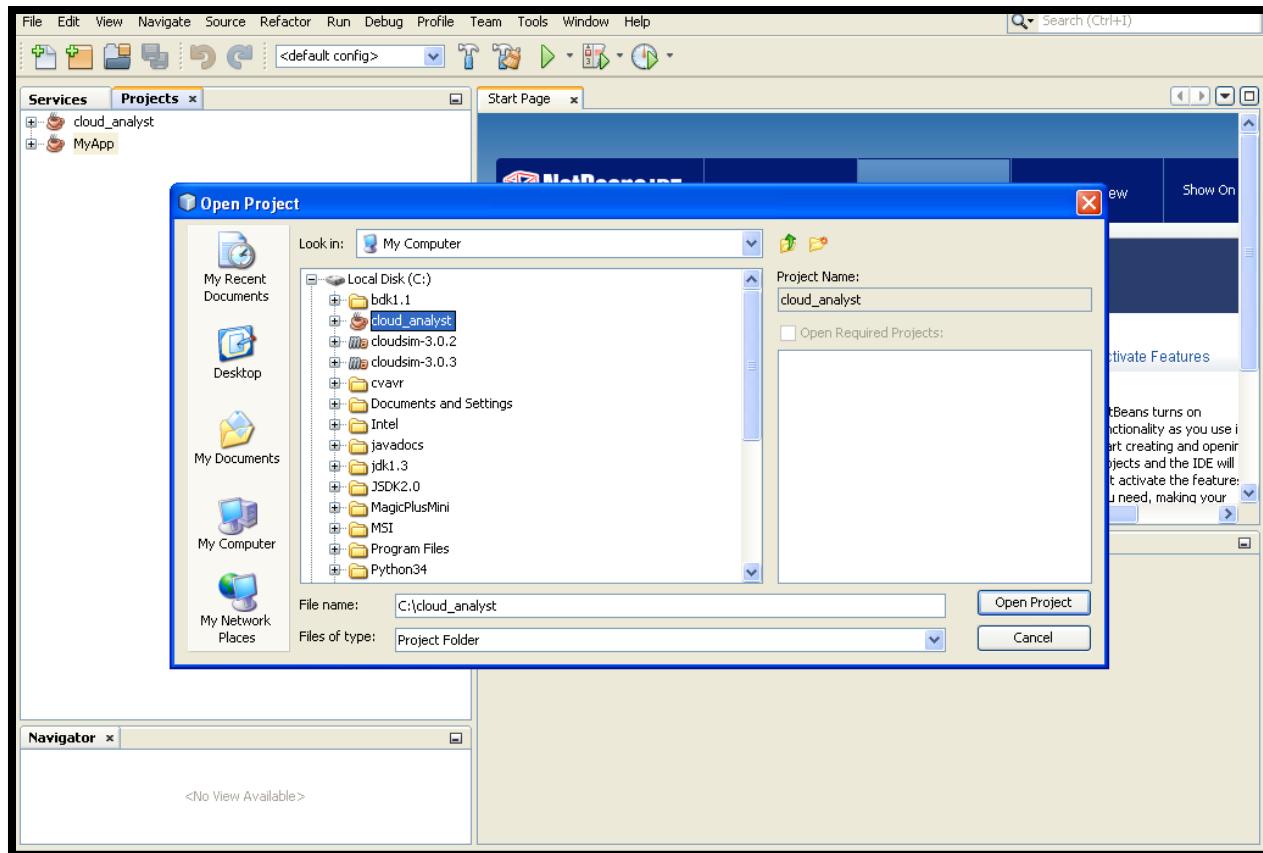
Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. In this tool, communities of users and data centers supporting the social networks are characterized and, based 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 drive .....then unzip it to use

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



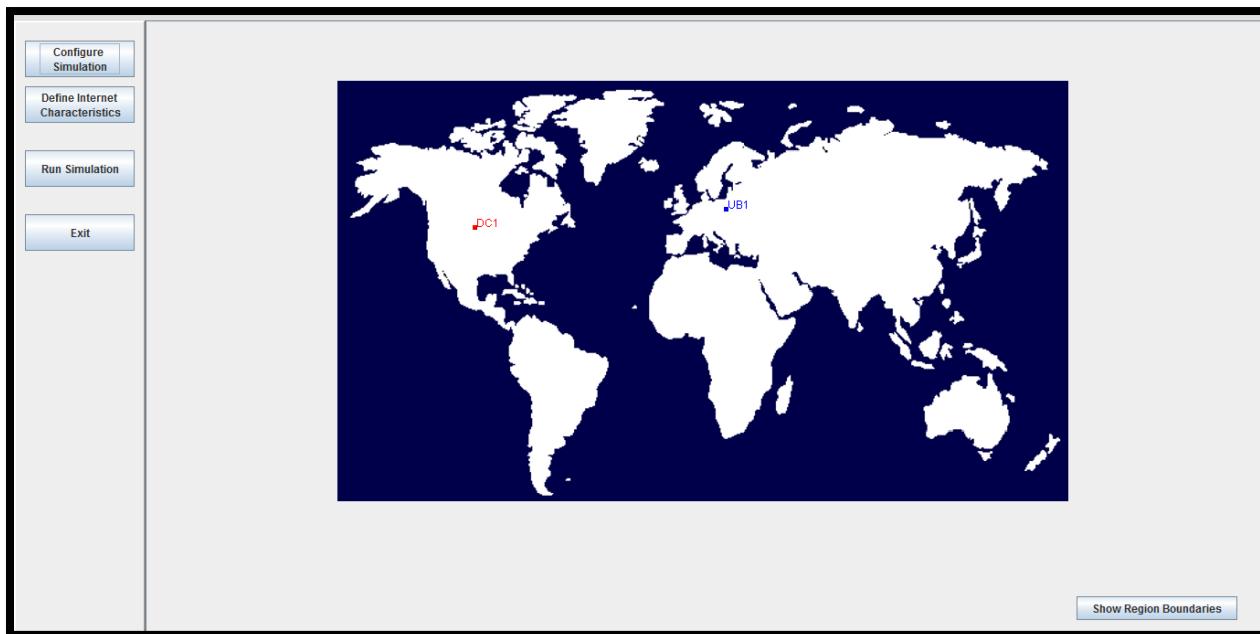
Expand source package folder inside which open cloudsim.ext.gui

right click on the GuiMain.java and click run

```

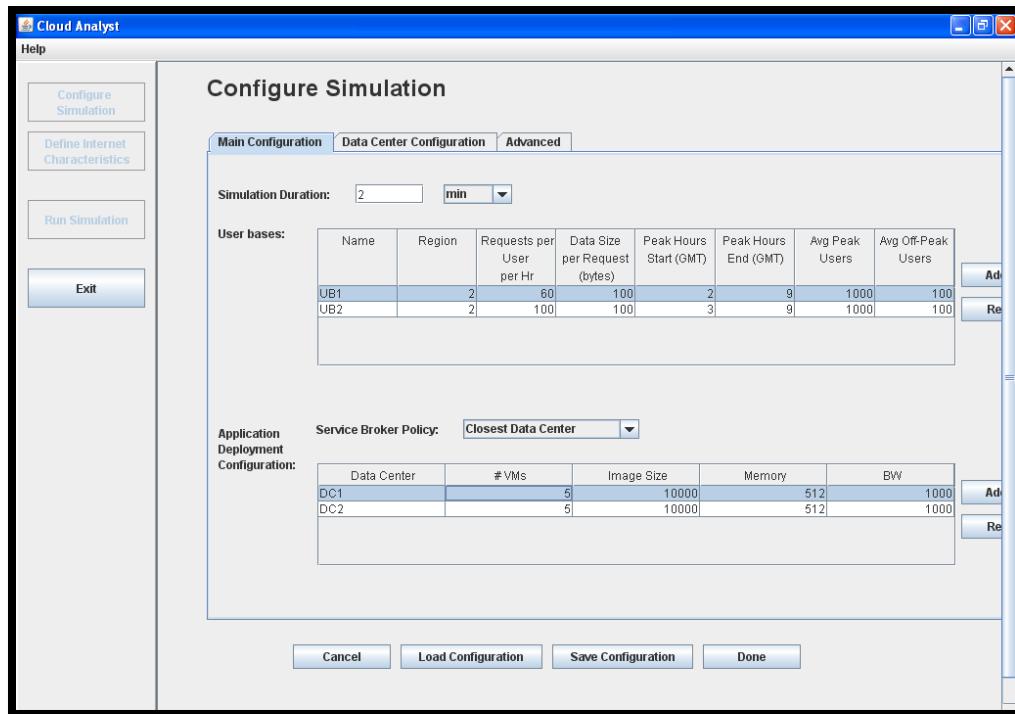
1 package cloudsim.ext.gui;
2
3 import java.awt.BorderLayout;
4 import java.awt.CardLayout;
5 import java.awt.Dimension;
6 import java.awt.Toolkit;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import java.util.HashMap;
10 import java.util.Map;
11
12 import javax.swing.Box;
13 import javax.swing.BoxLayout;
14 import javax.swing.JButton;
15 import javax.swing.JDialog;
16 import javax.swing.JFrame;
17 import javax.swing.JLabel;
18 import javax.swing.JMenuBar;
19 import javax.swing.JMenu;

```

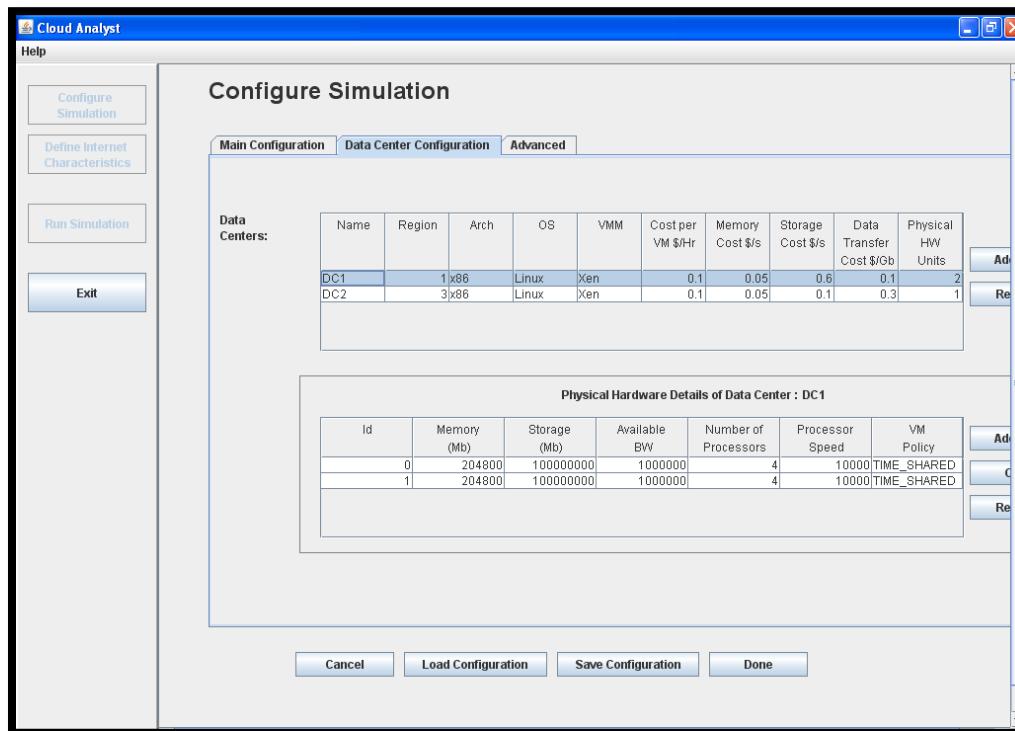


Now we are ready for checking algorithm performance

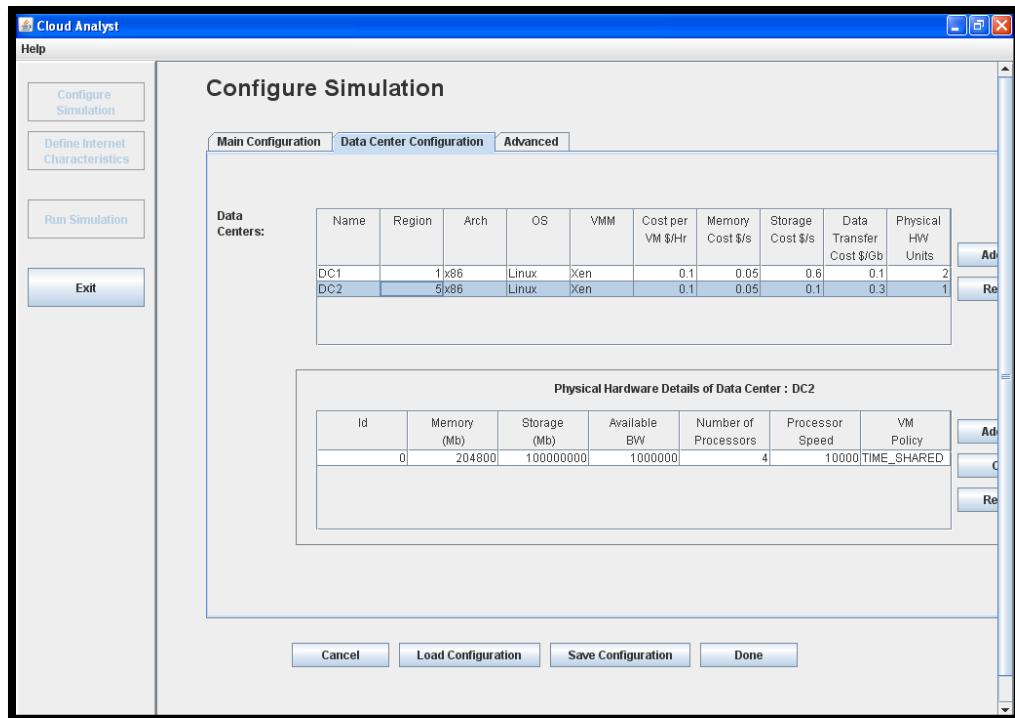
In Configure simulation window -> Main configuration tab set the values to



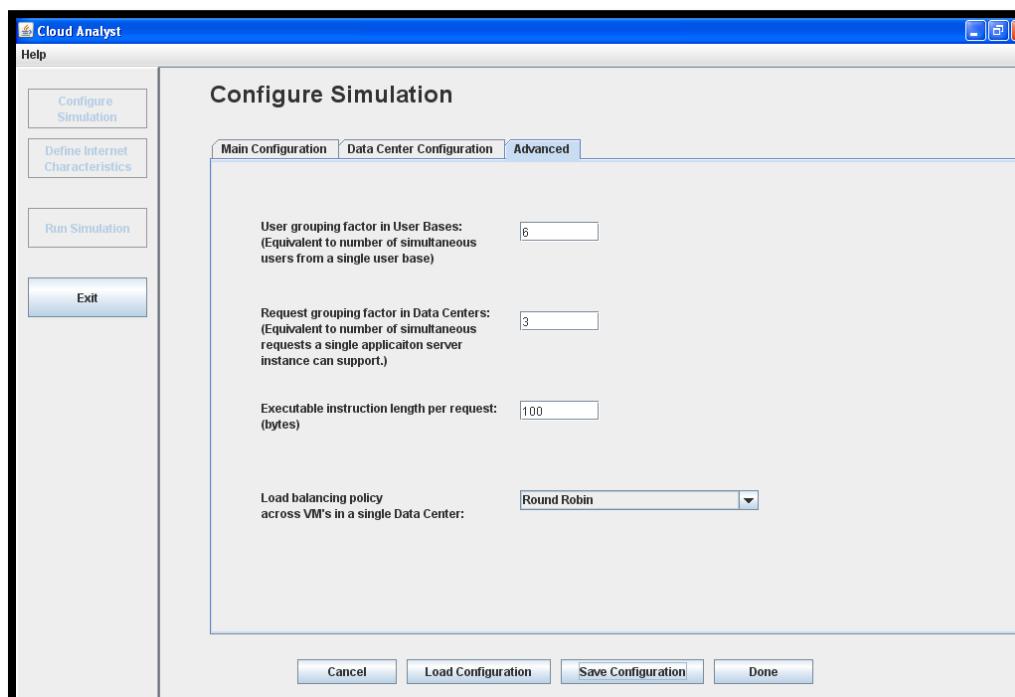
In data center configuration tab

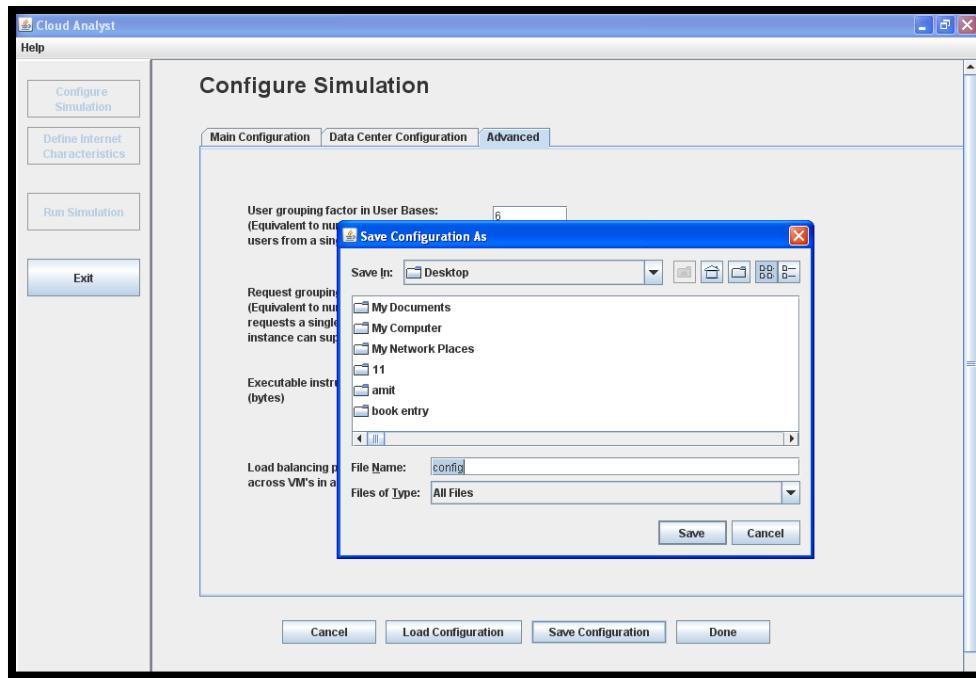


and

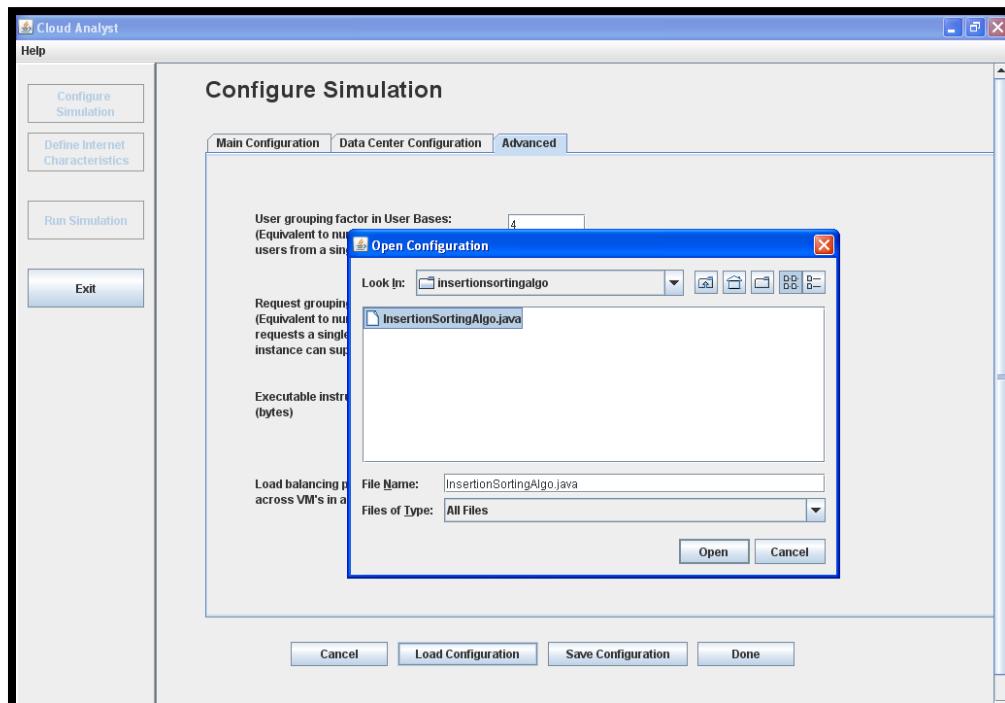


In advanced tab

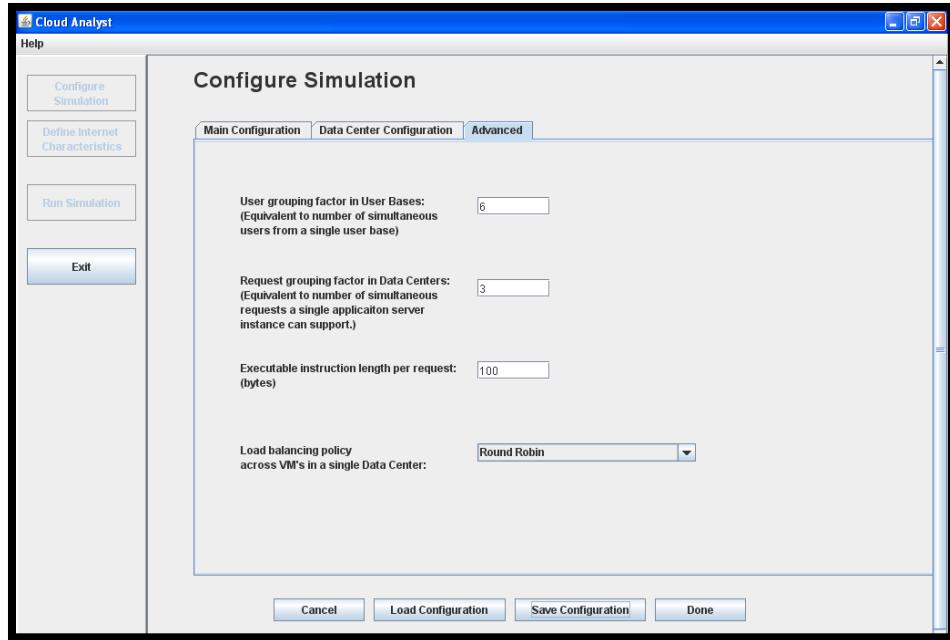




then click on load configuration button and select algorithm as

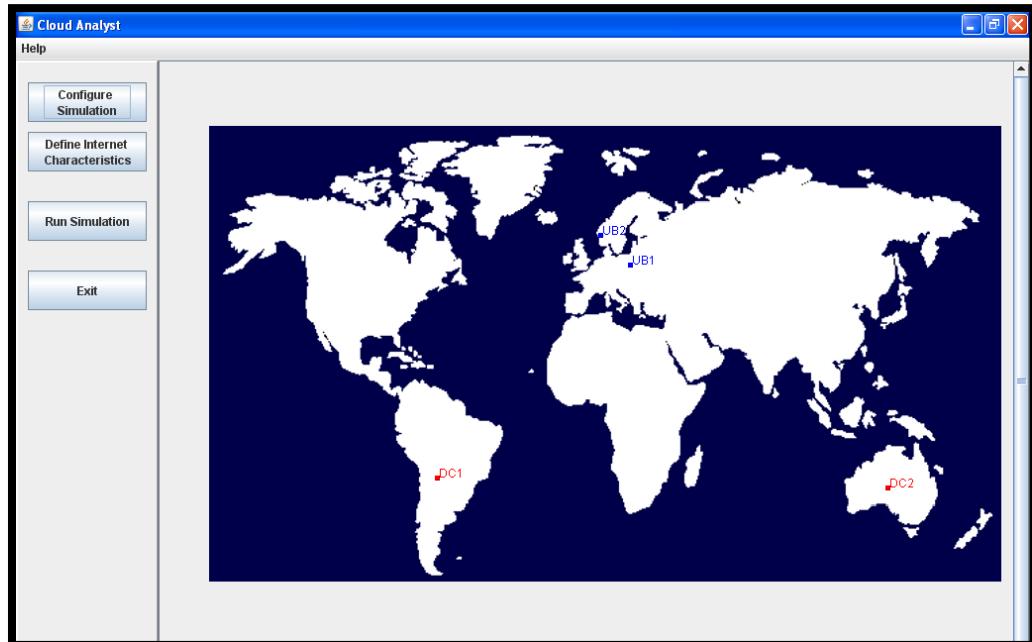


then save configuration as

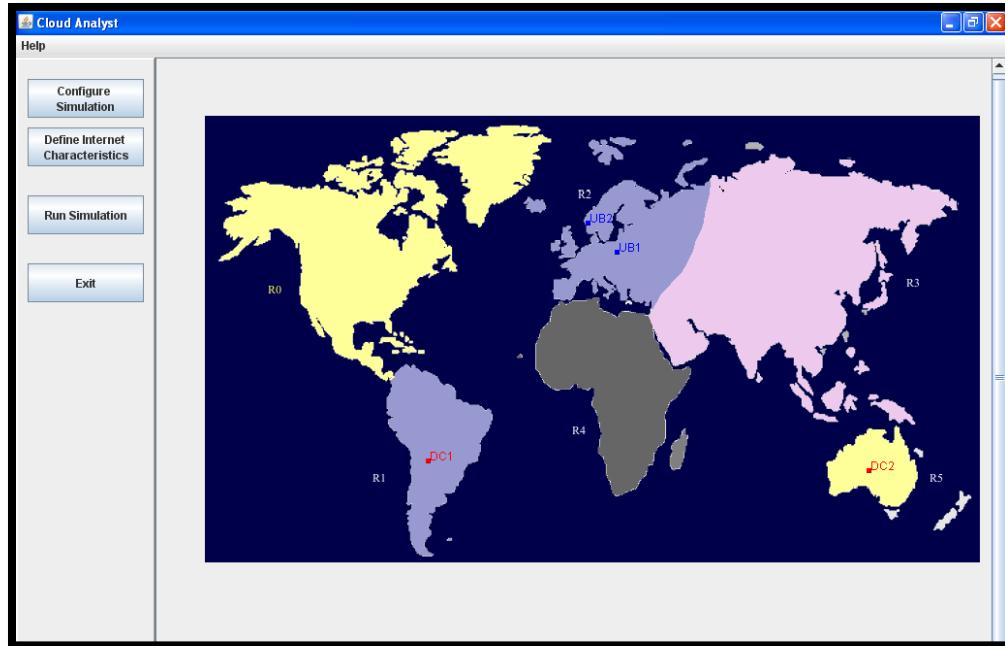


.sim file will saved to given location

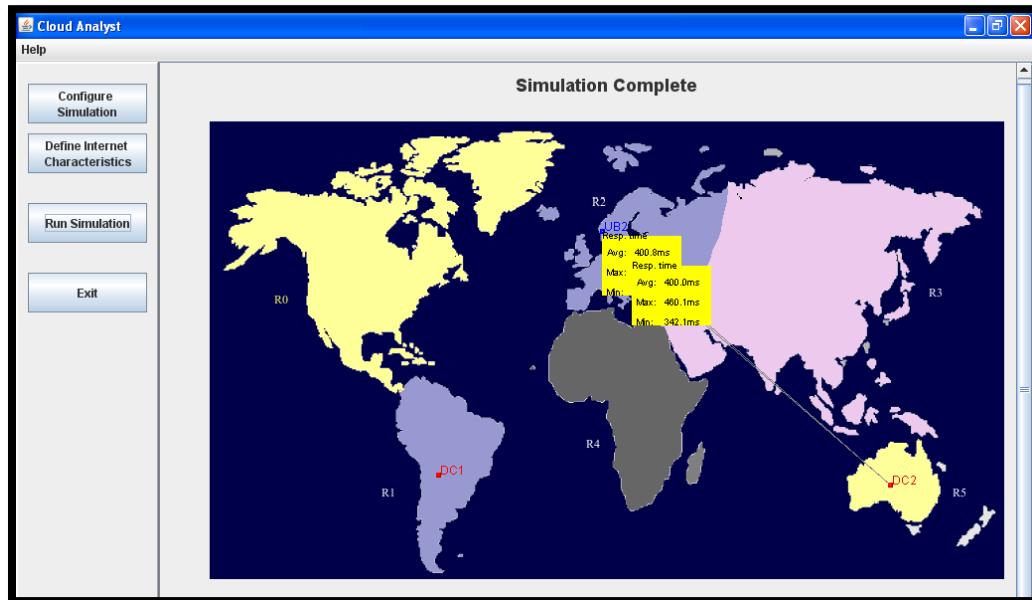
then click on done

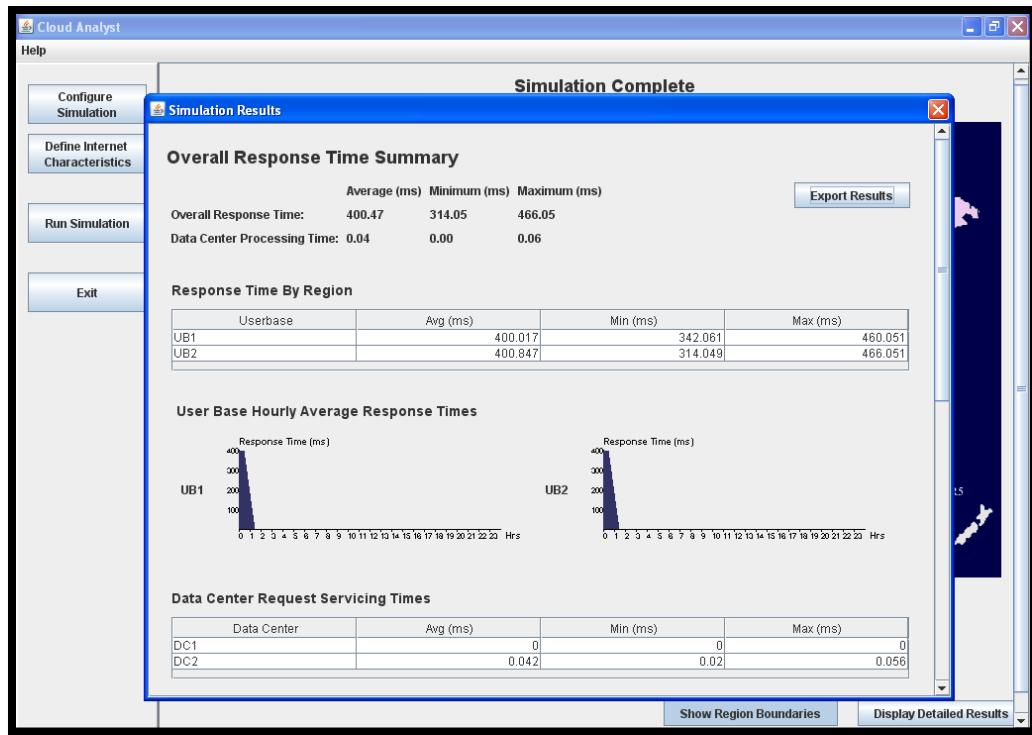


click on show region boundaries



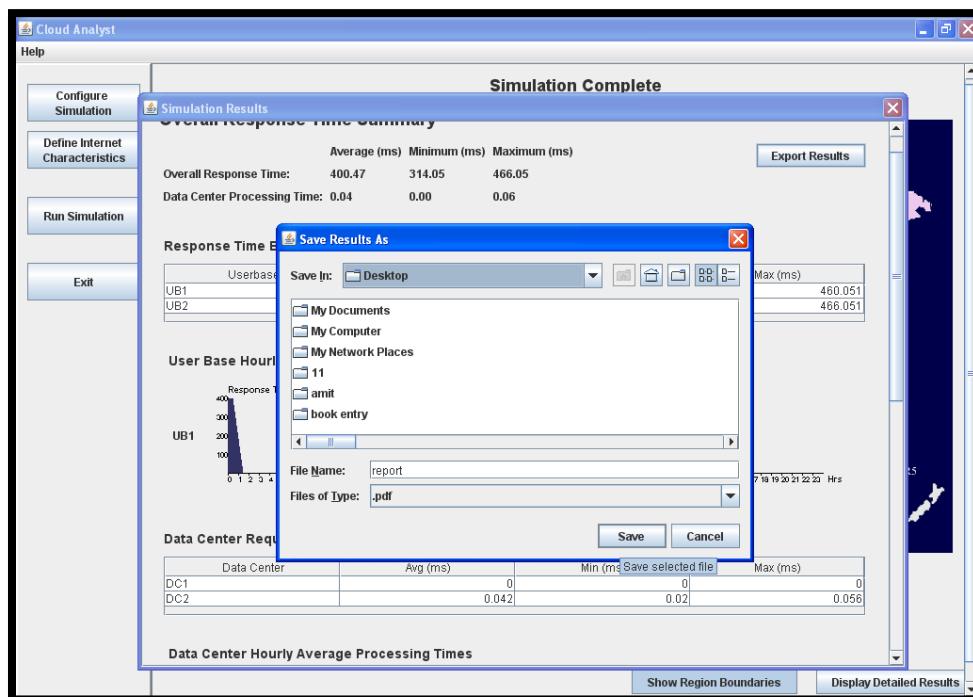
then click run simulation





then click on export result

pdf will be saved to given location



**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
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\*\*\*\*

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.

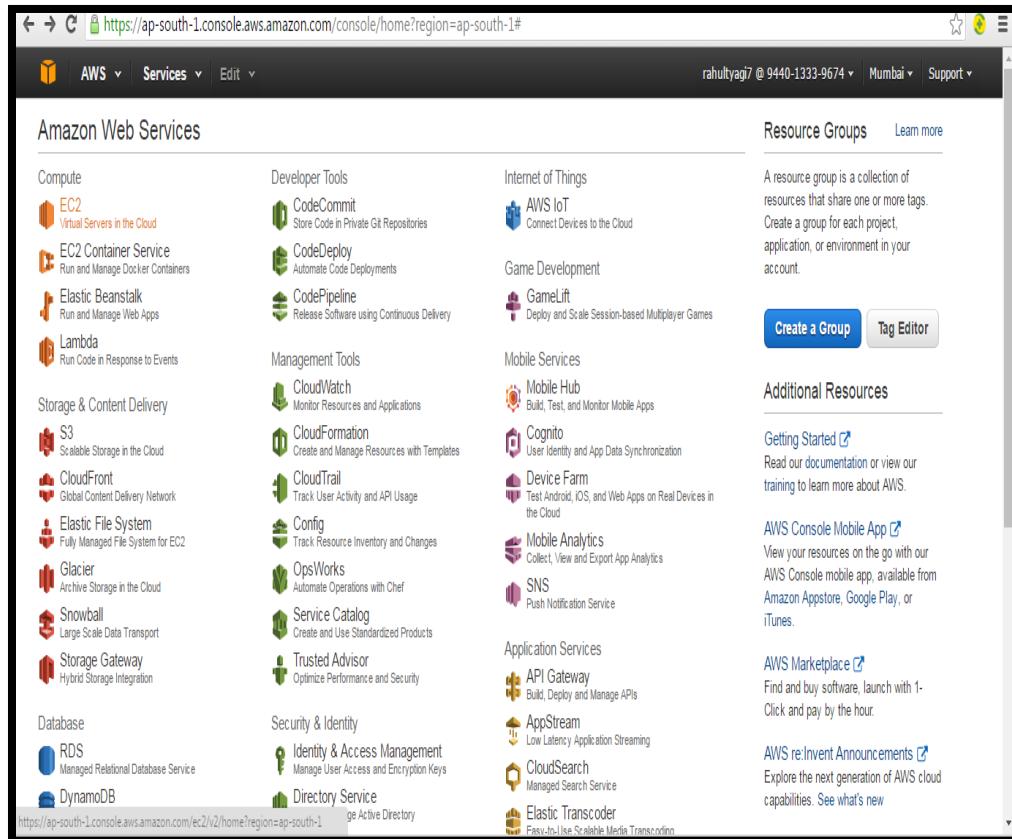
Note :

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



Click on EC2 and open it ,then select Launch INSTANCE

The screenshot shows the AWS EC2 Dashboard for the Asia Pacific (Mumbai) region. The left sidebar includes links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Dedicated Hosts, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, and Network & Security. The main content area displays the following statistics:

Category	Value
Running Instances	0
Dedicated Hosts	0
Volumes	0
Key Pairs	1
Placement Groups	0
Elastic IPs	0
Snapshots	0
Load Balancers	0
Security Groups	1

Below this, there's a callout for "Amazon Simple Workflow Service". A "Create Instance" section contains a "Launch Instance" button. The "Service Health" section shows "Service Status: Asia Pacific (Mumbai)" and "No events". The "Scheduled Events" section shows "No events". The right side of the dashboard lists "Account Attributes" such as Supported Platforms (VPC), Default VPC (vpc-7ada3913), and Resource ID length management. It also includes links for Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us. The "AWS Marketplace" section offers free software trial products and popular AMIs.

## Select Linux AMI

The screenshot shows the "Step 1: Choose an Amazon Machine Image (AMI)" step of the instance creation wizard. The top navigation bar includes links for AWS, Services, Edit, and account information. Below the navigation, a progress bar shows steps 1 through 7. The main content area is titled "Quick Start" and lists three AMI options:

Image	Name	Description	Action
Amazon Linux	Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type - ami-fbdd790	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 <small>Free tier eligible</small>	Select
Red Hat	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 <small>Free tier eligible</small>	Select
SUSE	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. Root device type: ebs Virtualization type: hvm <small>Free tier eligible</small>	Select

On the right, there are navigation arrows for "1 to 22 of 22 AMIs".

## Choose the instance type

The screenshot shows the AWS EC2 instance creation wizard at Step 2: Choose an Instance Type. The page title is "Step 2: Choose an Instance Type". A sub-header states: "Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs." Below this is a table with columns: Family, Type, vCPUs, Memory (GiB), Instance Storage (GiB), EBS-Optimized Available, and Network Performance. The table lists several t2 instances: t2.nano, t2.micro (marked as "Free tier eligible"), t2.small, t2.medium, and t2.large. The "t2.micro" row is highlighted. At the bottom are buttons for "Cancel", "Previous", "Review and Launch" (which is blue and bold), and "Next: Configure Instance Details".

## Configure the Instance

The screenshot shows the AWS EC2 instance creation wizard at Step 3: Configure Instance Details. The page title is "Step 3: Configure Instance Details". A sub-header states: "Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more." The configuration options include: Number of instances (set to 1), Purchasing option (checkbox for Request Spot instances), Network (vpc-7ada3913 (172.31.0.0/16) | DEFAULT-VPC), Subnet (No preference (default subnet in any Availability Zone)), Auto-assign Public IP (Use subnet setting (Enable)), IAM role (None), Shutdown behavior (Stop), Enable termination protection (checkbox for Protect against accidental termination), and Monitoring (checkbox for Enable CloudWatch detailed monitoring). At the bottom are buttons for "Cancel", "Previous", "Review and Launch" (blue and bold), and "Next: Add Storage".

## 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

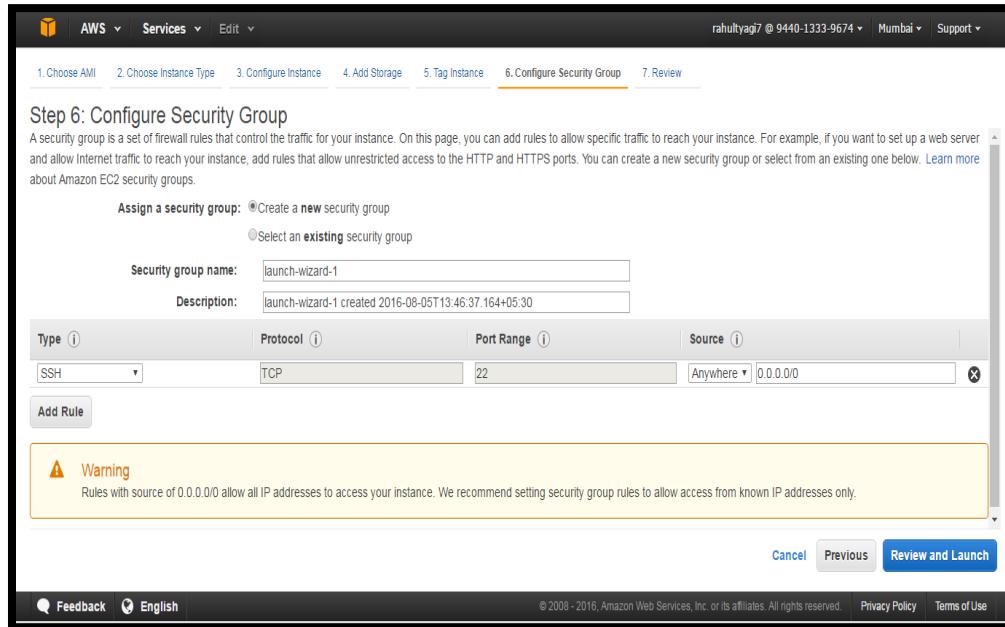
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

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

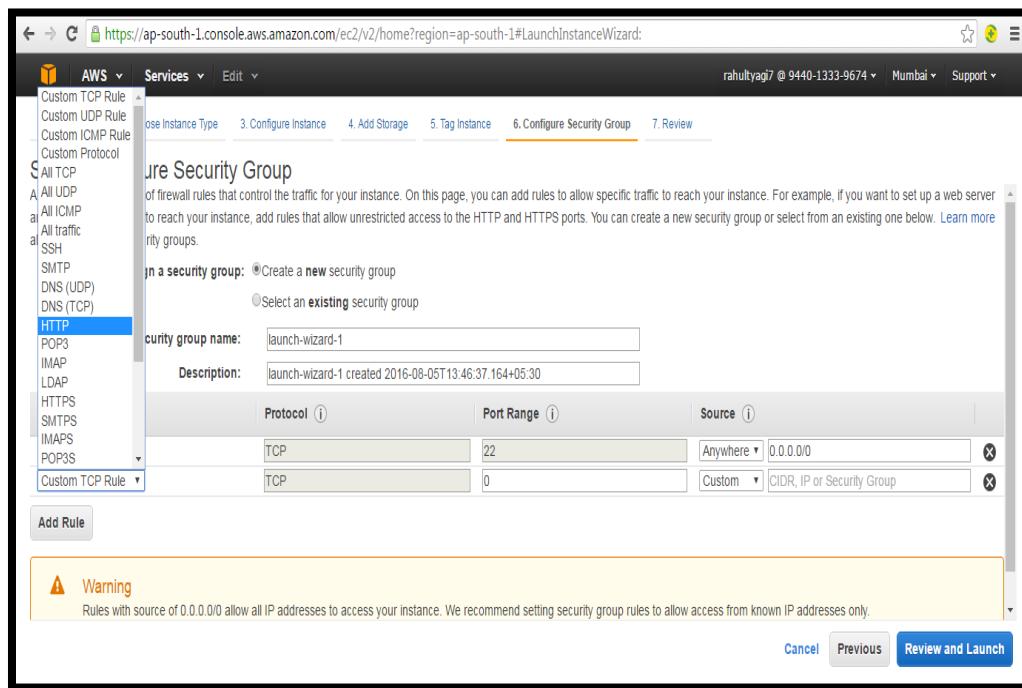
Create Tag (Up to 10 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

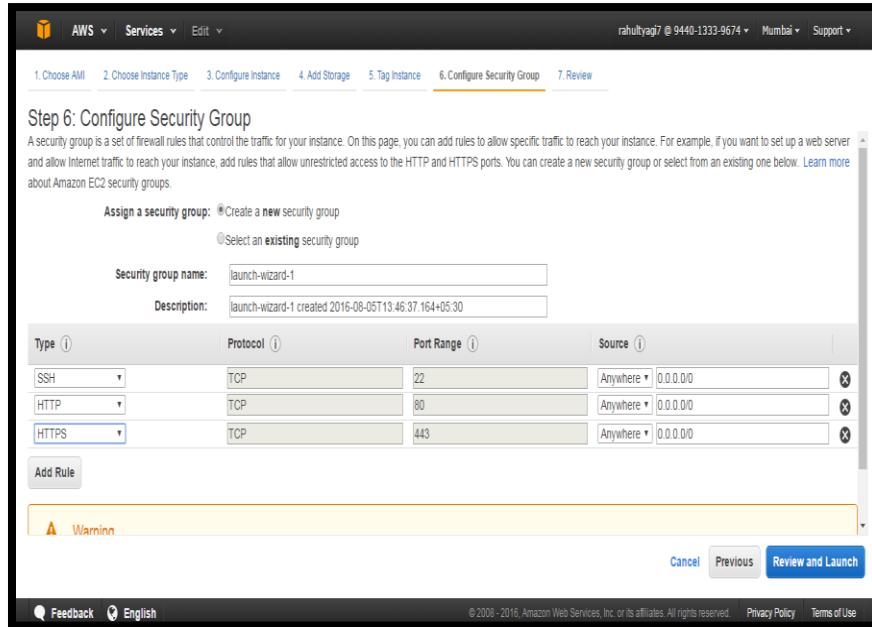
## Configure the security group by adding some rule..



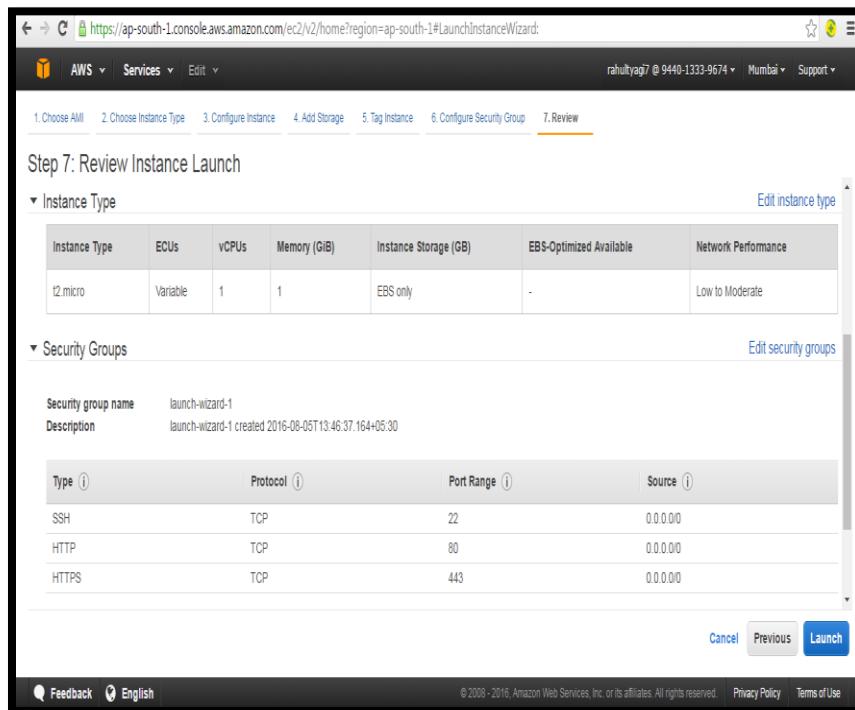
Add the first rule Http.



## Add the second rule Https

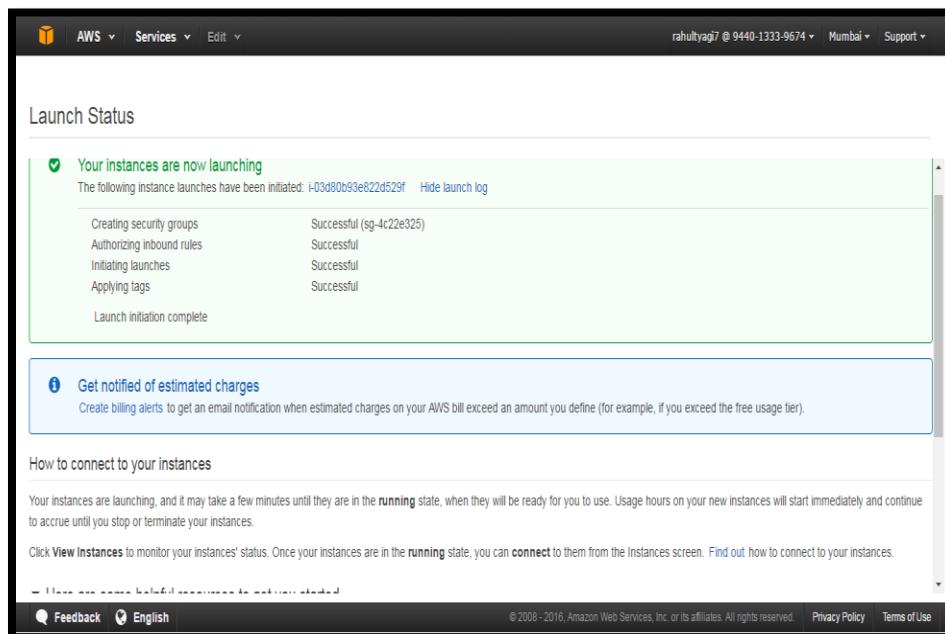
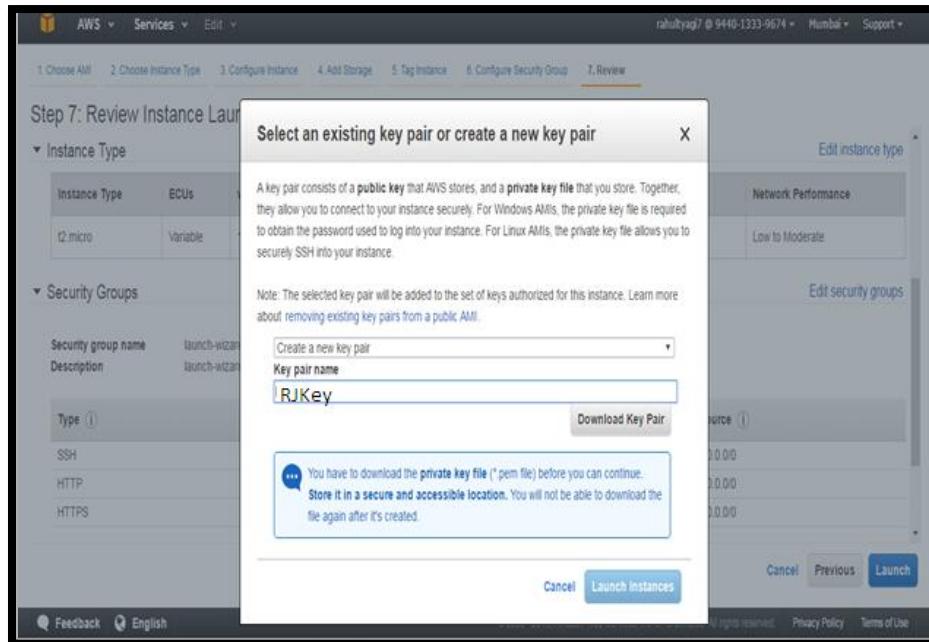


## Review the Instance before launch



Once by Launching instance.thedailoug box is pop up.by the massage select key creating option.

And Download it.



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 Instances page. The left sidebar includes options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, and Placement Groups. The main content area has tabs for Launch Instance, Connect, and Actions. A search bar shows results for 'i-03d80b93e822d529f'. The table lists one instance: i-03d80b93e822d529f, t2.micro, ap-south-1a, running, Initializing, None, ec2-52-66-122-231.ap-south-1.compute.amazonaws.com, 52.66.122.231. Below the table, detailed information for the instance is shown in a card:

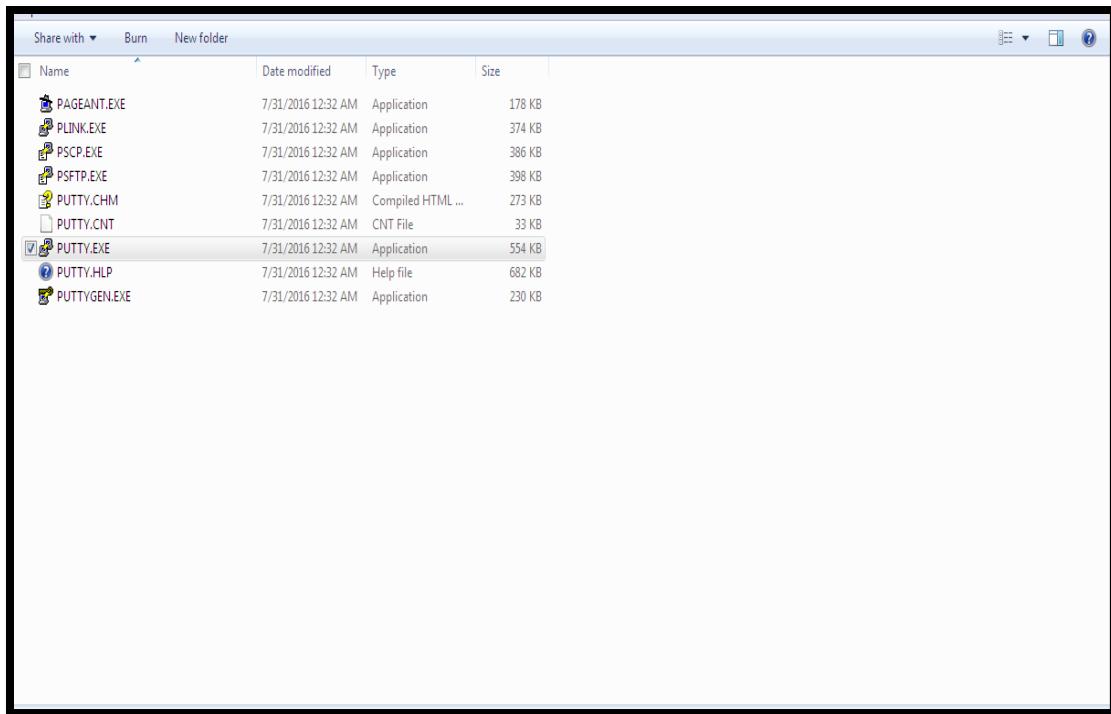
Description		Status Checks		Monitoring		Tags	
Instance ID	i-03d80b93e822d529f	Public DNS	ec2-52-66-122-231.ap-south-1.compute.amazonaws.com	Public IP	52.66.122.231	Elastic IPs	
Instance state	running	Private DNS	ip-172-31-29-235.ap-south-1.compute.internal	Availability zone	ap-south-1a	Security groups	launch-wizard-1, view rules
Instance type	t2.micro	Private IPs	172.31.29.235	Scheduled events	No scheduled events	AMI ID	amzn-ami-hvm-2016.03.3.x86_64-002 (ami-03d80b93e822d529f)
Secondary private IPs		VPC ID	vpc-7ada3913				

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

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

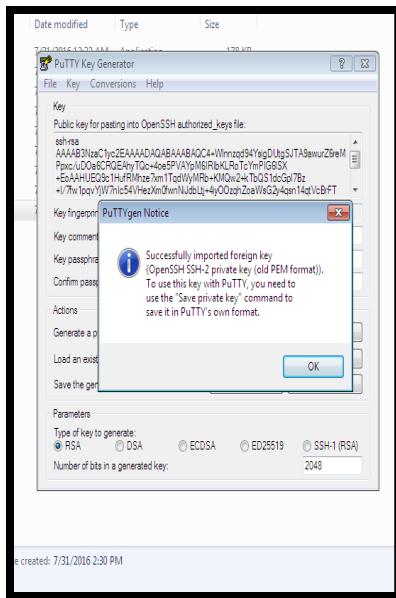
Download the puttygen

and putty for the key configuration..



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 use for verification for secure connection.



## Save the private key.

The screenshot shows the AWS EC2 Dashboard. On the left, the navigation menu includes AWS Services, EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, AMIs, and Elastic Block Store. The Instances section is selected, showing details for an instance named "i-03d80b93e822d529". The "Description" tab is active, showing the instance ID, name, state, type, DNS, and VPC ID. The "Status Checks" tab shows one check with status "None". The "Actions" tab contains the Putty Key Generator dialog box. The dialog box has the following fields:

- Key:** Public key for pasting into OpenSSH authorized\_keys file.
- Key fingerprint:** ssh-rsa 2048 35:46:84:26:11:9a:51:71:73:5f:29:6c:85:a3:be:89
- Key comment:** imported.openssh-key
- Key passphrase:** (empty)
- Confirm passphrase:** (empty)
- Actions:**
  - Generate a public/private key pair (Generate button)
  - Load an existing private key file (Load button)
  - Save the generated key (Save public key and Save private key buttons)
- Parameters:**
  - Type of key to generate: RSA (radio button selected)
  - DSA (radio button)
  - ED25519 (radio button)
  - SSH-1 (RSA) (radio button)
  - Number of bits in a generated key: 2048

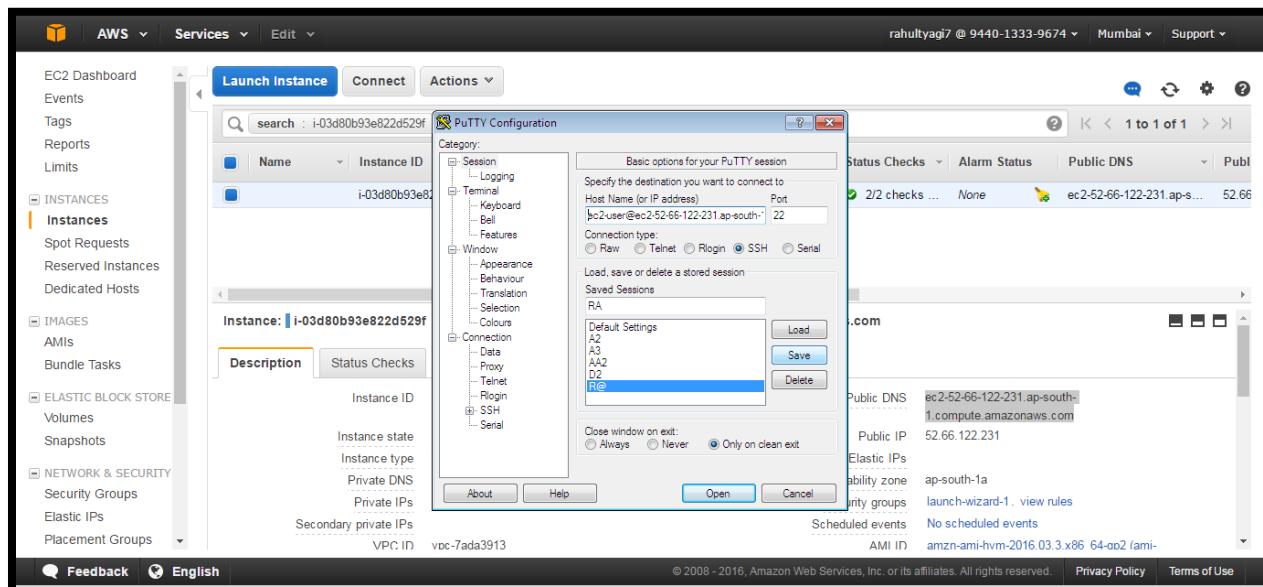
Now open the putty for the key configuration.

In Host Name(Or IP address) write

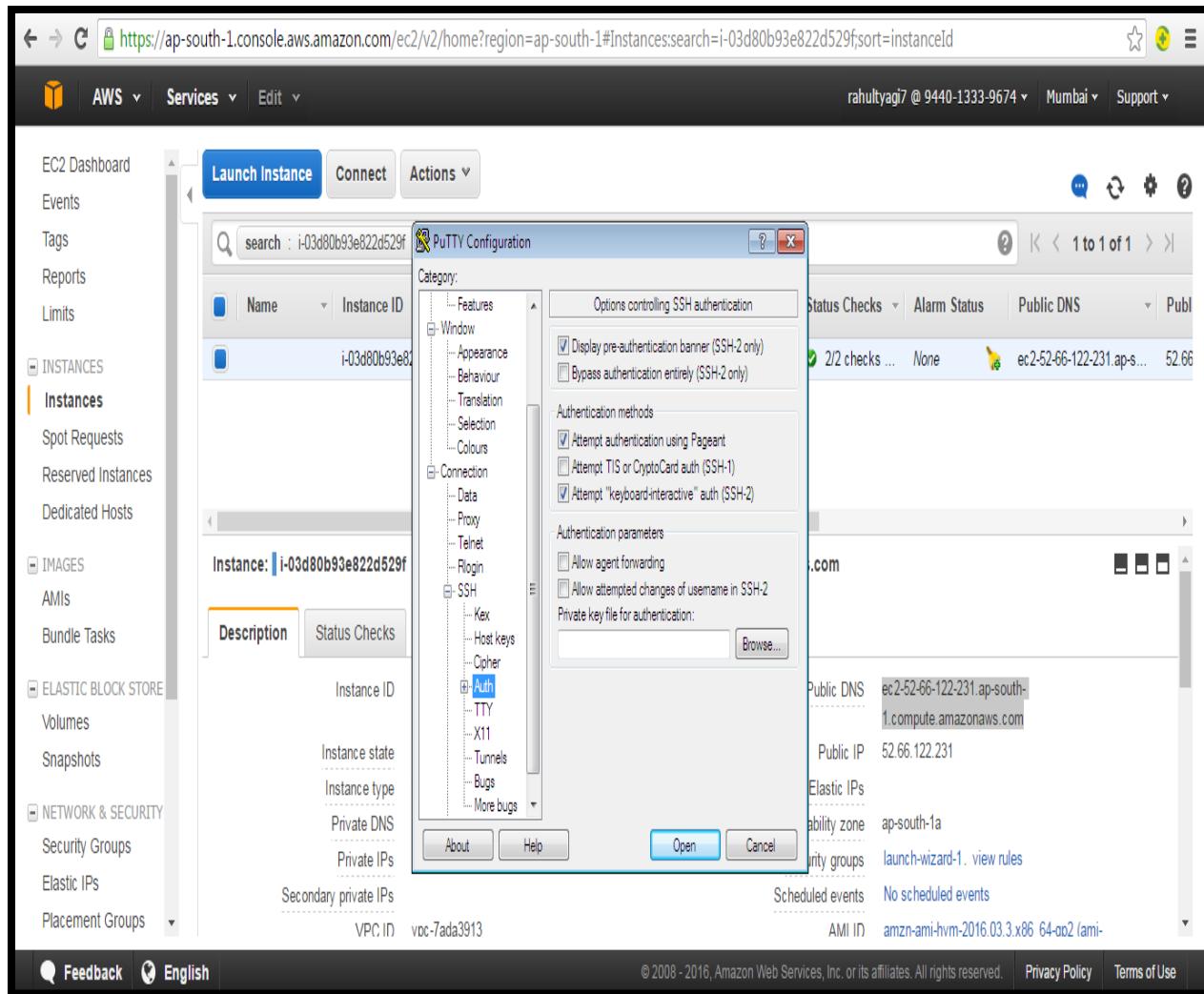
Ec2-user@public DNS.

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.....

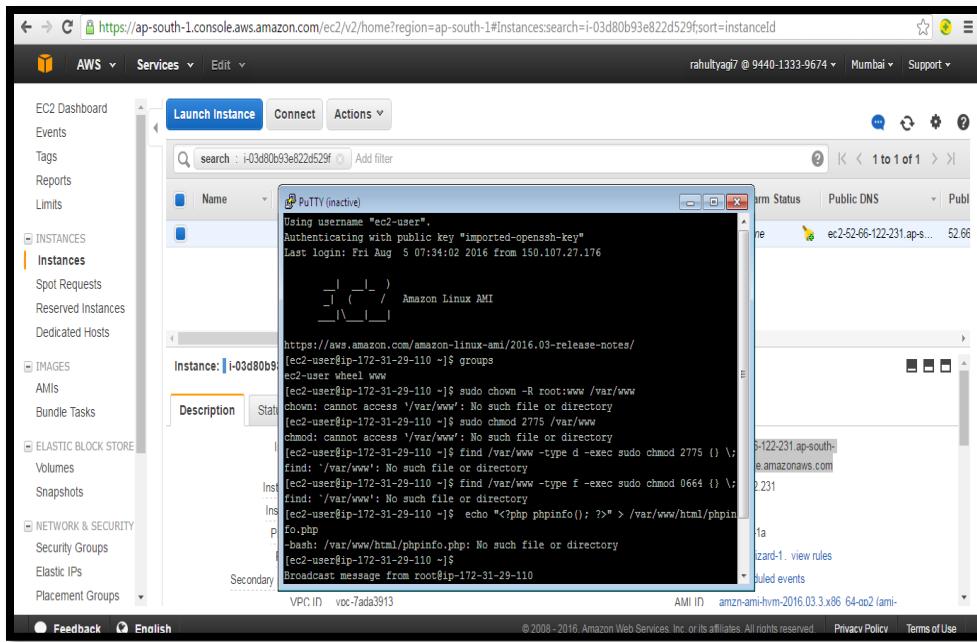


Now select the private key you have save.



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

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



## Command for launching Lamp server.

```

new1 - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
New1
1 1 Connect to your instance.
2 -----
3
4 2 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-mysqlnd
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

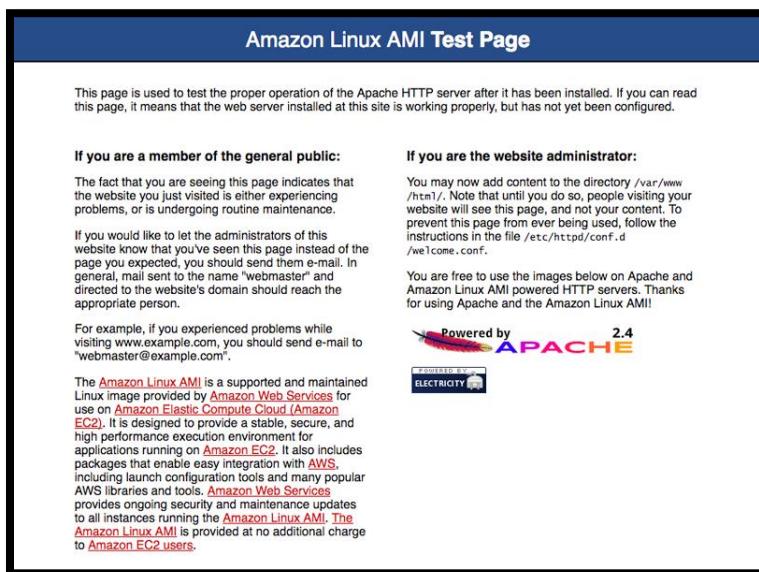
```

This is a screenshot of a Windows Notepad++ application window titled "new1". It contains a list of numbered steps for launching a LAMP server. The steps include connecting to an instance, updating software packages, installing required packages, starting the Apache web server, and configuring it to start at boot. Several lines of the command history are highlighted with red underlines, likely indicating specific commands or parts of the process that were run in a terminal session.

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

A screenshot of a terminal window titled "new 1". The window shows a series of commands being run in a shell. The commands include setting up the Apache document root, adding the "www" group, and adding the user to the "www" group. The terminal also displays a note about logging out and back in to pick up the new group. The bottom status bar shows "length: 2102 lines: 66 Ln: 16 Col: 39 Sel: 0|0 Dos:Windows UTF-8 INS".

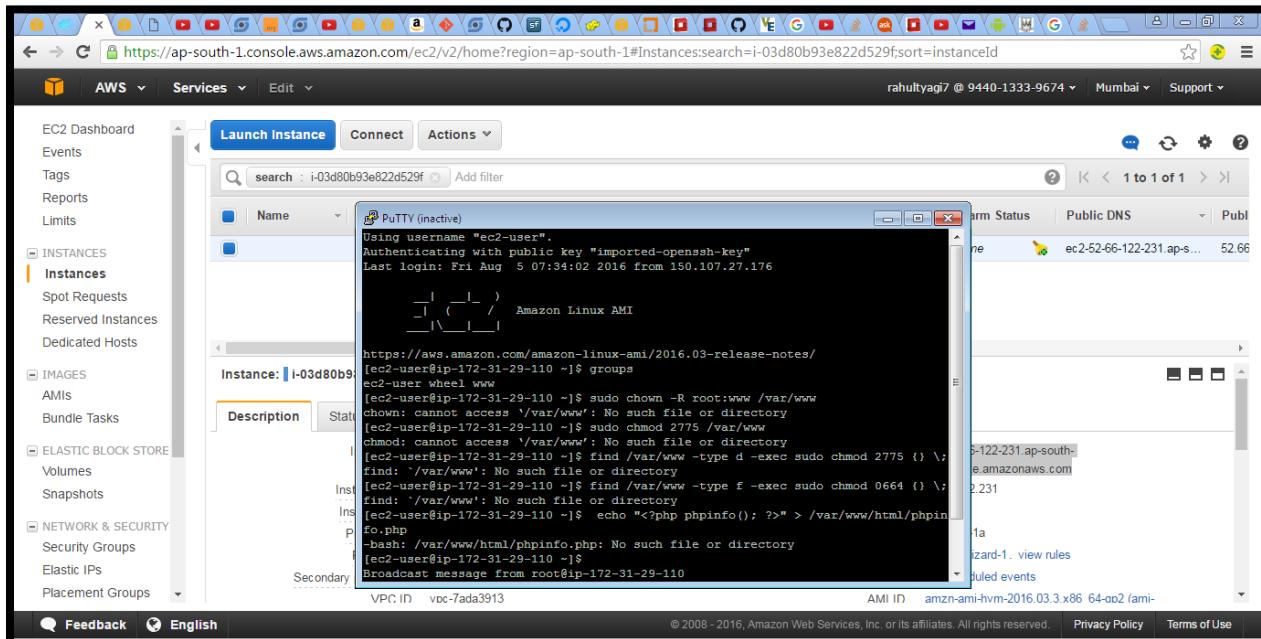
```

new-Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
File New Open Save Save As Print Find Replace Go Preferences Help
New12
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 "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```



Now

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.d20-bz2.ini, /etc/php-5.6.d20-calendar.ini, /etc/php-5.6.d20-ctype.ini, /etc/php-5.6.d20-curl.ini, /etc/php-5.6.d20-dom.ini, /etc/php-5.6.d20-exif.ini, /etc/php-5.6.d20-fontini.ini, /etc/php-5.6.d20-ftp.ini, /etc/php-5.6.d20-gettext.ini, /etc/php-5.6.d20-iconv.ini, /etc/php-5.6.d20-mysqlnd.ini, /etc/php-5.6.d20-pdo.ini, /etc/php-5.6.d20-phar.ini, /etc/php-5.6.d20-posix.ini, /etc/php-5.6.d20-shmop.ini, /etc/php-5.6.d20-simplexml.ini, /etc/php-5.6.d20-sockets.ini, /etc/php-5.6.d20-splitini.ini, /etc/php-5.6.d20-sysvmsg.ini, /etc/php-5.6.d20-sysvsem.ini, /etc/php-5.6.d20-sysvshm.ini, /etc/php-5.6.d20-tokenizer.ini, /etc/php-5.6.d20-xml.ini, /etc/php-5.6.d20-xmwriter.ini, /etc/php-5.6.d20-xsl.ini, /etc/php-5.6.d20-zip.ini, /etc/php-5.6.d30-mysqli.ini, /etc/php-5.6.d30-mysqliini.ini, /etc/php-5.6.d30-pdo_mysql.ini, /etc/php-5.6.d30-pdo_sqlite.ini, /etc/php-5.6.d30-wddx.ini, /etc/php-5.6.d30-xmireader.ini, /etc/php-5.6.d40-json.ini, /etc/php-5.6.diph.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.

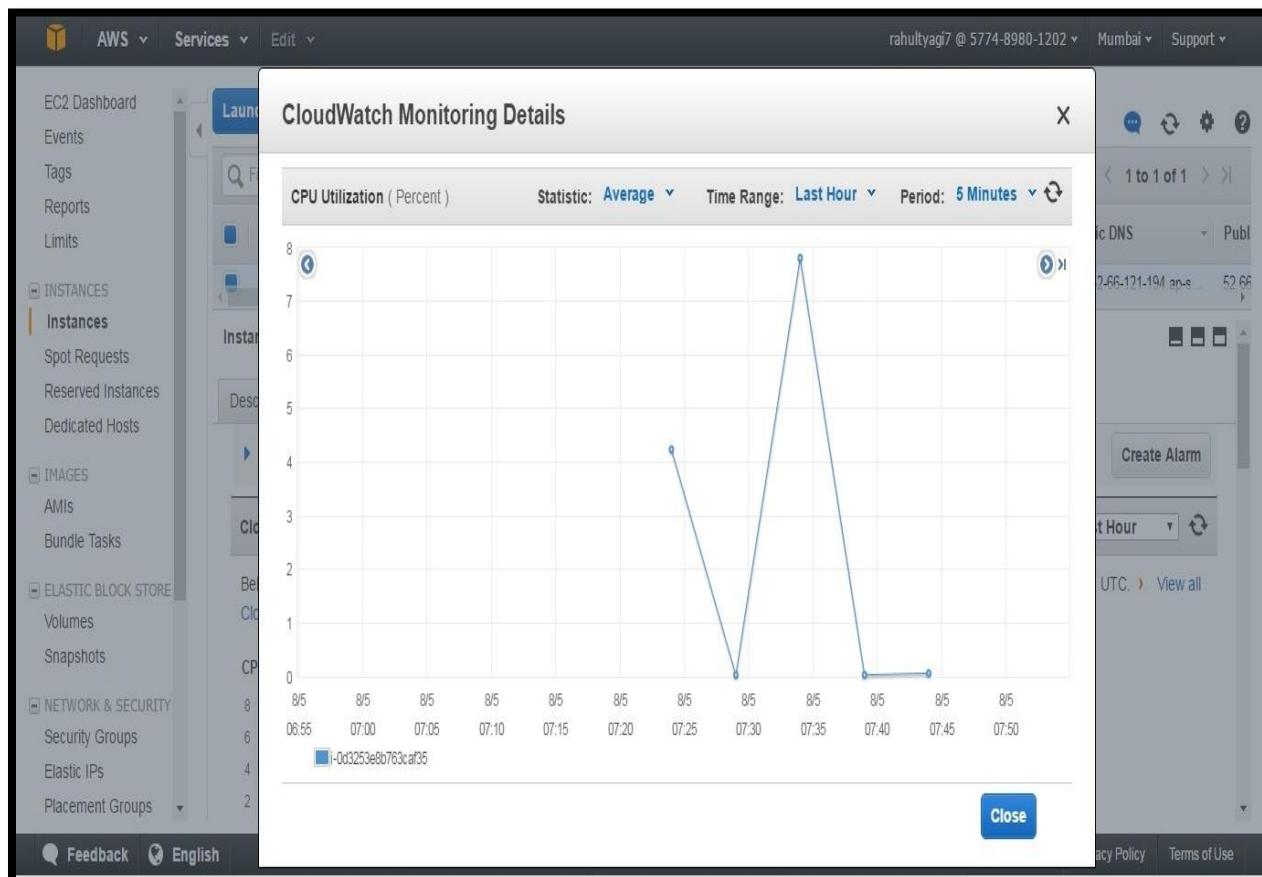
## Practical No : 4

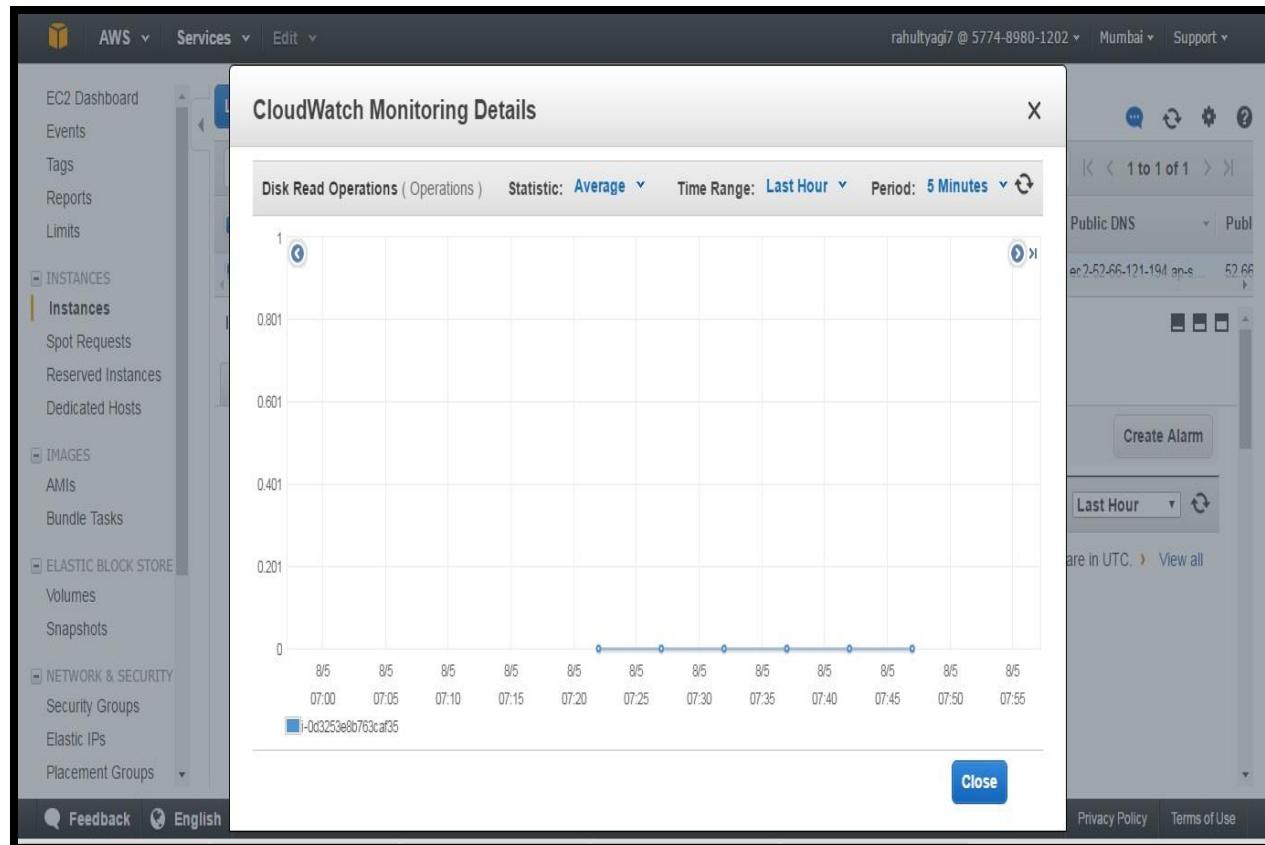
Aim : Demonstrate any Cloud Monitoring tool.

### 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..





## Practical 5

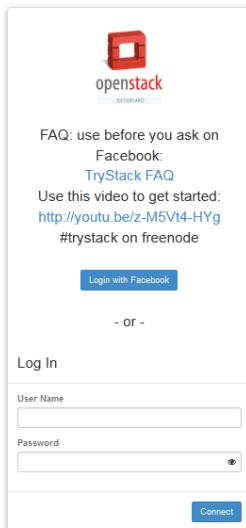
Aim : Evaluate a Private IAAS Cloud using TryStack

Note :

1. Install cygwin and then start with practical.

Go to link

<https://x86.trystack.org/dashboard/auth/login/?next=/dashboard>  
L



Login with facebook account....

After login.....

The screenshot shows the OpenStack dashboard for project 'facebook1044928395592108'. The left sidebar is collapsed, and the main content area displays the 'Overview' page. It includes a 'Limit Summary' section with five pie charts showing usage for Instances, VCPUs, RAM, Floating IPs, and Security Groups. Below this is a 'Usage Summary' section with a date range selector from '2018-07-26' to '2018-07-27', a 'Submit' button, and a note about date format. A message below states 'Active Instances: 0 Active RAM: 0Bytes This Period's VCPU-Hours: 0.00 This Period's GB-Hours: 0.00 This Period's RAM-Hours: 0.00'. The 'Usage' section contains a table with columns for Instance Name, VCPUs, Disk, RAM, and Time since created, showing 'No items to display.' at the moment.

Go to Network(left side) ->

The screenshot shows the OpenStack dashboard for project 'facebook10153965192826476'. The left sidebar is expanded, showing sections for Project, Compute, Network, Object Store, and Identity. The 'Network' section is currently selected. The main content area displays the 'Networks' page with a table header for Name, Subnets Associated, Shared, Status, Admin State, and Actions. A message below the table says 'No items to display.' and 'Displaying 0 items'. On the right side of the table, there is a 'Create Network' button.

Click on Create Network(Right side tab)

**Create Network**

Network Subnet Subnet Details

Network Name: RJC

Admin State: UP

Create Subnet

Create a new network. In addition, a subnet associated with the network can be created in the next panel.

Cancel « Back Next »

## At Subnet tab

**Create Network**

Network Subnet Subnet Details

Subnet Name: RJCSub

Network Address: 192.168.27.0/24

IP Version: IPv4

Gateway IP: [empty input field]

Disable Gateway

Create a subnet associated with the network. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel « Back Next »

## At subnet Details

Specify additional attributes for the subnet.

Enable DHCP

Allocation Pools ?

DNS Name Servers ?

Host Routes ?

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

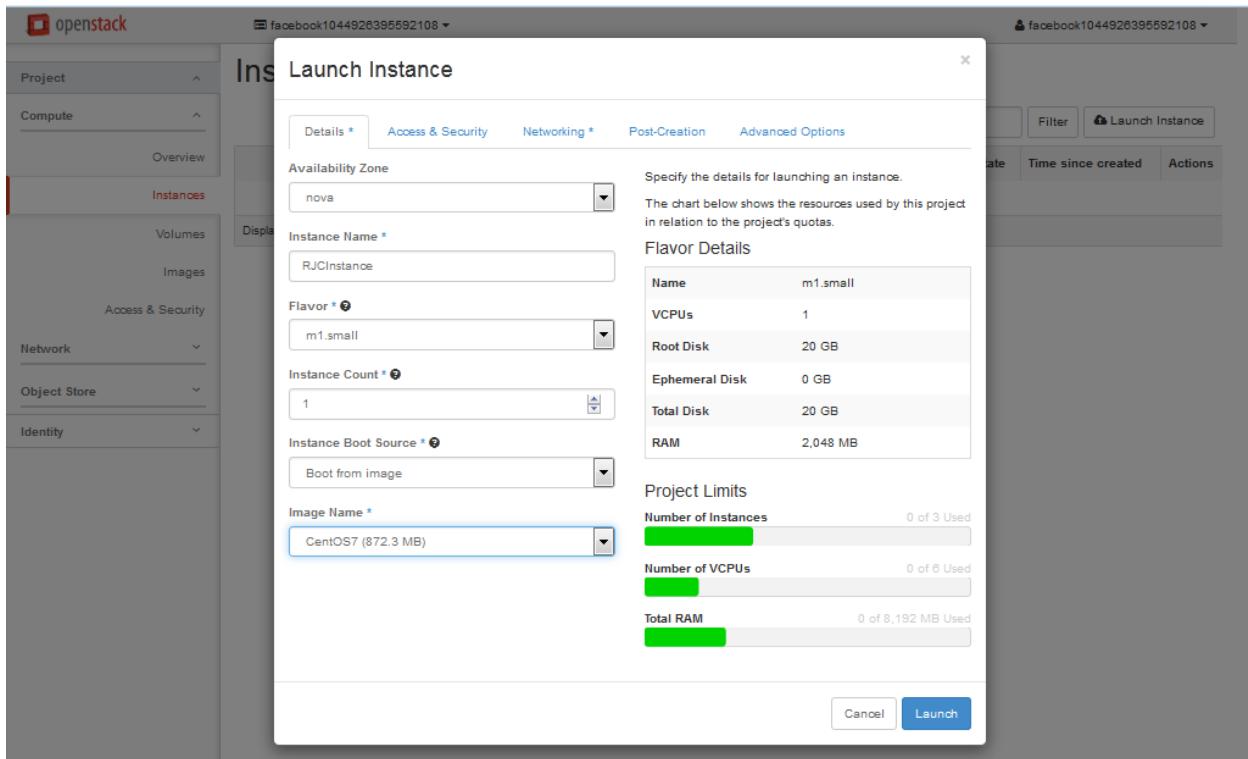
So we have created network

Networks

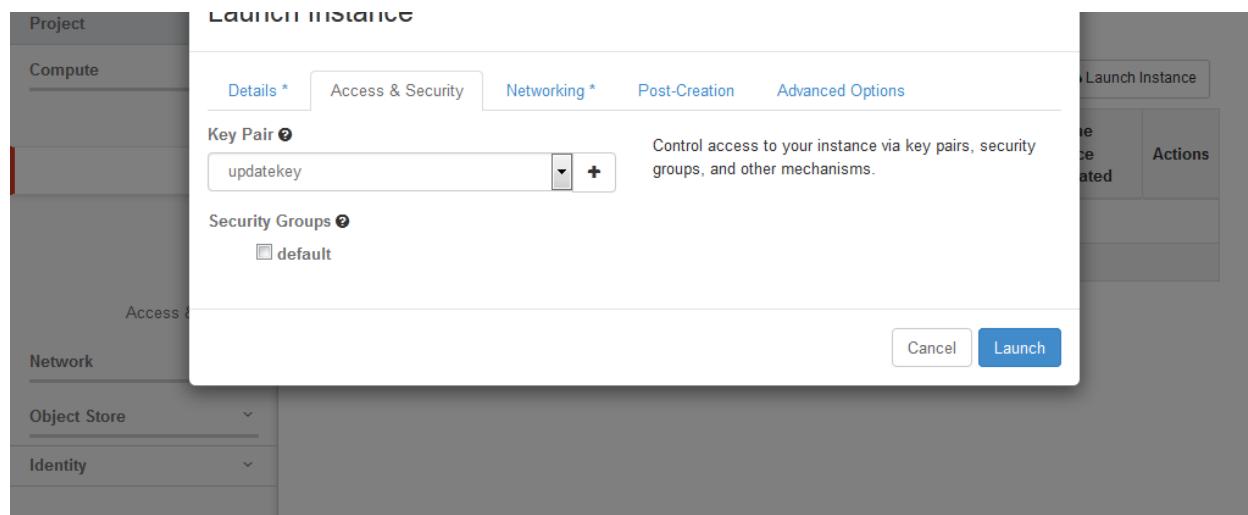
<input type="checkbox"/>	Name	Subnets Associated	Shared	Status	Admin State	Actions
<input type="checkbox"/>	RJC	RJC Sub 192.168.27.0/24	No	Active	UP	<input type="button" value="Edit Network"/>

Displaying 1 item

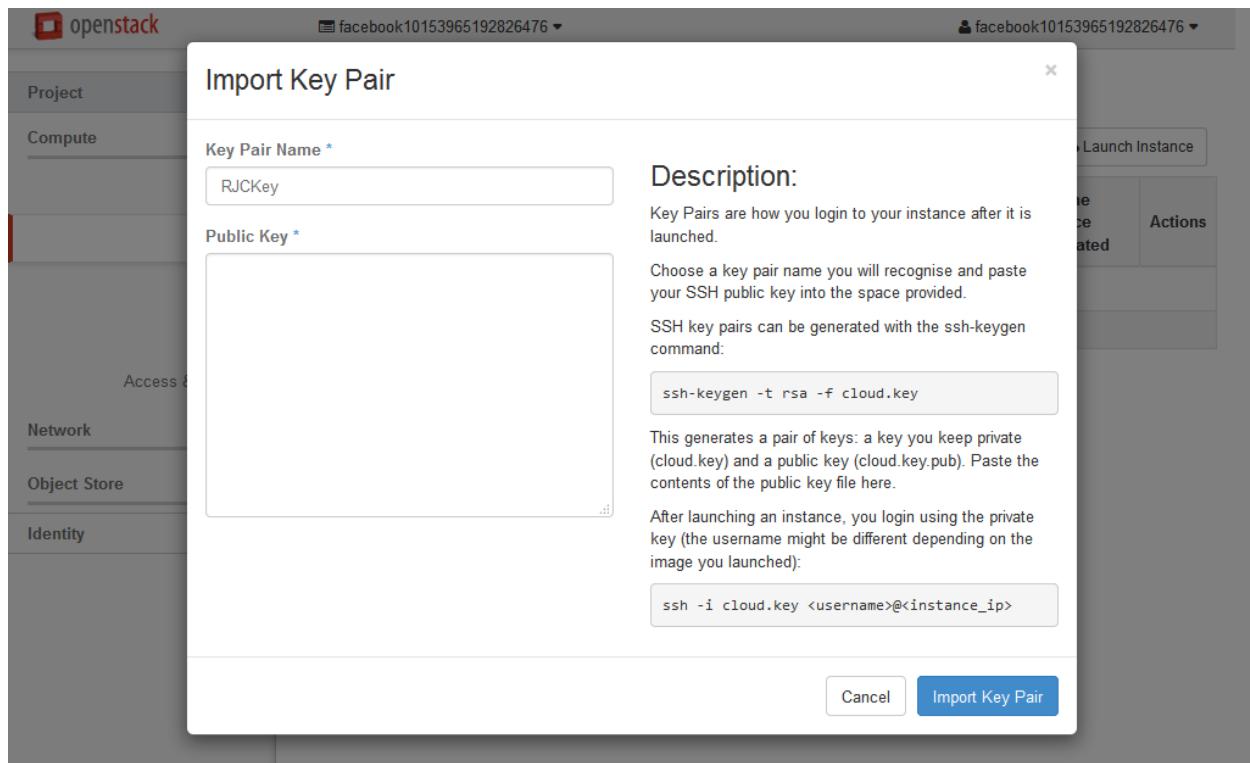
Go to compute(left side) -> Instance-> Launch Instance(Right Side)



## Go to Access & security



Click on “ + “ Sign



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

```

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/Y9YkIPyprduFmBoPfbklia1++v8 MSC@MSC-2
The key's randomart image is:
+---[RSA 2048]----+
|   ... 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
m8YNdoAIC88Pnwu4TKnGRwMSyZeKnEfUXpZ9PrBg2fQrpHbuTBzSpMz0p31DyRz0S9xKpJr+vYhZ31tSAEosMQG0TM
jYsjjoAkblDkNzqR1Dyz7Paz9/I0Cse3uwJa1yKDUFLQO/H20okd2NwBcFWFYTOiv/7cejsP08j1F0pisqCCTgbCpL
pTtwo3dbovRP5GVa4d3wo6sAP1cw9ikASfvuHf7yEtqs2GirG/EBbWBjwx8f+Vi1bdbM2ME2zt/4gzXd5mD8cZmuXkP
czmwt15JH+OEAvvh MSC@MSC-2

MSC@MSC-2 ~
$
```

Select public key from terminal as shown below

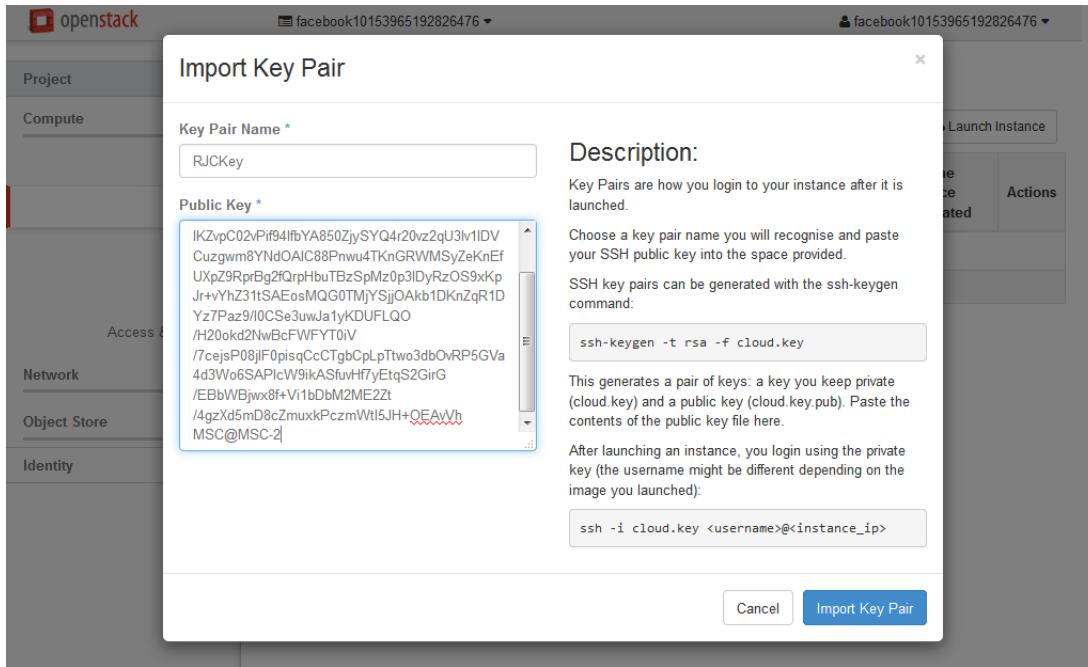
```

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/Y9YkIPyprduFmBoPfbklia1++v8 MSC@MSC-2
The key's randomart image is:
+---[RSA 2048]----+
|   ... 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
m8YNdoAIC88Pnwu4TKnGRwMSyZeKnEfUXpZ9PrBg2fQrpHbuTBzSpMz0p31DyRz0S9xKpJr+vYhZ31tSAEosMQG0TM
jYsjjoAkblDkNzqR1Dyz7Paz9/I0Cse3uwJa1yKDUFLQO/H20okd2NwBcFWFYTOiv/7cejsP08j1F0pisqCCTgbCpL
pTtwo3dbovRP5GVa4d3wo6sAP1cw9ikASfvuHf7yEtqs2GirG/EBbWBjwx8f+Vi1bdbM2ME2zt/4gzXd5mD8cZmuXkP
czmwt15JH+OEAvvh MSC@MSC-2

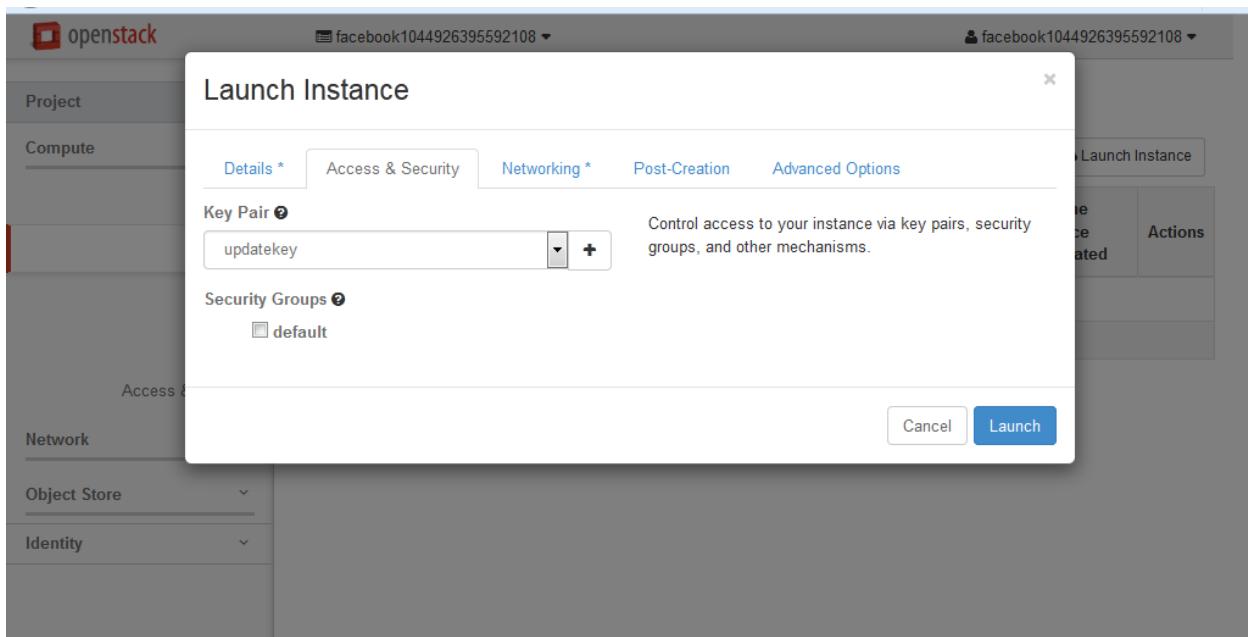
MSC@MSC-2 ~
$
```

Now copy it as shown below

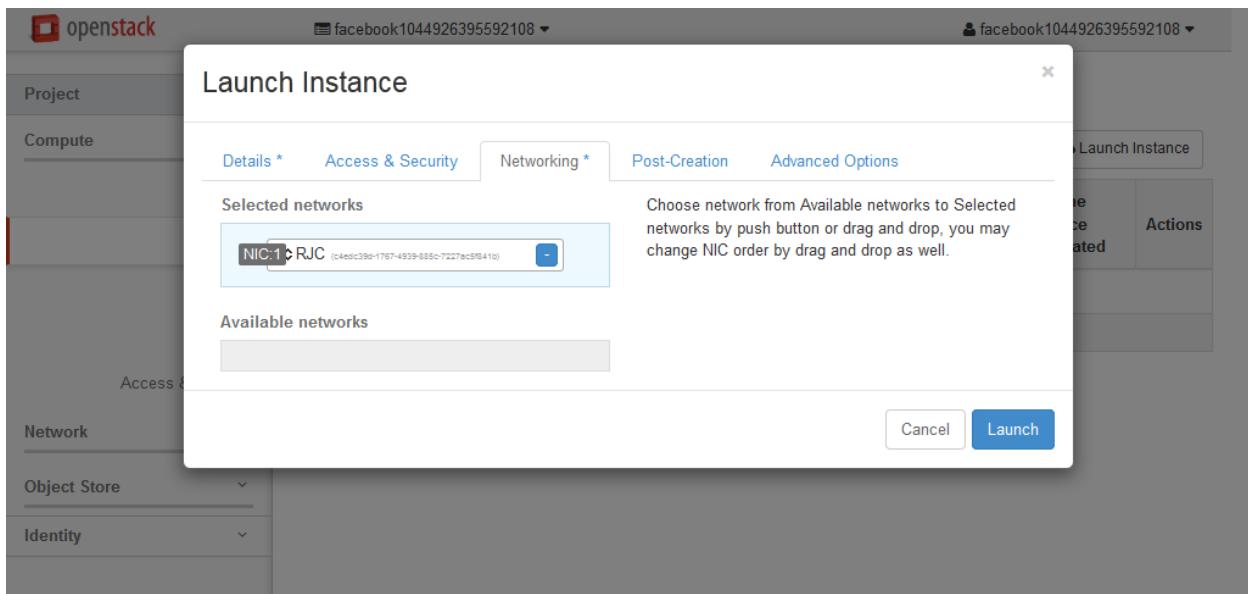


Click on Import key pair.....

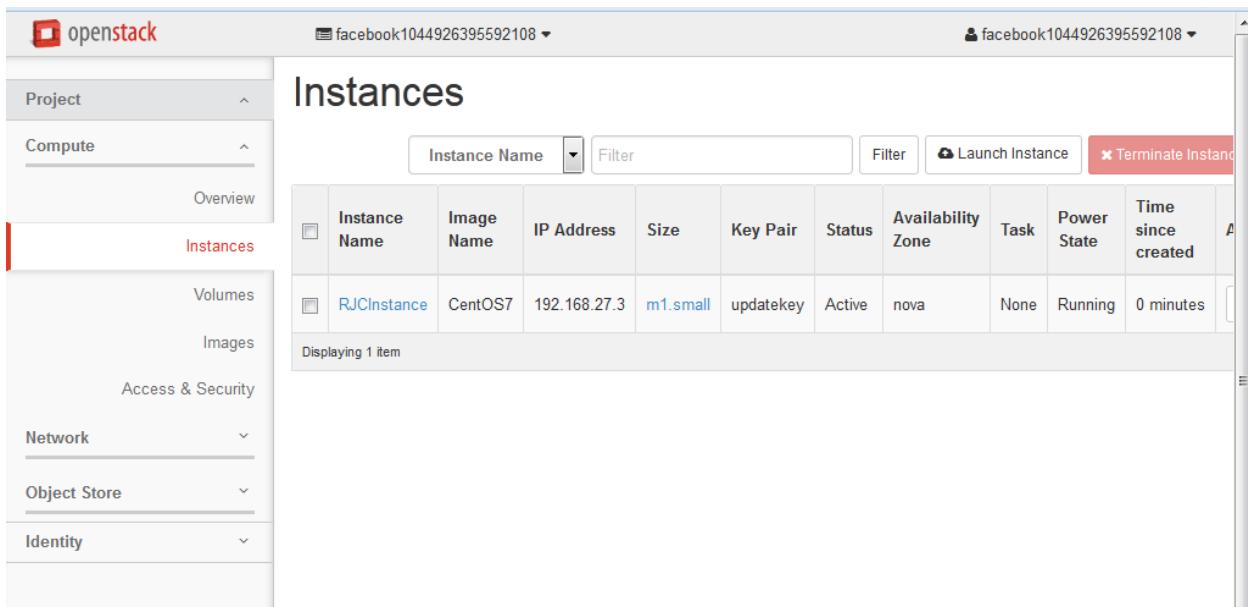
Select name of the key from dropdown.....



At networking tab



Click on Launch



Go to Network -> Routers -> Create Routers

openstack facebook1044926395592108

Create Router

Router Name \* RJCRouter

Admin State UP

External Network public

Description:  
Creates a router with specified parameters.

Cancel Create Router

Project Compute Network Object Store Identity

+ Create Router Actions

This screenshot shows the 'Create Router' dialog box from the OpenStack interface. The 'Router Name' field is populated with 'RJCRouter'. The 'Admin State' dropdown is set to 'UP'. The 'External Network' dropdown is set to 'public'. A descriptive text below the form states 'Creates a router with specified parameters.' At the bottom right of the dialog are 'Cancel' and 'Create Router' buttons. The background shows the main navigation menu with 'Network' selected.

Click on Create Router

openstack facebook1044926395592108

Routers

Filter + Create Router (Quota exceeded) ✖ Delete Routers

Name	Status	External Network	Admin State	Actions
RJCRouter	Active	public	UP	<a href="#">Clear Gateway</a>

Displaying 1 item

Project Compute Network Object Store Identity

Routers

This screenshot shows the 'Routers' list page from the OpenStack interface. A single router named 'RJCRouter' is listed with the status 'Active', external network 'public', and admin state 'UP'. There is a 'Clear Gateway' action button next to the router entry. The 'Routers' tab is selected in the sidebar. The background shows the main navigation menu with 'Network' selected.

Click on Name of Router (RJCRouter)

The screenshot shows the openstack interface with the title "Router Details". The left sidebar has sections for Project, Compute, Network (Network Topology, Networks), Routers (highlighted in red), Object Store, and Identity. The main content area has tabs for Overview, Interfaces (selected), and Static Routes. Under "Overview", there are details for a router named "RJCRouter" with ID "9349d0cb-6b1a-4a44-b46c-3b4461738b92", Project ID "106610f9a1f242bbb62c6ac8db2a96f", Status "Active", and Admin State "UP". The "External Gateway" section shows a public network with IP address 8.43.87.39 and SNAT enabled. A red "Clear Gateway" button is at the top right.

Go to Interfaces->

This screenshot shows the same openstack interface as above, but the "Interfaces" tab is selected under the "Router Details" header. The "Add Interface" button is visible. The table below shows no items displayed, with the message "No items to display." and "Displaying 0 items".

Add Interface

Add Interface

**Subnet \***

RJC: 192.168.27.0/24 (RJCSUB)

**IP Address (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.

Click on Add Interface

openstack

facebook1044926395592108

Project Compute Network

Network Topology Networks Routers

Object Store Identity

## Router Details

Overview Interfaces Static Routes

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

Displaying 1 item

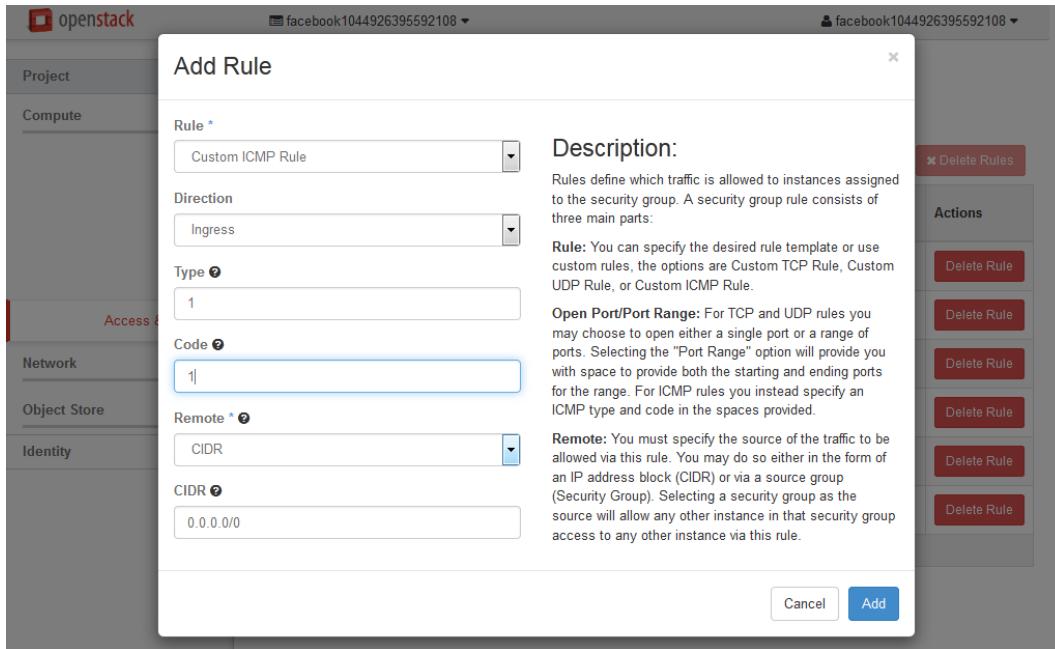
Go to Compute -> Acess & Security

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 currently selected). Below these are Network, Object Store, and Identity options. The main content area is titled "Access & Security" and shows the "Security Groups" tab selected. It includes a search bar with a filter, a "Create Security Group" button, and a "Delete Security Groups" button. A table displays one item: "default" (Description: Default security group). A "Manage Rules" button is located next to the table. At the bottom, it says "Displaying 1 item".

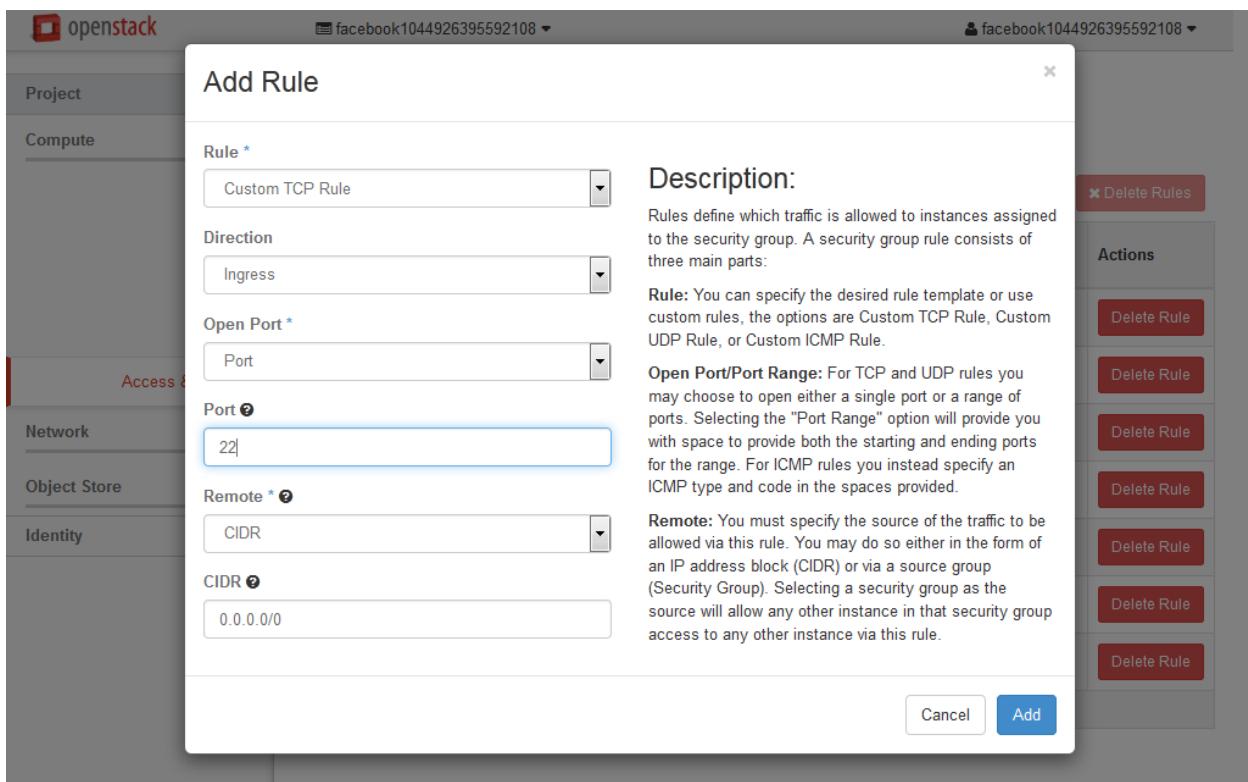
Click on Manage Rules

The screenshot shows the "Manage Security Group Rules" interface for the "default" security group. The title is "Manage Security Group Rules: default (f75cb1d2-2f94-4e2b-902c-aa9b6caee607)". It features a header with "Add Rule" and "Delete Rules" buttons. A table lists six rules: 1. Ingress IPv6 Any Any - default (Delete Rule) 2. Egress IPv6 Any Any ::/0 - (Delete Rule) 3. Ingress IPv4 Any Any - default (Delete Rule) 4. Egress IPv4 Any Any 0.0.0.0/0 - (Delete Rule) 5. Ingress IPv4 ICMP Any 0.0.0.0/0 - (Delete Rule) 6. Ingress IPv4 TCP 22 (SSH) 0.0.0.0/0 - (Delete Rule) At the bottom, it says "Displaying 6 items".

Click on AddRule & select following options



Again click on AddRule & now insert following options



After adding 2 rules

The screenshot shows the OpenStack Compute interface for managing security group rules. The left sidebar has sections for Project, Compute (selected), Instances, Volumes, Images, Access & Security (selected), Network, Object Store, and Identity. The main area title is "Manage Security Group Rules: default (f75cb1d2-2f94-4e2b-902c-aa9b6caee607)". It displays a table of rules with columns: Direction, Ether Type, IP Protocol, Port Range, Remote IP Prefix, Remote Security Group, and Actions. There are 6 items displayed.

	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 Instance .....drop down create snapshot & select Associate floating IP

OpenStack Instances

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	m1.small	updatekey	Active	nova	None	Running	18 minutes	<a href="#">Create Snapshot</a>

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**
- Rebuild Instance

Manage Floating IP Associations

Success: Allocated Floating IP 8.43.87.42.

Actions
Allocate Instances

IP Address \*

No floating IP addresses allocated

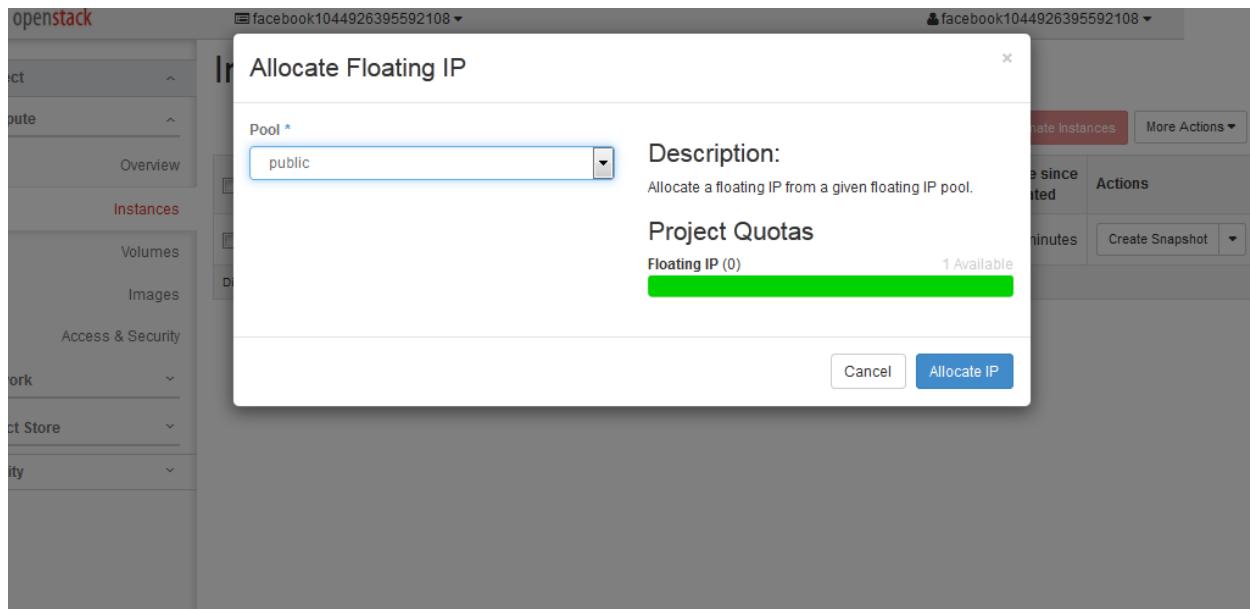
Select the IP address you wish to associate with the selected instance or port.

Port to be associated \*

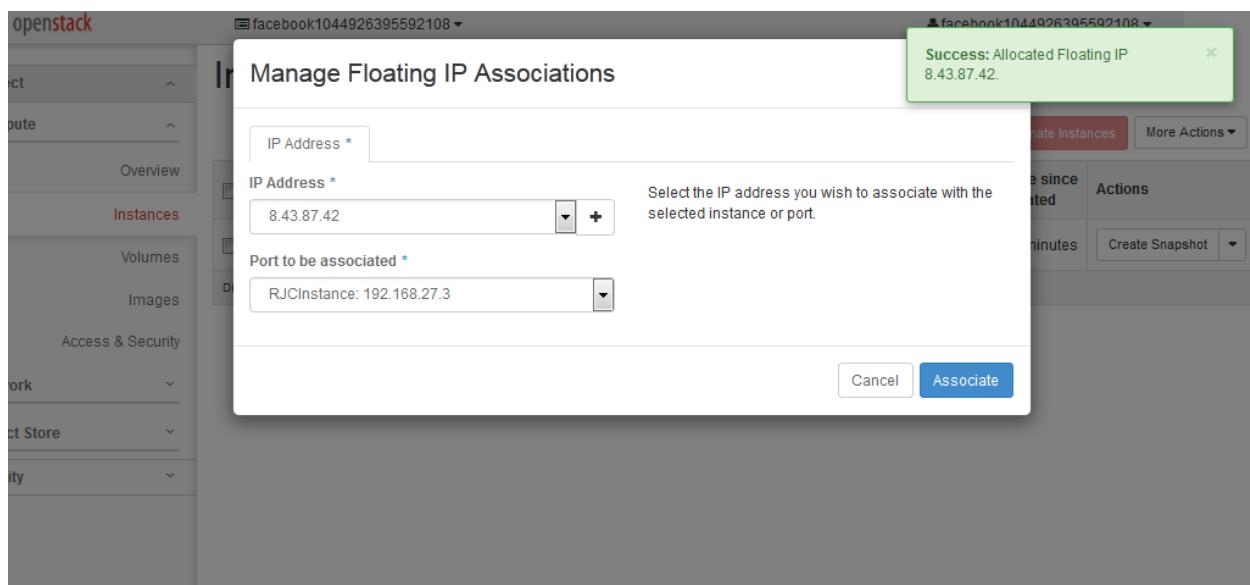
RJInstance: 192.168.27.3

Cancel Associate

Click on “ + “ sign



Click on Allocate IP



Click on Associate

openstack

Project

Compute

Overview

Instances

Volumes

Images

Access & Security

Network

Object Store

Identity

facebook1044926395592108

facebook1044926395592108

## Instances

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
RJCInstance	CentOS7	192.168.27.3 Floating IP: 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

openstack

Project

Compute

Network

Network Topology

Networks

Routers

Object Store

Identity

facebook1044926395592108

facebook1044926395592108

## Network Topology

Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.

Toggle labels    Toggle Network Collapse    Launch Instance    Create Network    Create Router (Quota exceeded)

Conclusion : Successfully evaluated a private cloud.

# 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

The screenshot shows the Amazon DynamoDB console interface. The top navigation bar includes links for Most Visited, Web Development Co..., Sean Halpin - a web d..., Free Vector Icons and ..., Getting Started, AWS Services, and a search bar. The main content area has a blue header bar with the message: "The new DynamoDB console is now complete, and becomes your default experience. Following the preview phase in which we analyzed and incorporated your feedback, we have completed the new DynamoDB console, making it even easier for you to manage your data and resources. Let us know what you think. You can still choose to return to the previous console from the navigation pane." Below this, the main heading is "Amazon DynamoDB: A fast and flexible NoSQL database service for any scale". To the right, there's a "Get started" section with a "Create table" button, and a "Pricing" section with a detailed description of costs. On the left, a sidebar menu lists options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Reserved capacity, DAX, Clusters, Subnet groups, Parameter groups, Events, Tell us what you think, and Return to the previous console. The bottom navigation bar includes links for Feedback, AWS Services, a search bar, and footer links for © 2022, Amazon Internet Services Private Ltd. or its affiliates., Privacy, Terms, and Cookie preferences.

**Create table**

**Table details** Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

1 to 255 characters and case sensitive.

The screenshot shows the 'Create table' step in the AWS DynamoDB console. The configuration includes:

- Read/write capacity**: Using provisioned capacity mode with 5 units each.
- Secondary indexes**: None.
- Table class**: Standard table class.
- Tags**: None.

At the bottom, there are 'Cancel' and 'Create table' buttons.

The screenshot shows the 'Tables' page in the AWS DynamoDB console. The table 'Zeesshan-practical-8' is listed with the following details:

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
Zeesshan-practical-8	Creating	Roll-no (\$)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)

The screenshot shows the AWS DynamoDB console with the URL <https://us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#table?initial=true>. The left sidebar has 'Tables' selected under 'Update settings'. The main area shows a table named 'Zeeshan-Practical-8' with the following details:

- General information:** Partition key is 'Roll-no (Number)', Sort key is '-', Capacity mode is 'Provisioned', Table status is 'Active' with 'No active alarms'.
- Items summary:** Item count is 0, Table size is 0 bytes, Average item size is 0 bytes.

The screenshot shows the same AWS DynamoDB console as above, but the 'Actions' dropdown menu is open, revealing options such as Edit capacity, Update table class, Delete table, Create item, Create index, Create replica, Export to S3, Enable TTL, Manage tags, and Create access control policy.

The screenshot shows the 'Explore items' section of the AWS DynamoDB console. The 'Tables' section lists 'Zeeshan-Practical-8'. The main area displays the 'Scan/Query items' interface for the 'Zeeshan-Practical-8' table, which includes a 'Scan' tab, a 'Table or index' dropdown set to 'Zeeshan-Practical-8', and a 'Filters' section with an 'Attribute name' input field containing 'Enter attribute', a 'Type' dropdown set to 'String', a 'Condition' dropdown set to 'Equal to', and a 'Value' input field containing 'Enter attribute'. There are also 'Add filter', 'Run', and 'Reset' buttons.

## 2) Create Item

The screenshot shows the AWS DynamoDB Item Editor interface. At the top, there's a navigation bar with links like 'Most Visited', 'Web Development Co...', 'Web Development Co...', 'Sean Halpin - a web d...', 'Free Vector Icons and ...', and 'Getting Started'. Below that is the AWS logo and a search bar. The main area is titled 'Create item' and has a 'Form' tab selected. Under 'Attributes', there are three fields: 'Roll-no - Partition key' (Value: 99, Type: Number), 'Name' (Value: Goliath, Type: String), and 'Phone-No' (Value: 6969696969, Type: Number). There are 'Add new attribute' and 'Remove' buttons. At the bottom right are 'Cancel' and 'Create Item' buttons.

## 3) Scan Items.

The screenshot shows the AWS DynamoDB Scan Results page. At the top, it says 'Completed' with a green checkmark and 'Read capacity units consumed: 0.5'. Below that is a section titled 'Items returned (3)'. A table lists three items:

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

▼ Scan/Query items

Table or index

▼ 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)

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

#### 4) Query Items.

▼ Scan/Query items

Scan    Query

Table or index  
Zeeshan-Practical-8

Roll-no (Partition key)  
99

▼ Filters  
Add a filter to get started.  
Add filter

Run    Reset

Completed   Read capacity units consumed: 0.5

Items returned (1)

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

## 5) Delete table

Delete table

You are about to delete a table.

- Zeeshan-Practical-8

Delete all CloudWatch alarms for this table.

Create a backup of this table before deleting it.  
If you do not select this check box, you will not be able to restore data being deleted.

To confirm the deletion of this table, type *delete* in the box.

*delete*

**Cancel**    **Delete table**

The screenshot shows the AWS DynamoDB console interface. On the left, there's a navigation sidebar with 'DynamoDB' selected. The main area displays a success message: 'The request to delete the "Zeehan-Practical-8" table has been submitted successfully.' Below this, the 'Tables (1) Info' section shows a table with one item: 'Zeehan-Practical-8' (Status: Deleting). The table includes columns for Name, Status, Partition key, Sort key, Indexes, Read capacity mode, Write capacity mode, Size, and Table class.

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

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