PRACTICAL: 1

AIM:

To implement Cloud-based infrastructures and services, it is required to set up the complete system requirements. Researchers & industry-based developers can focus on specific system design issues that they want to investigate, without taking more concerned about the low-level details so the Cloudsim is very much useful for these activities and it cansupport simulation environment to implement cloud-based infrastructure solutions.
Overview of Cloudsim functionalities:
Support for modeling and simulation of large-scale Cloud computing data centers
Support for modeling and simulation of virtualized server hosts, with customizable policies for provisioning host resources to virtual machines
Support for modeling and simulation of data center network topologies and message-passing applications
Support for dynamic insertion of simulation elements, stop and resume of simulation
Support for user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines
Perform Cloud Computing Set up using Cloudsim Tool:

1) **Introduction to Cloudsim tool.**
2) **Perform Installation steps of Cloudsim on NetBeans.**

THEORY:

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

- Test application services in a repeatable and controllable environment.
- Tune the system bottlenecks before deploying apps in an actual cloud.
- Experiment with different workload mix and resource performance scenarios on simulated infrastructure for developing and testing adaptive application provisioning techniques Core features of CloudSim are:
- The Support of modeling and simulation of large scale computing environment as federatedcloud data centers, virtualized server hosts, with customizable policies for provisioning host resources to virtual machines and energy-aware computational resources
- It is a self-contained platform for modeling cloud's service brokers, provisioning, and allocation policies.
- It supports the simulation of network connections among simulated system elements.
- Support for simulation of federated cloud environment, that inter-networks resources fromboth private and public domains.
- Availability of a virtualization engine that aids in the creation and management of multipleindependent and co-hosted virtual services on a data center node.

- Flexibility to switch between space shared and time shared allocation of processing coresto virtualized services.

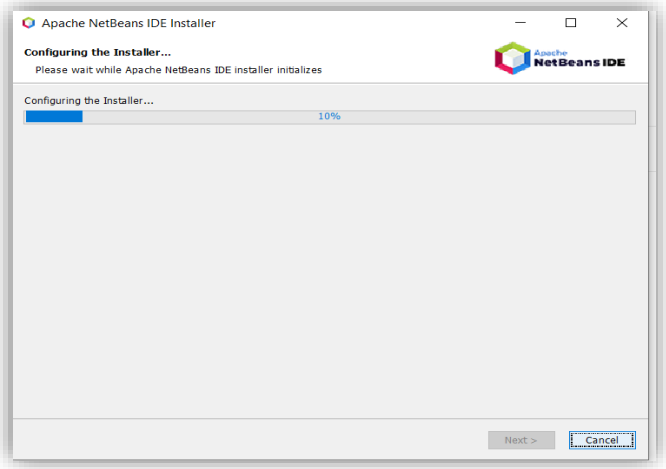
INSTALLATION-STEPS FOR NETBEANS:

You need to have a setup file of the NetBeans JAVA into your setup.
If you didn't have the setup you can download from thefollowing
https://netbeans.org/images_www/v6/download/community/8.2

You can download any type of setup as per your requirements from the above mention web page.

Right-click on the setup or you can Double-Click on the setup by using the mouse.

Click on the next option.

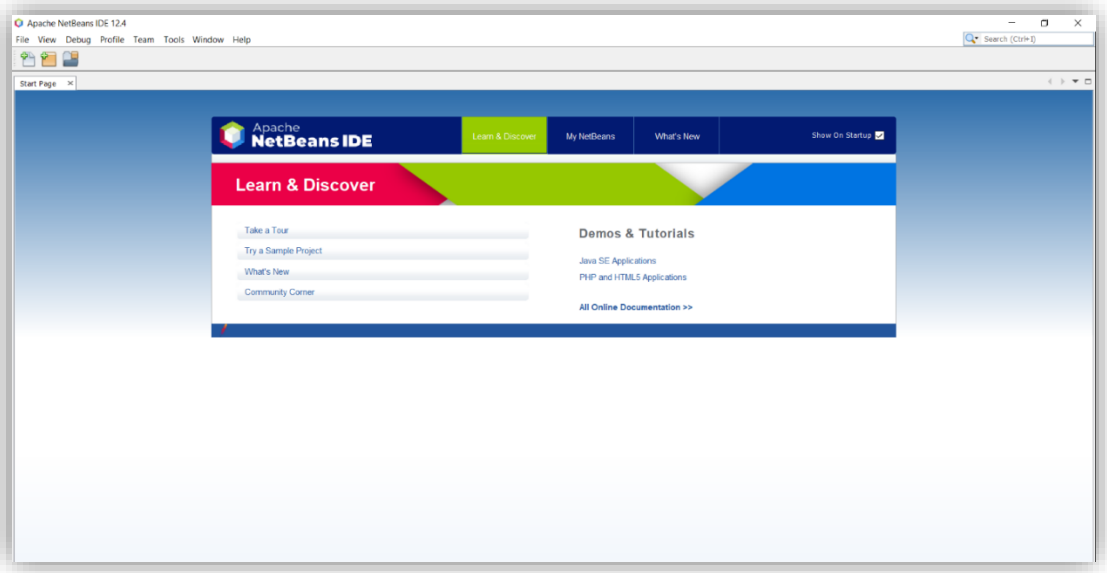


Click on the “Install” button.

Wait for the while till the time the setup is properly Installed into the Computer

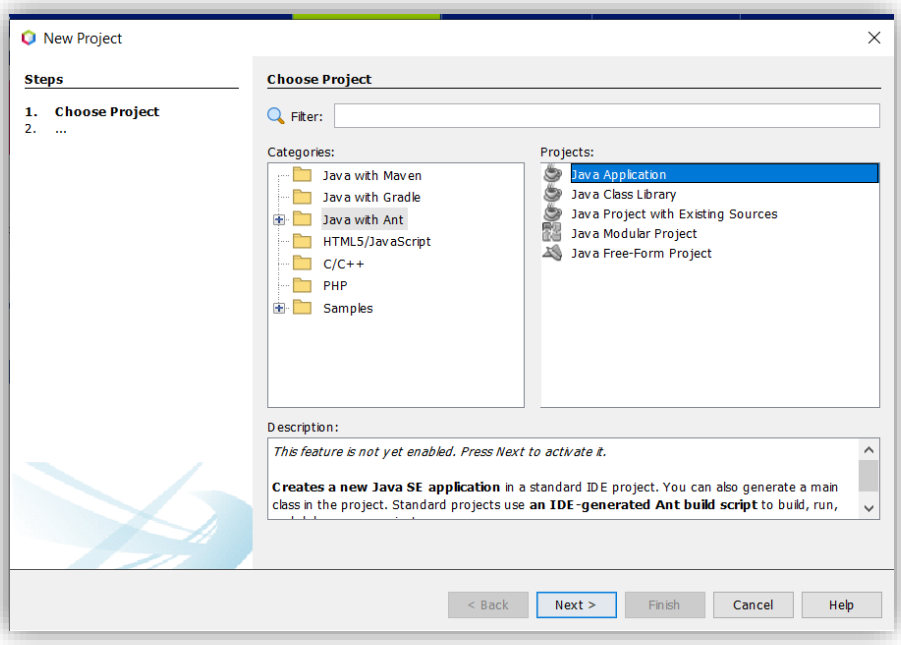
After complication of the setup you can click on the “Finish” button or you can also register the Software, for Further Assistance because it is a Free Software.

Now you can start the NetBeans for further use.



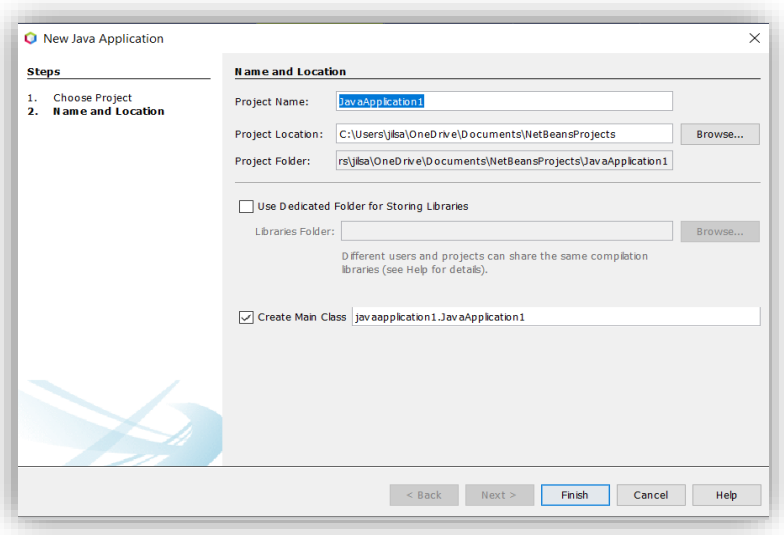
INSTALLATION-STEPS FOR CLOUDSIM:

Open Netbeans, Go to file–>>new project.

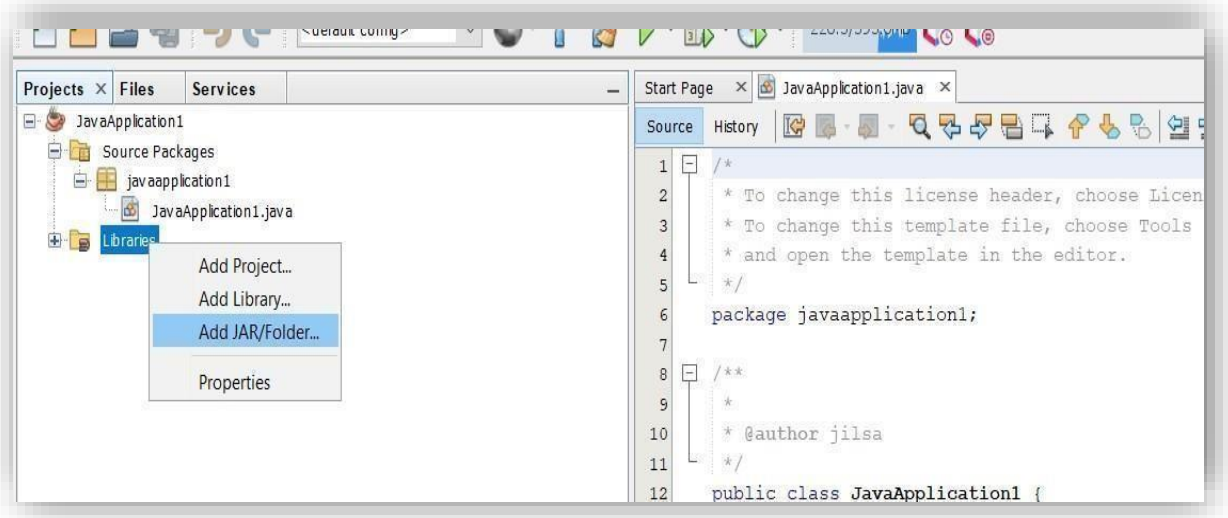


select “Java with Ant” folder then select first option Java Application, Press next

Now give a name to the project as you wish.

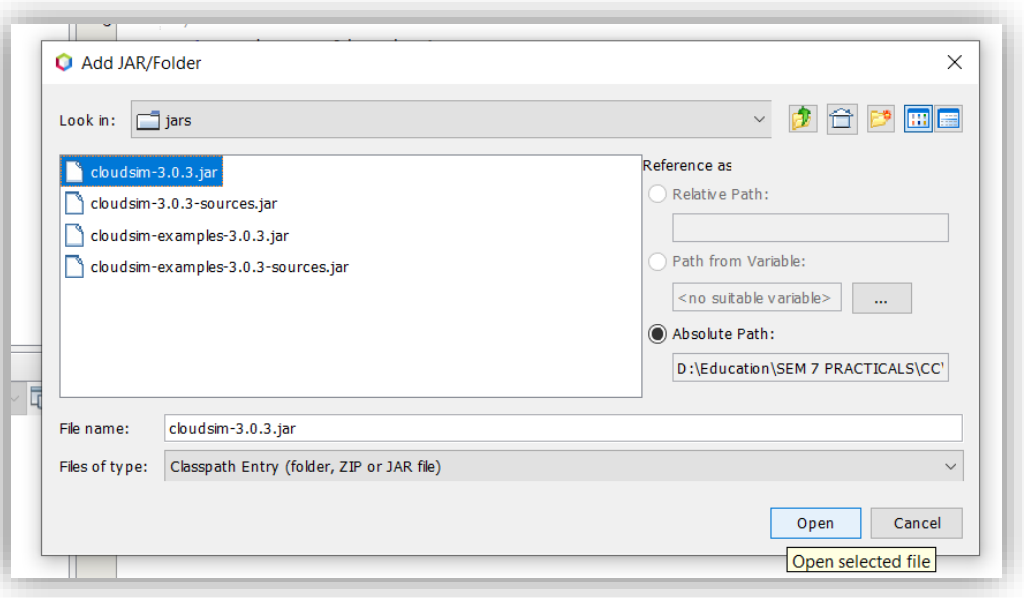


Go to library, right click on it, a menu will come,



click on “Add jars/Folders”

Now browse the cloudsim folder which you have extracted from zip file .and go to that folder and select “cloudsim-3.0.3.jar”.



That’s how cloudsim can be installed in Netbeans.

CONCLUSION:

In this practical, we learnt about Netbeans and Cloudsim. We installed both the tools in oursystem.

PRACTICAL: 2

AIM:

Cloud Computing aims for Internet based application services to deliver reliable, secure, fault-to lerant, scalable infrastructure. It is a tremendous challenging task to model and schedule the different applications and services on real cloud infrastructure which requiresto handle different workload and energy performance parameters. Consider the real-worldanalogy into cloudsim and

Perform following Programs:

1) **Write a program in cloudsim using NetBeans IDE to create a datacenter with one host and runfour cloudlets on it.**
2) **Write a program in cloudsim using NetBeans IDE to create a datacenter with three hosts andrun three cloudlets on it.**

CODE:

(1) Write a program in cloudsim using NetBeans IDE to create a datacenter with one host and run four cloudlets on it.

```
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;
```

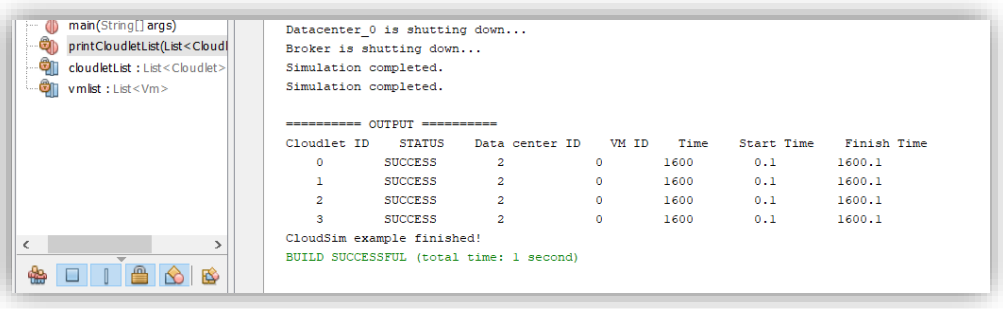
```
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

public class Prac2_1 {

   /**
    * @param args the command line arguments
    */
      private static List<Cloudlet> cloudletList;

   private static List<Vm> vmlist;

   public static void main(String[] args)
   {
      Log.printLine("Starting Cloudsim ...");

      try
      {
         int num_user = 1;
         Calendar calendar = Calendar.getInstance();
         boolean trace_flag = false;

         CloudSim.init(num_user, calendar, trace_flag);

         Datacenter datacenter0 = createDatacenter("Datacenter_0");

         DatacenterBroker broker = createBroker("Broker");
         int brokerId = broker.getId();

         vmlist = new ArrayList<Vm>();

         int vmid = 0;
         int mips = 1000;
         long size = 10000;
         int ram = 512;
         long bw = 1000;
         int pesNumber = 1;
         String vmm = "Xen";

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

         vmlist.add(vm);

         broker.submitVmList(vmlist);

         cloudletList = new ArrayList<Cloudlet>();
```

```
        int id = 0;
        long length = 400000;
        long fileSize = 300;
        long outputSize = 300;
        UtilizationModel utilizationModel = new UtilizationModelFull();

        Cloudlet cloudlet0 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
        utilizationModel, utilizationModel, utilizationModel);
        cloudlet0.setUserId(brokerId);
        cloudlet0.setVmId(vmid);
        cloudletList.add(cloudlet0);

        Cloudlet cloudlet1 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
        utilizationModel, utilizationModel, utilizationModel);
        cloudlet1.setUserId(brokerId);
        cloudlet1.setVmId(vmid);
        cloudletList.add(cloudlet1);

        Cloudlet cloudlet2 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
        utilizationModel, utilizationModel, utilizationModel);
        cloudlet2.setUserId(brokerId);
        cloudlet2.setVmId(vmid);
        cloudletList.add(cloudlet2);

        Cloudlet cloudlet3 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
        utilizationModel, utilizationModel, utilizationModel);
        cloudlet3.setUserId(brokerId);
        cloudlet3.setVmId(vmid);
        cloudletList.add(cloudlet3);

        broker.submitCloudletList(cloudletList);

        CloudSim.startSimulation();

        CloudSim.stopSimulation();

        List<Cloudlet> newList = broker.getCloudletReceivedList();
        printCloudletList(newList);

        Log.printLine("CloudSim example finished!");
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
        Log.printLine("Unwanted error occured");
```

```
    }
  }

  public static Datacenter createDatacenter(String name)
  {
    List<Host> hostList = new ArrayList<Host>();

    List<Pe> peList = new ArrayList<Pe>();

    int mips = 1000;

    peList.add(new Pe(0, new PeProvisionerSimple(mips)));

    int hostId = 0;
    int ram = 2048;
    long storage = 1000000;
    int bw = 10000;

    hostList.add(new Host(hostId, new RamProvisionerSimple(ram), new
    BwProvisionerSimple(bw), storage, peList, new VmSchedulerTimeShared(peList)));

    String arch = "x86";
    String os = "Linux";
    String vmm = "Xen";
    double time_zone = 10.0;
    double cost = 3.0;
    double costPerMem = 0.05;
    double costPerStorage = 0.001;
    double costPerBw = 0.0;

    LinkedList<Storage> storageList = new LinkedList<Storage>();

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

    Datacenter datacenter = null;
    try
    {
      datacenter=new Datacenter(name, characteristics, new
      VmAllocationPolicySimple(hostList), storageList, 0);
    }
    catch(Exception ex) {
      ex.printStackTrace();
    }

    return datacenter;
```

```
    }

    private static DatacenterBroker createBroker(String name) {
        DatacenterBroker broker = null;
        try
        {
            broker = new DatacenterBroker(name);
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
        return broker;
    }

    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 :



(2) Write a program in cloudsim using NetBeans IDE to create a datacenter with three hosts and run three cloudlets on it.

```
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;

public class Prac2_2 {

    private static List<Cloudlet> cloudletList;
    private static List<Vm> vmlist;

    public static void main(String[] args) {
        // TODO code application logic here
        Log.printLine("Staring CloudSim ...");
```

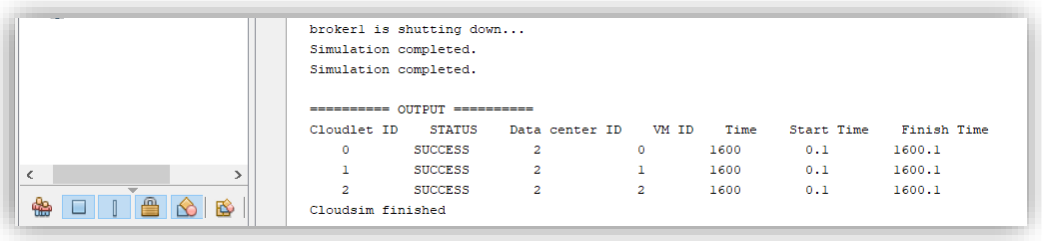
```
try
{
   int num_user = 1;
   Calendar calendar = Calendar.getInstance();
   boolean trace_flag = false;

   CloudSim.init(num_user, calendar, trace_flag);

   Datacenter datacenter0 = createDatacenter("Datacenter_0");

   DatacenterBroker broker = createBroker("broker1");
   int brokerId = broker.getId();

   vmlist = new ArrayList<Vm>();

   int vmid = 0;
   int mips = 250;
   long size = 10000;
   int ram = 512;
   //int ram = 1024;
   long bw = 1000;
   int pesNumber = 1;
   String vmm = "Xen";
   Vm vm0 = new Vm(vmid++, brokerId, mips, pesNumber, ram, bw, size, vmm, new
   CloudletSchedulerTimeShared());
   Vm vm1 = new Vm(vmid++, brokerId, mips, pesNumber, ram, bw, size, vmm, new
   CloudletSchedulerTimeShared());
   Vm vm2 = new Vm(vmid++, brokerId, mips, pesNumber, ram, bw, size, vmm, new
   CloudletSchedulerTimeShared());

   //vmlist.add(vm);
   vmlist.add(vm0);
   vmlist.add(vm1);
   vmlist.add(vm2);
   broker.submitVmList(vmlist);

   cloudletList = new ArrayList<Cloudlet>();

   int id = 0;

   long length = 400000;

   long fileSize = 300;

   long outputSize = 300;

   UtilizationModel utilizationModel = new UtilizationModelFull();

   Cloudlet cloudlet0 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
```

```
            utilizationModel, utilizationModel, utilizationModel);
            cloudlet0.setUserId(brokerId);
            cloudlet0.setVmId(0);
            cloudletList.add(cloudlet0);

            Cloudlet cloudlet1 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
            utilizationModel, utilizationModel, utilizationModel);
            cloudlet1.setUserId(brokerId);
            cloudlet1.setVmId(1);
            cloudletList.add(cloudlet1);

            Cloudlet cloudlet2 = new Cloudlet(id++, length, pesNumber, fileSize, outputSize,
            utilizationModel, utilizationModel, utilizationModel);
            cloudlet2.setUserId(brokerId);
            cloudlet2.setVmId(2);
            cloudletList.add(cloudlet2);

            broker.submitCloudletList(cloudletList);

            CloudSim.startSimulation();

            CloudSim.stopSimulation();

            List<Cloudlet> newList = broker.getCloudletReceivedList();
            printCloudletList(newList);

            Log.printLine("Cloudsim finished");
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
            Log.printLine("Unwanted error occured");
        }
    }

    public static Datacenter createDatacenter(String name)
    {
        List<Host> hostList = new ArrayList<Host>();

        List<Pe> peList0 = new ArrayList<Pe>();

        List<Pe> peList1 = new ArrayList<Pe>();

        List<Pe> peList2 = new ArrayList<Pe>();

        int mips = 1000;

        peList0.add(new Pe(0, new PeProvisionerSimple(mips)));
        peList1.add(new Pe(0, new PeProvisionerSimple(mips)));
```

```
        peList2.add(new Pe(0, new PeProvisionerSimple(mips)));

        int hostId = 0;
        int ram = 2048;
        long storage = 1000000;
        int bw = 10000;

        hostList.add(new Host(hostId++, new RamProvisionerSimple(ram), new
        BwProvisionerSimple(bw), storage, peList0, new VmSchedulerTimeShared(peList0)));
        hostList.add(new Host(hostId++, new RamProvisionerSimple(ram), new
        BwProvisionerSimple(bw), storage, peList1, new VmSchedulerTimeShared(peList1)));
        hostList.add(new Host(hostId++, new RamProvisionerSimple(ram), new
        BwProvisionerSimple(bw), storage, peList2, new VmSchedulerTimeShared(peList2)));

        String arch = "x86";
        String os = "Linux";
        String vmm = "Xen";
        double time_zone = 10.0;
        double cost = 3.0;
        double costPerMem = 0.05;
        double costPerStorage = 0.001;
        double costPerBw = 0.0;
        LinkedList<Storage> storageList = new LinkedList<Storage>();
        DatacenterCharacteristics characteristics = new DatacenterCharacteristics(arch, os, vmm,
        hostList, time_zone, cost, costPerMem, costPerStorage, costPerBw);

        Datacenter datacenter = null;
        try
        {
            datacenter = new Datacenter(name, characteristics, new
            VmAllocationPolicySimple(hostList), storageList, 0);
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
            return null;
        }
        return datacenter;
    }
    public static DatacenterBroker createBroker(String name)
    {
        DatacenterBroker broker = null;
        try
        {
            broker = new DatacenterBroker(name);
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
```

```
            return null;
        }
        return broker;
    }
    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 :



CONCLUSION:

In this practical, we learnt about cloudsim architecture and implemented several different scenariosusing different number of datacenters, hosts and cloudlet

PRACTICAL: 3

AIM:

Perform following using Cloud Analyst:

1. **Install a Cloud Analyst and Integrate with NetBeans. Monitor the performance of an ExistingAlgorithms given in Cloud Analyst.**

Modify or propose a new load balancing algorithm compatible with Cloud Analyst.

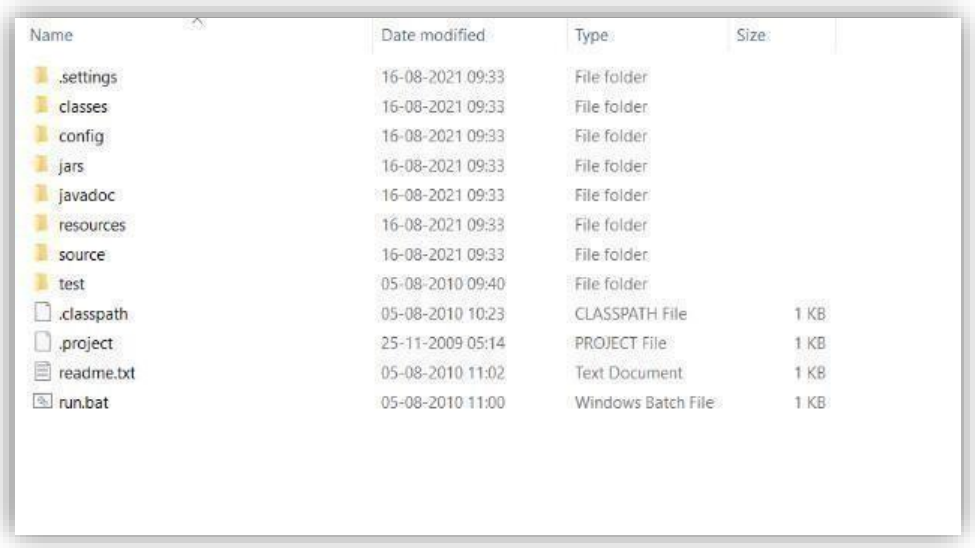
THEORY:

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.

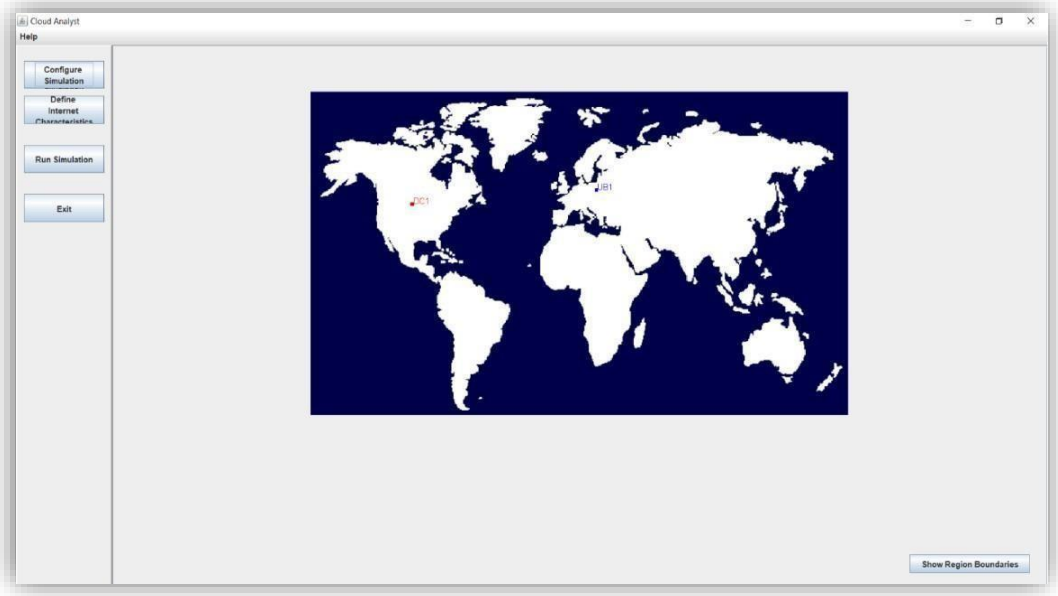
PRACTICAL:

- Download CloudAnalyst from
- **http://www.cloudbus.org/cloudsim/CloudAnalyst.zip**
- Extract Files from the Zip file which will give following folder structure.

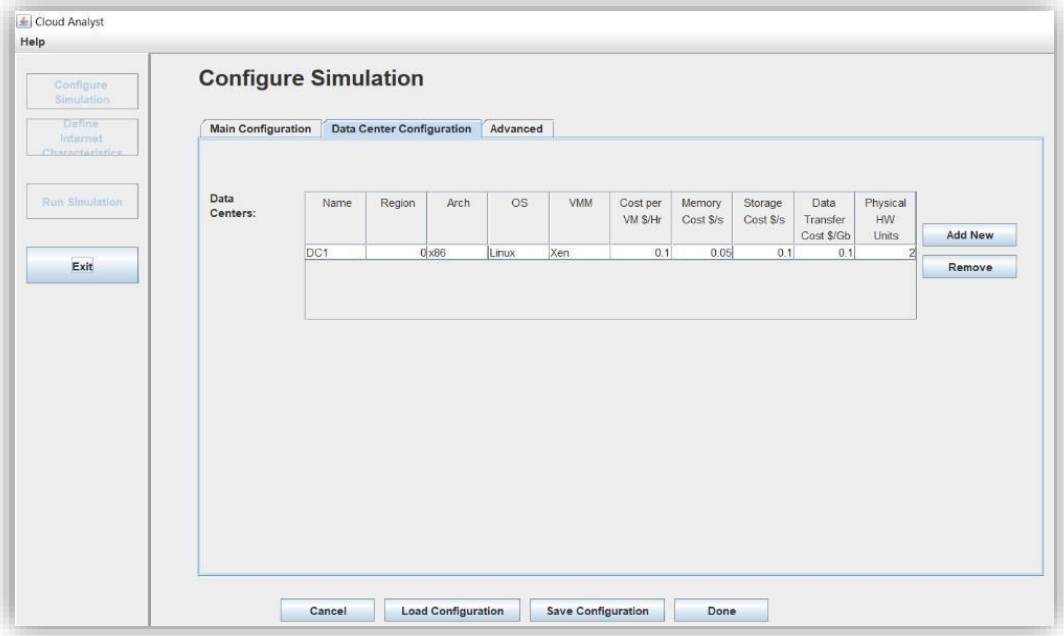


- If you want to Run from Command line then type the following command in cmd.

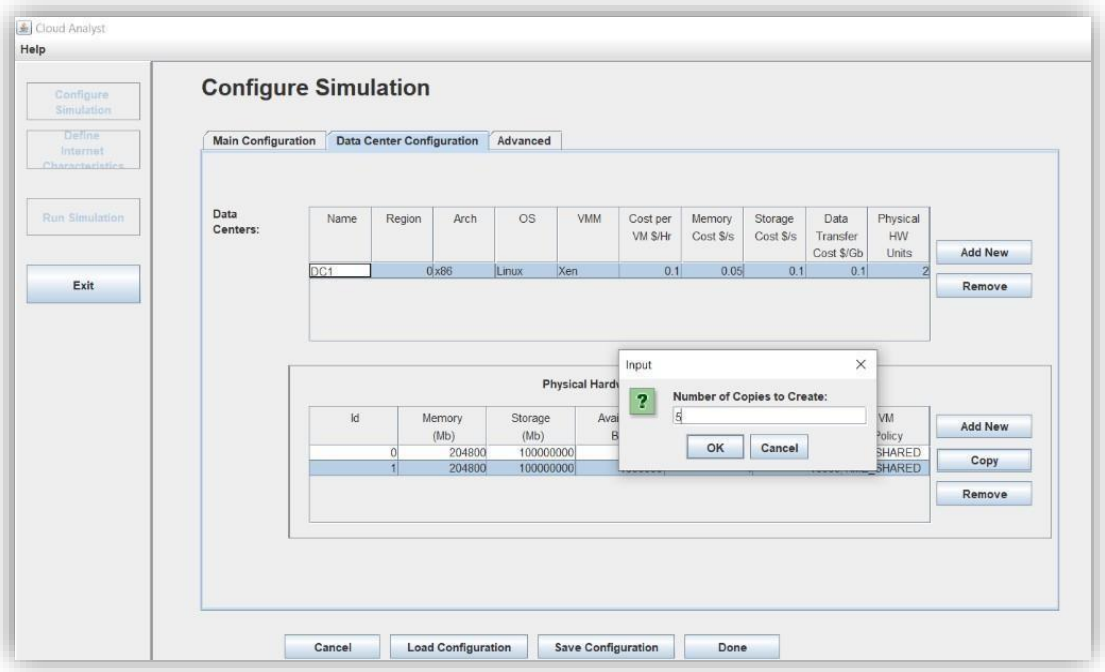
java -cp jars\simjava2.jar;jars\gridsim.jar;jars\iText-2.1.5.jar;classes;. cloudsim.ext.gui.GuiMain



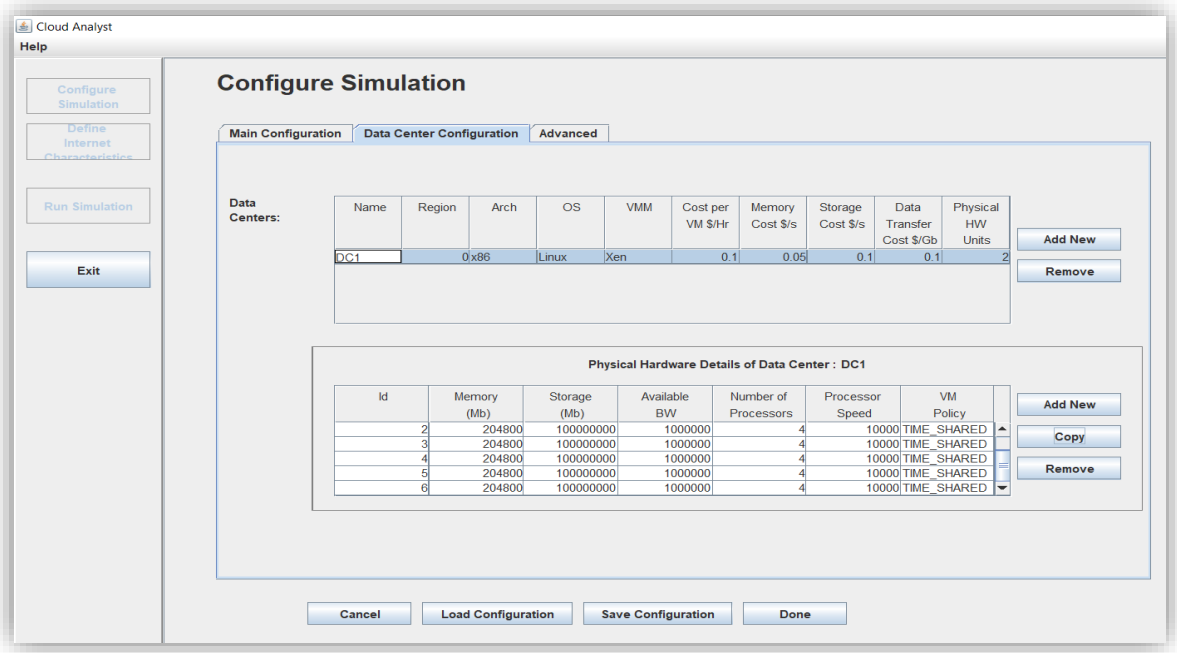
- Click on CONFIGURE Simulation.
- Here you can configure:
- Data Center Configuration where you can manage Physical Hardware Details of Data center.



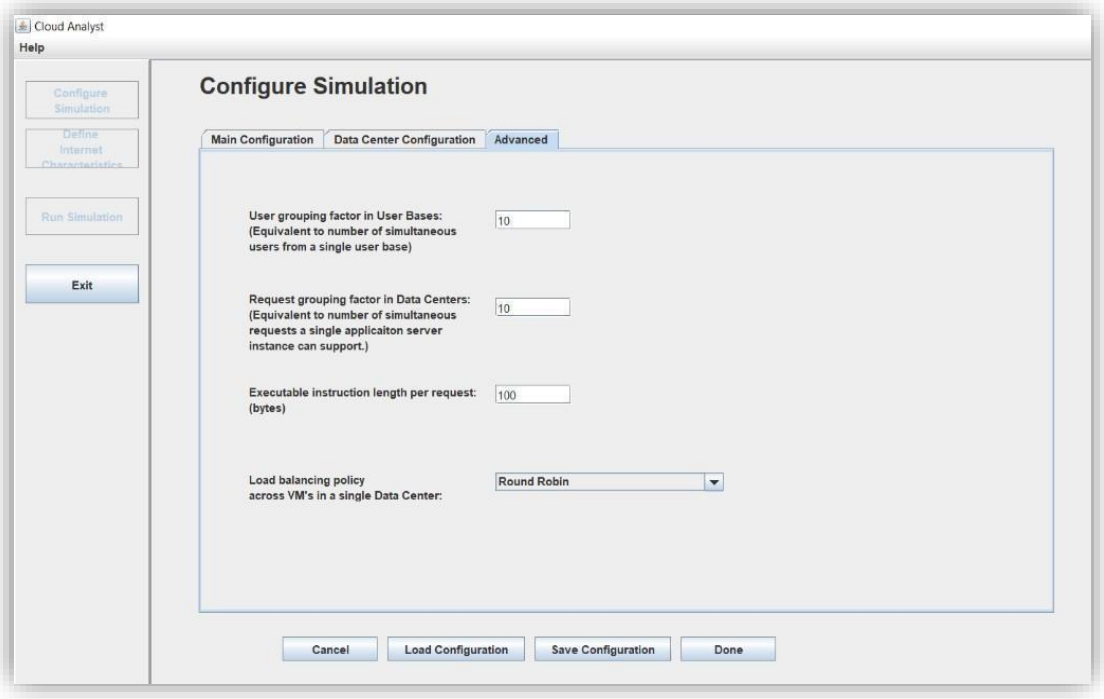
- We can double click on Datacenter to open Host information.
-
- We can click on any host and copy it as many time as we want.



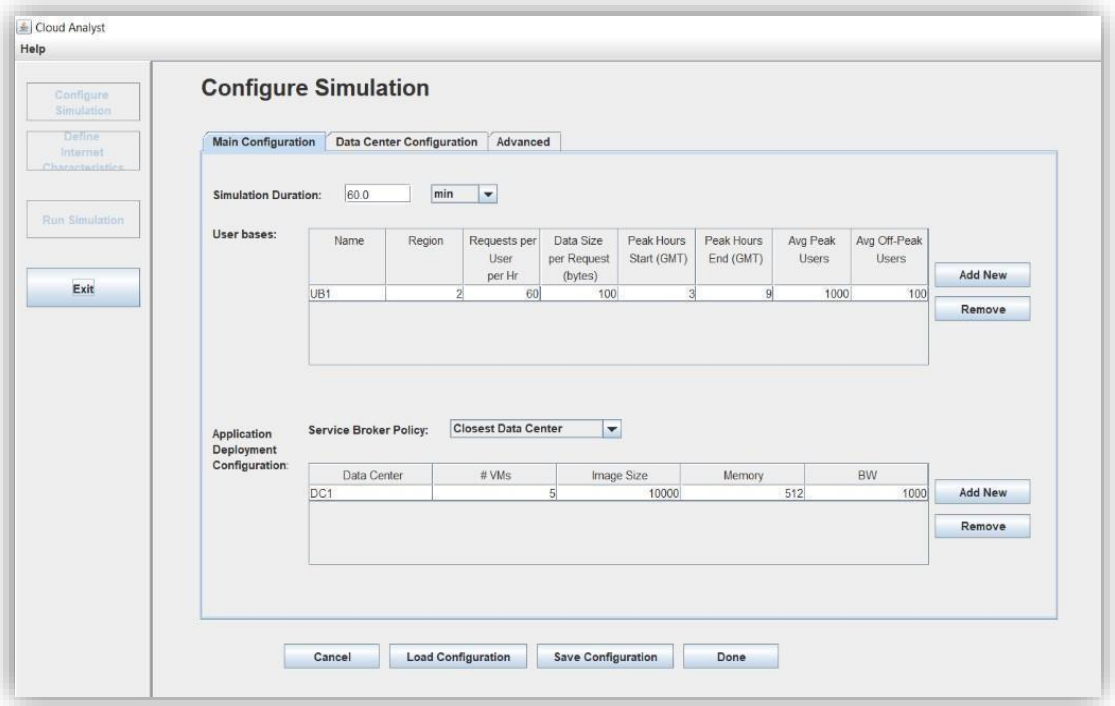
- Here we are creating 5 copies of one of them so it will give us total of 6 hosts.



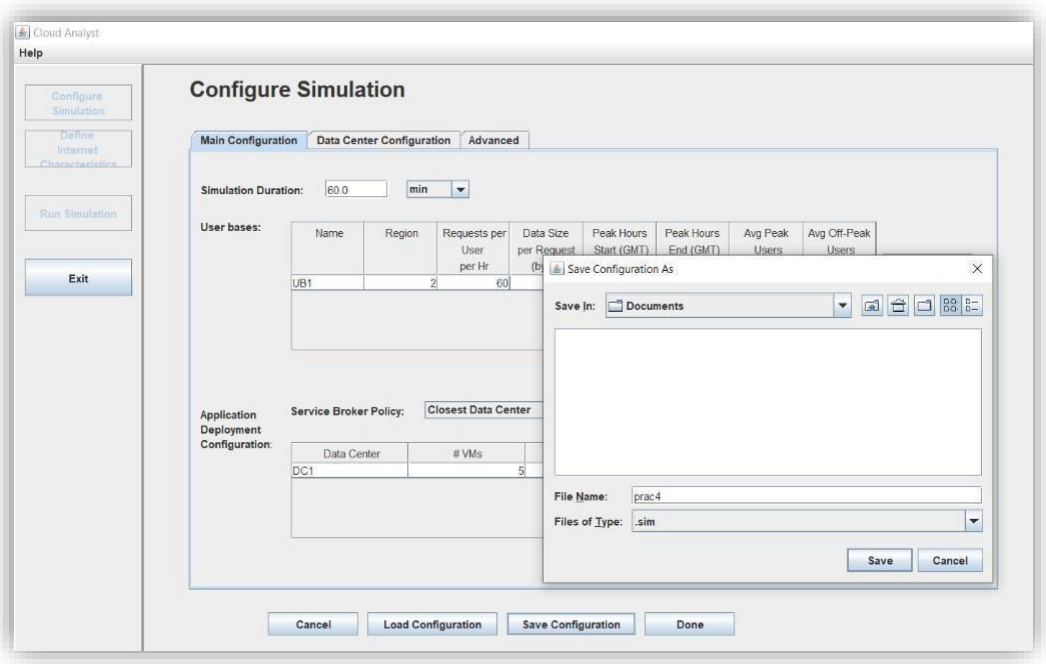
- We can also configure advanced settings.



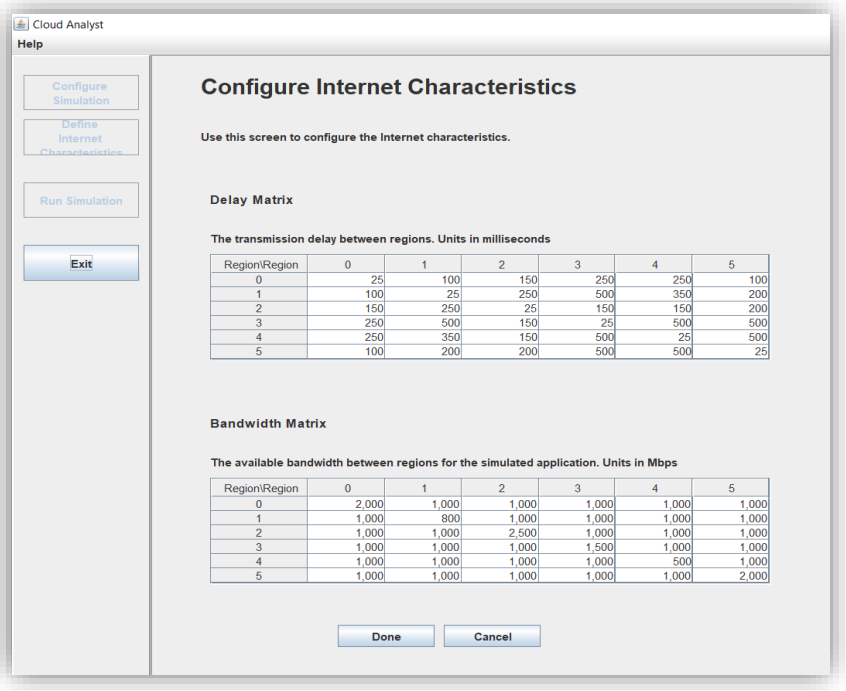
- We can also customize user base that is models a group of users and generates traffic representing the users.



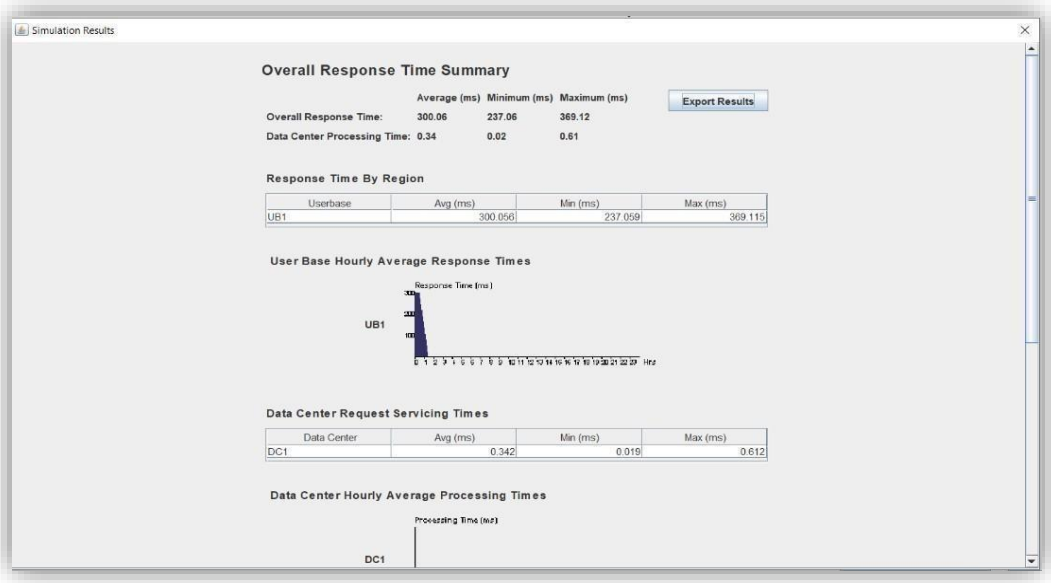
- You can Save this Configuration as well in case you want to use it later. It is stored as .sim file. XML data is generated and saved as Sim file.



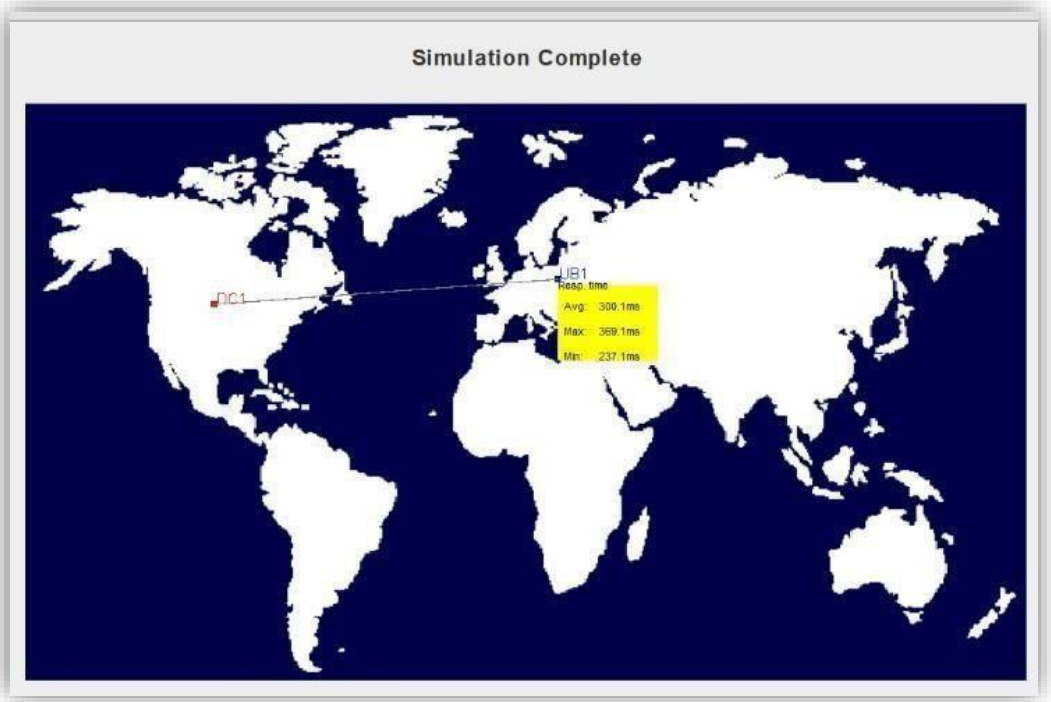
- Saved configuration can be loaded anytime easily into CloudAnalyst.
- So you need to enter data each time you want to run simulation.
- Once your are done with Configuration; click on Done!!!
- We can check bandwidth and delay between regions in "Define Internet Characteristic"



- Then we can run simulation that would give us overall report of simulation.



- We can close it.
- Main Window will give all statistics.



CONCLUSION:

In this practical, we learnt about cloudAnalyst and simulated a simple case with singledatacenter with 6 hosts and single user base.

PRACTICAL: 4

AIM:

Perform following using Google Cloud Platform:

1. **Introduction to Google Cloud.**
2. **Perform Google Cloud Hands-on Labs.**

Create and setup a Virtual Machine, GCP Essentials and Compute Engine:

1. **Qwik Start - Windows on Google Cloud Platform.**
2. **Compute Engine: Qwik Start – Windows**

THEORY:

Introduction To Google Cloud Platform

Google has been one of the leading software and technology developer in the world. Every year Google comes up with different innovations and advancement in the technological field which is brilliant and helps the people all over the world.

In the recent years, Google Cloud Platform is one of such innovations that have seen an increase in its usage because more and more people are adopting Cloud. Since there has been a great demand in the computing needs, a number of Google cloud services have been launched for global customers.

What is Google Cloud Platform?

Google cloud platform is a medium with the help of which people can easily access the cloud systems and other computing services which are developed by Google. The platform includes a wide range of services that can be used in different sectors of cloud computing, such as storage and application development. Anyone can access the Google cloud platform and use it according to their needs.

Various Elements of Google Cloud Platform

As you can understand that Google cloud platform is made up of a different set of elements which are helpful for the people in multiple ways. Here, in this section coming up, we are going to talk about various such elements which are present in the Google Cloud.

- **Google Compute Engine**: This computing engine has been introduced with the IaaSservice by Google which effectively provides the VMs similar to Amazon EC2.

- **Google Cloud App Engine:** The app engine has the PaaS service for the correct hostingapplications directly. This is a very powerful and important platform which helps to develop mobile and different web applications.

- **Google Cloud Container Engine**: this particular element is helpful because it allows the user to run the docker containers present up on the Google Cloud Platform that is effectively triggered by Kubernetes.

- **Google Cloud Storage:** the ability to store data and important resources on the cloud platform is very important. Google cloud platform has been popular with the storage facilities and have allows the users to back up or store data on the cloud servers whichcan be accessed from anywhere at any time.

- **Google BigQuery Service:** the Google BigQuery Service is an efficient data analysis service which enables the users to analyze their business for Big data. It also has a highlevel of storage facility which can hold up to terabytes of storage.

- **Google Cloud Dataflow:** the cloud data flow allows the users to manage consistent parallel data-processing pipelines. It helps to manage the lifecycle of Google Computeservers of the pipelines that are being processed.

- **Google Cloud Job Discovery:** the Google Cloud Platform is also a great source for jobsearch, career options etc. The advanced search engine and machine learning capabilities make it possible to find out different ways of finding jobs and business opportunities.

- **Google Cloud Test Lab:** this service provided by Google allows the users to test theirapps with the help of physical and virtual devices present in the cloud. The various instrumentation tests and robotic tests allow the users to get more insights about their applications.

- **Google Cloud Endpoints:** this particular feature helps the users to develop andmaintain secured application program interface running on the Google Cloud Platform.

- **Google Cloud Machine Learning Engine:** as the name suggests, this element presentin Google Cloud helps the users to develop models and structures which enables the users to concentrate on Machine learning abilities and framework.

Google Cloud Platform Services	
Category	Services
	Compute Engine App Engine
Compute	Kubernetes Engine Cloud Functions
Storage	Cloud Storage Persistent Disk Cloud Memorystore Cloud Firestore Cloud Storage for Firebase Cloud Filestore
Databases	Cloud SQL Cloud BigTable Cloud Spanner Cloud Datastore Firebase Realtime Database
Migration	Data Transfer Transfer Appliance Cloud Storage Transfer Service BigQuery Data Transfer Service
Networking	Virtual Private Cloud (VPC) Cloud Load Balancing Cloud Armor Cloud CDN Cloud Interconnect Cloud DNS Network Service Tiers

Lab fundamentals

Features and components

Regardless of topic or expertise level, all labs share a common interface. The lab that you're taking should look similar to this:

Start Lab (button)

Clicking this button creates a temporary Google Cloud environment, with all the necessary services and credentials enabled, so you can get hands-on practice with the lab's material. This also starts a countdown timer.

Credit

The price of a lab. 1 Credit is *usually* equivalent to 1 US dollar (discounts are available when you purchase credits in bulk). Some introductory-level labs (like this one) are free. The more specialized labs cost more because they involve heavier computing tasks and demand more Google Cloud resources.

Time

Specifies the amount of time you have to complete a lab. When you click the Start Lab button, the timer will count down until it reaches 00:00:00. When it does, your temporary Google Cloud environment and resources are deleted. Ample time is given to complete a lab, but make sure you don't work on something else while a lab is running: you risk losing all of your hard work!

Score

Many labs include a score. This feature is called "activity tracking" and ensures that you complete specified steps in a lab. To pass a lab with activity tracking, you need to complete all the steps *in order*. Only then will you receive completion credit.

Reading and following instructions

This browser tab contains the lab instructions. When you start a lab, the lab environment in this case, the Google Cloud Console user interface opens in a new browser tab. You will need to

switch between the two browser tabs to read the instructions and then perform the tasks. Depending on your physical computer setup, you could also move the two tabs to separate monitors.

Accessing the Cloud Console

Start the lab

- Now that you understand the key features and components of a lab, click **Start Lab**.

It may take a moment for the Google Cloud environment and credentials to spin up. When the timer starts counting down and the Start Lab button changes to a red End Lab button, everything is in place and you're ready to sign in to the Cloud Console.

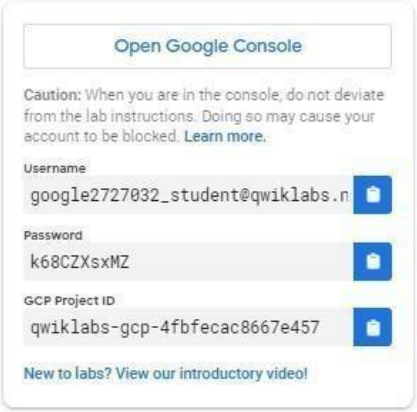
Task 2. View all projects

You actually have access to more than one Google Cloud project. In fact, in some labs you may be given more than one project in order to complete the assigned tasks.

1. In the Google Cloud Console title bar, next to your project name, click the drop-down menu.
2. In the **Select a project** dialog, click **All**. The resulting list of projects includes a "Qwiklabs Resources" project.

Connection Details pane

Now that your lab instance is up and running, look at the left pane. It contains an Open Google Console button, credentials (username and password), and a Project ID field.



Open Google Console

This button opens the Cloud Console: the web console and central development hub for GoogleCloud. You will do the majority of your work in Google Cloud from this interface.

Project ID

A Google Cloud project is an organizing entity for your Google Cloud resources. It

often contains resources and services; for example, it may hold a pool of virtual machines, a set ofdatabases, and a network that connects them together. Projects also contain settings and permissions, which specify security rules and who has access to what resources.
A *Project ID* is a unique identifier that is used to link Google Cloud resources and APIs to your specific project. Project IDs are unique across Google Cloud: there can be only one **qwiklabs-gcp-xxx. ..**, which makes it globally identifiable.

Username and Password

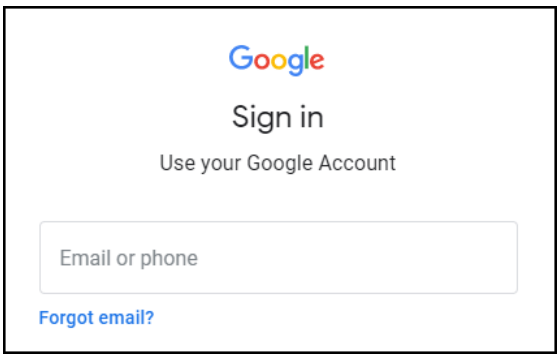
These credentials represent an identity in the Cloud Identity and Access Management (Cloud IAM) service. This identity has access permissions (a role or roles) that allow you to work with Google Cloud resources in the project you've been allocated. These credentials are *temporary* and will only work for the access time of the lab. When the timer reaches 00:00:00, you will no longer have access to your Google Cloud project with those credentials.

Task 1. Sign in to Google Cloud

Now that you have a better understanding of the Connection Details pane, use its contents to sign in to the Cloud Console.

1. Click **Open Google Console**.

This opens the Google Cloud sign-in page in a new browser tab.



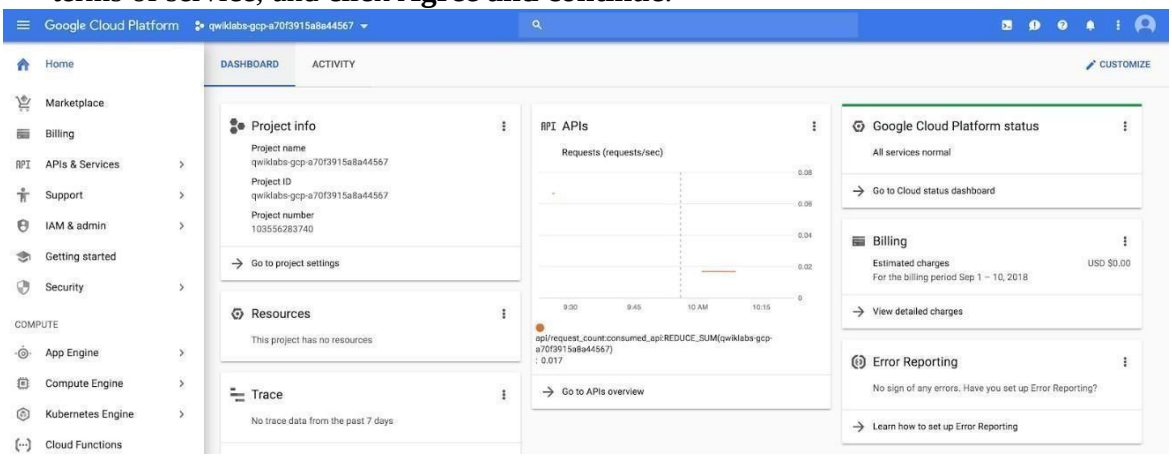
If the **Choose an account** page opens, click **Use Another Account**.

2. Copy the **Username** from the Connection Details pane, paste it in the **Email or phone** field, and click **Next**.

3. Click **I understand** to indicate your acknowledgement of Google's terms of service and privacy policy.

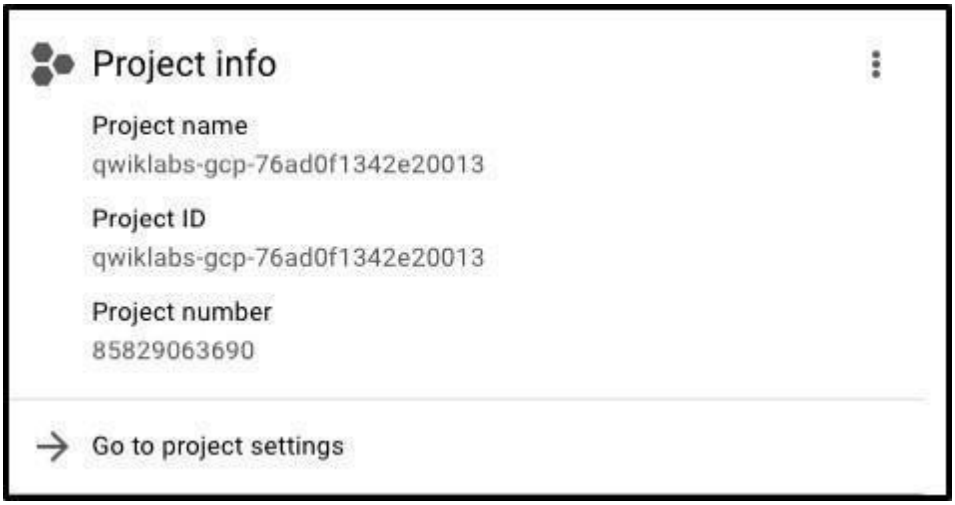
4. On the **Protect your account** page, click **Confirm**.

5. On the **Welcome student!** page, check **Terms of Service** to agree to Google Cloud's terms of service, and click **Agree and continue**.



Projects in the Cloud Console

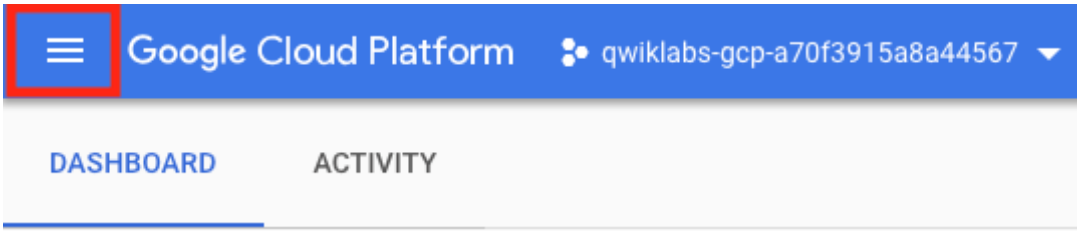
Google Cloud projects were explained in the section about the contents of the Connection Details pane. Here's the definition once again:



Your project has a *name*, *ID*, and *number*. These identifiers are frequently used when interacting with Google Cloud services. You are working with one project to get experience with a specific service or feature of Google Cloud.

Navigation menu and services

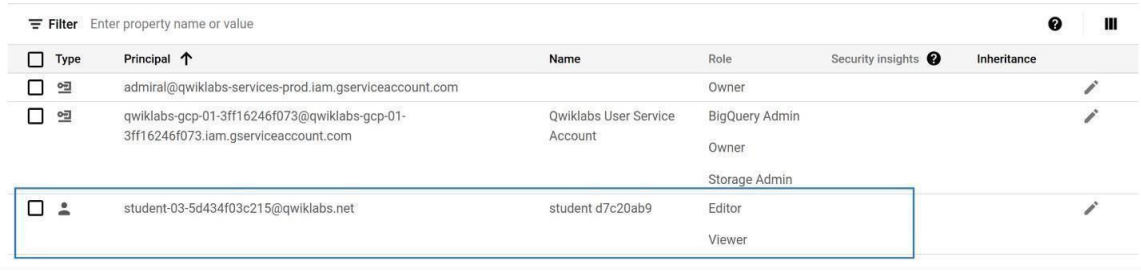
The Google Cloud Console title bar also contains a button labeled with a three-line icon:



Task 3. View your roles and permissions

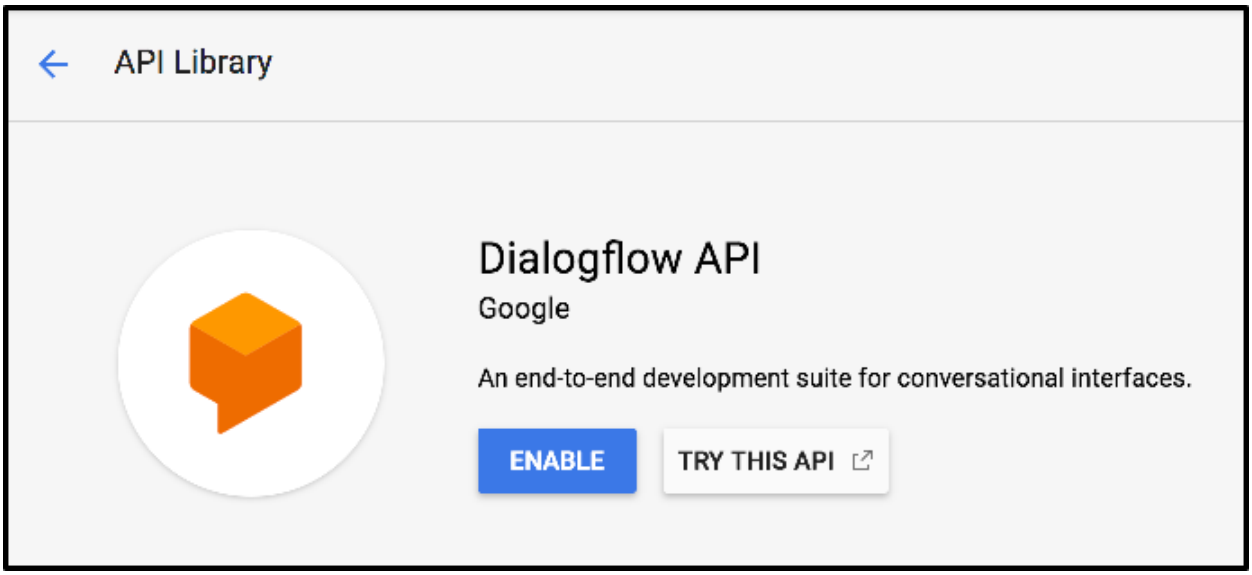
1. On the **Navigation menu** (☰), click **IAM & Admin**. This opens a page that contains a list of users and specifies permissions and roles granted to specific accounts.

2. Find the "@qwiklabs" username you signed in with:



Task 4. View available APIs

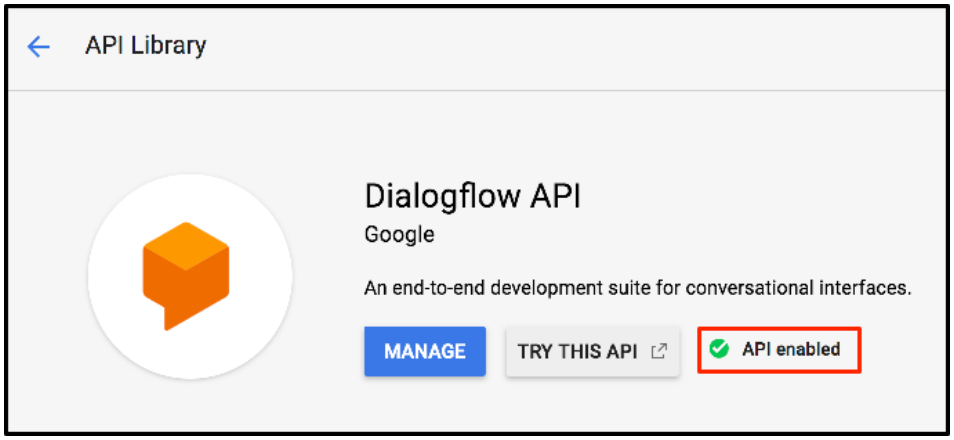
1. On the **Navigation menu** (☰), click **APIs & Services** > **Library**. The left pane, under the header **CATEGORY**, displays the different categories available.
2. In the API search bar, type **Dialogflow**, and then click **Dialogflow API**. The Dialogflow description page opens.



The Dialogflow API allows you to build rich conversational applications (e.g., for Google Assistant) without having to understand the underlying machine learning and natural language schema.

3. Click **Enable**.

4. Click the back button in your browser to verify that the API is now enabled.



5. Click **Try this API**. A new browser tab displays documentation for the Dialogflow API. Explore this information, and close the tab when you're finished.

6. To return to the main page of the Cloud Console, on the **Navigation menu**, click **Cloud overview**.

Ending your lab

Now that you're finished with the lab, click **End Lab** and then click **Submit** to confirm it.

Create a virtual machine instance

1. In the Cloud Console, on the **Navigation menu** ☰, click **Compute Engine > VM instances**, and then click **Create Instance**.

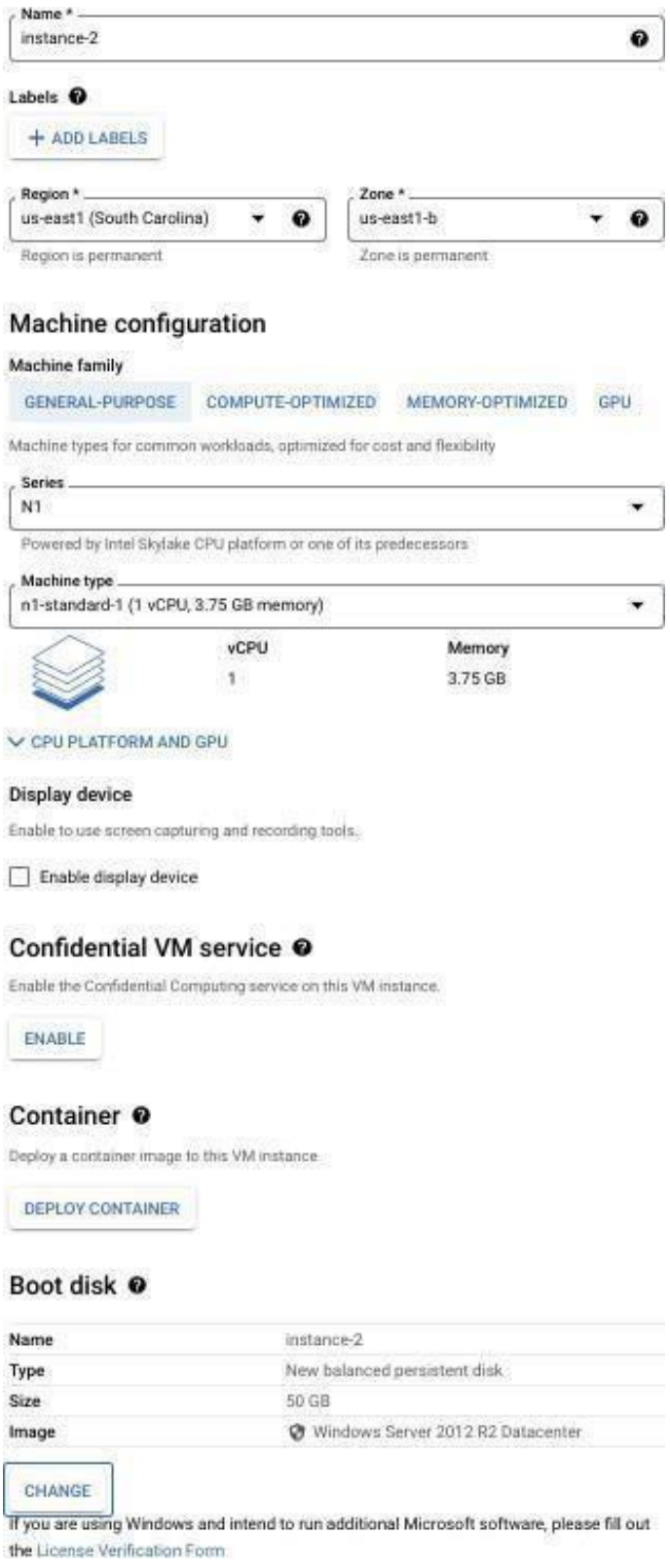
2. Select region **us east1** and zone **us east1-b**.

3. In the **Machine configuration** section, for **Series** select **N1**.

4. In the **Boot disk** section, click **Change** to begin configuring your boot disk.

5. Under **Operating system** select **Windows Server** and under **Version** select **Windows Server 2012 R2 Datacenter**, and then click **Select**. Leave all other settings as their defaults.

6. Click **Create** to create the instance.


Name *
instance-2
Labels
+ ADD LABELS
Region *
us-east1 (South Carolina)
Region is permanent
Zone *
us-east1-b
Zone is permanent
Machine configuration
Machine family
GENERAL-PURPOSE
COMPUTE-OPTIMIZED
MEMORY-OPTIMIZED
GPU
Machine types for common workloads, optimized for cost and flexibility
Series
N1
Powered by Intel Skylake CPU platform or one of its predecessors
Machine type
n1-standard-1 (1 vCPU, 3.75 GB memory)
vCPU
1
Memory
3.75 GB
CPU PLATFORM AND GPU
Display device
Enable to use screen capturing and recording tools.
Enable display device
Confidential VM service
Enable the Confidential Computing service on this VM instance.
ENABLE
Container
Deploy a container image to this VM instance.
DEPLOY CONTAINER
Boot disk
Name
instance-2
Type
New balanced persistent disk
Size
50 GB
Image
Windows Server 2012 R2 Datacenter
CHANGE
If you are using Windows and intend to run additional Microsoft software, please fill out the License Verification Form

Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** at the top of the Google Cloud console.
2. Click **Continue**.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your **PROJECT_ID**. The output contains a line that declares the **PROJECT_ID** for this session:

```
Your Cloud Platform project in this session is set to YOUR_PROJECT_ID
```

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

3. (Optional) You can list the active account name with this command:

gcloud auth list
Copied!

content_copy

Output:

```
ACTIVE: *
ACCOUNT: student-01-xxxxxxxxxxxx@qwiklabs.net
To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

(Optional) You can list the project ID with this command:

gcloud config list project
Copied!

content_copy

Output:

```
[core]
project = <project_ID>
```

Example output:

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

Remote Desktop (RDP) into the Windows Server

Test the status of Windows Startup

After a short time, the Windows Server instance will be provisioned and listed on the VM Instances page with a green status icon .

However the server instance may not yet be ready to accept RDP connections, as it takes a while for all the OS components to initialize.

To see whether the server instance is ready for an RDP connection, run the following command at your Cloud Shell terminal command line:

gcloud compute instances get-serial-port-output instance-1
Copied!

content_copy

If prompted, type **n** and press **Enter**.

Repeat the command until you see the following in the command output, which tells you that the OS components have initialized and the Windows Server is ready to accept your RDP connection (attempt in the next step).

Instance setup finished. instance-1 is ready to use.

Copied!

content_copy

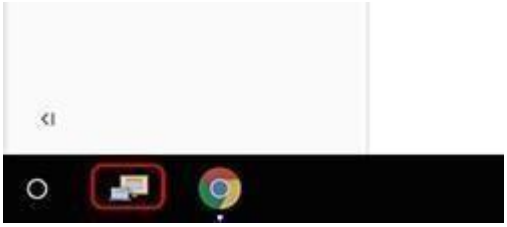
RDP into the Windows Server

To set a password for logging into the RDP, run the following command in Cloud Shell terminal and replace [instance] with the VM Instance that you have created and set [username] as **admin**.

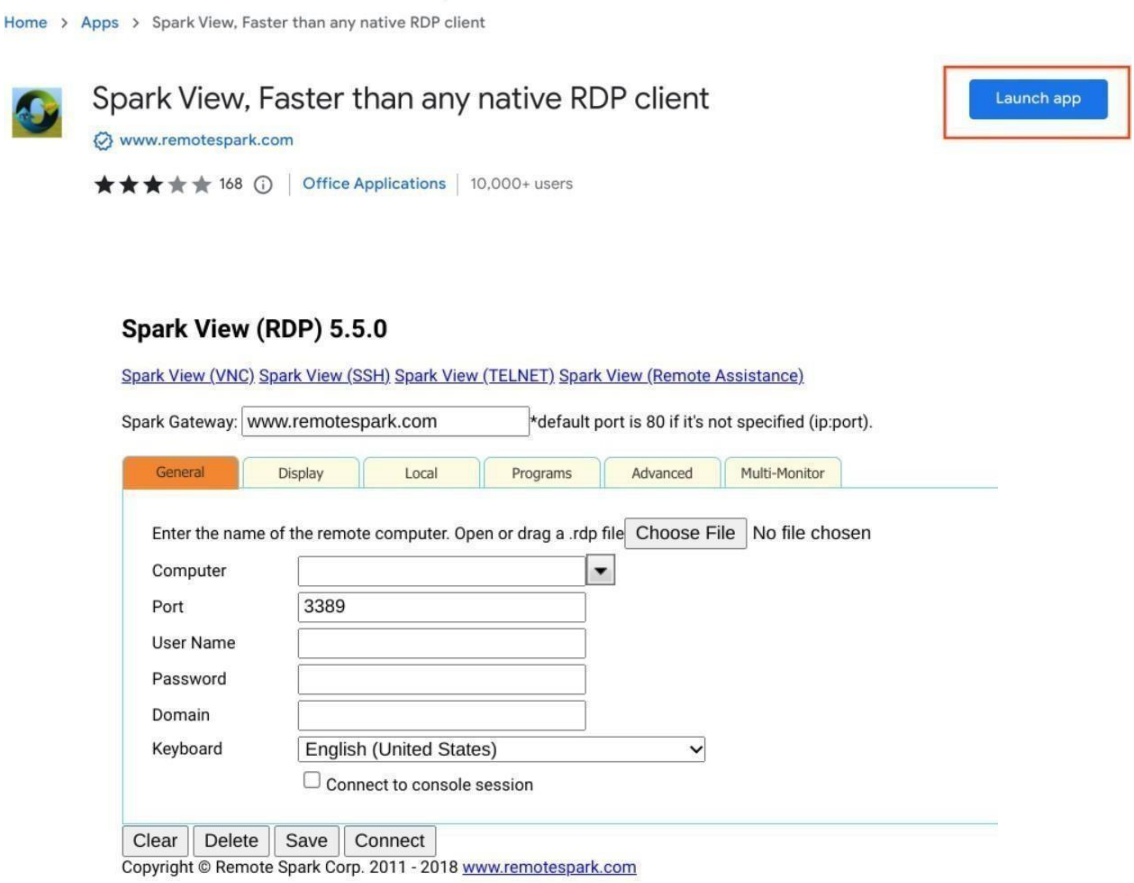
gcloud compute reset-windows-password [instance] --zone us-east1-b --user [username]
Copied!

content_copy

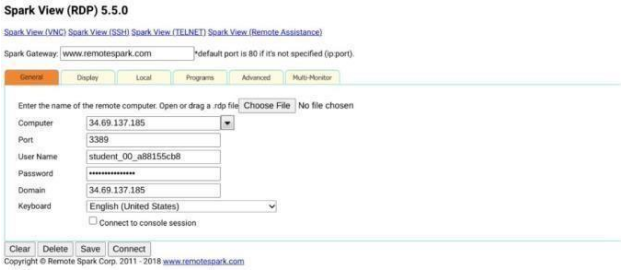
If asked Would you like to set or reset the password for [admin] (Y/n)?, enter **Y**.
If you are using a Chromebook or other machine at a Google Cloud event there is likely an RDP app already installed on the computer. Click the icon as below, if it is present, in the lower left corner of the screen and enter the external IP of your VM.



If you are not on Windows but using Chrome, you can connect to your server through RDP directly from the browser using the Spark View extension. Click on **Add to Chrome**. Then, click **Launch app** button.



Add your VM instance's External IP as your Domain. Click Connect to confirm you want to connect.



Once logged in, you should see the Windows desktop!

Copy and paste with the RDP client

Once you are securely logged in to your instance, you may find yourself copying and pasting commands from the lab manual.

- To paste, hold the **CTRL-V** keys (if you are a Mac user, using **CMND-V** will not work.)
- If you are in a Powershell window, be sure that you have clicked in to the window or else the paste shortcut won't work.
- If you are pasting into putty, **right click**.

CONCLUSION:

In this practical, we learnt about Basic Of Google cloud Platform As well ashow to make a instance of windows in cloud.

PRACTICAL: 5

AIM:

Introduction to cloud Shell and gcloud on Google Cloud. Perform Following task:
Practice using gcloud commands.
Connect to compute services hosted on Google Cloud.

THEORY:

Task 1. Configure your environment

In this section, you'll learn about aspects of the development environment that you can adjust.

1. Set the region to

   Run the command: gcloud config set compute/region asia-east1-a

```
API [compute.googleapis.com] not enabled on project [104089128095]. Would you like to enable and retry
minutes)? (y/N)?  y

Enabling service [compute.googleapis.com] on project [104089128095]...
Enabling service [compute.googleapis.com] on project [104089128095]...
WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
```

2. To view the project region setting, run the following command:

   Run the command: gcloud config get-value compute/region

```
Your active configuration is: [cloudshell-25756]
asia-east1-a
```

3. Set the zone to_____:

   Run the command: gcloud config set compute/zone asia-east1-a

```
API [compute.googleapis.com] not enabled on project [104089128095]. Would you like to
enable and retry (this will take a few minutes)? (y/N)?  t
Please enter 'y' or 'n':  y

Enabling service [compute.googleapis.com] on project [104089128095]...
WARNING: Property validation for compute/zone was skipped.
Updated property [compute/zone].
```

4. To view the project zone setting, run the following command:

   gcloud config get-value compute/zone

```
Your active configuration is: [cloudshell-25756]
asia-east1-a
```

Finding project information

1. Copy your project ID to your clipboard or text editor. The project ID is listed in 2 places:
   - In the Cloud Console, on the Dashboard, under **Project info**. (Click **Navigation menu** (☰), and then click **Cloud overview > Dashboard**.)
   - On the lab tab near your username and password.
2. In Cloud Shell, run the following gcloud command, to view the project id for your project:

gcloud config get-value project

```
Your active configuration is: [cloudshell-9733]
astral-volt-359709
```

3. In Cloud Shell, run the following gcloud command to view details about the project:

gcloud compute project-info describe --project $(gcloud config get-value project)

```
Your active configuration is: [cloudshell-14852]
commonInstanceMetadata:
  fingerprint: 6WmySwYh5e0=
  kind: compute#metadata
creationTimestamp: '2022-09-06T00:27:20.440-07:00'
defaultNetworkTier: PREMIUM
defaultServiceAccount: 278829130276-compute@developer.gserviceaccount.com
id: '3855182035555141815'
kind: compute#project
name: deft-grammar-361707
quotas:
- limit: 1000.0
  metric: SNAPSHOTS
  usage: 0.0
- limit: 5.0
  metric: NETWORKS
  usage: 1.0
- limit: 100.0
  metric: FIREWALLS
  usage: 4.0
- limit: 100.0
  metric: IMAGES
  usage: 0.0
- limit: 8.0
  metric: STATIC_ADDRESSES
  usage: 0.0
- limit: 200.0
  metric: ROUTES
  usage: 1.0
- limit: 15.0
  metric: FORWARDING_RULES
  usage: 0.0
- limit: 50.0
  metric: TARGET_POOLS
  usage: 0.0
- limit: 50.0
  metric: HEALTH_CHECKS
  usage: 0.0
- limit: 8.0
  metric: IN_USE_ADDRESSES
  usage: 0.0
- limit: 50.0
  metric: TARGET_INSTANCES
  usage: 0.0
- limit: 10.0
  metric: TARGET_HTTP_PROXIES
  usage: 0.0
```

Setting environment variables

Environment variables define your environment and help save time when you write scripts that contain APIs or executables.

1. Create an environment variable to store your Project ID:

Run the command: export PROJECT_ID=$(gcloud config get-value project)

```
Your active configuration is: [cloudshell-9733]
```

2. Create an environment variable to store your Zone:

Run the command: export ZONE=$(gcloud config get-value compute/zone)

```
Your active configuration is: [cloudshell-9733]
```

3. To verify that your variables were set properly, run the following commands

Run the command: echo –e "PROJECT ID: $PROJECT_ID\nZONE: $ZONE"

```
PROJECT ID: astral-volt-359709
ZONE: asia-east1-a
```

Creating a virtual machine with the gcloud tool

Use the gcloud tool to create a new virtual machine (VM) instance.

Run the command: gcloud compute instances create gcelab2 –machine-type e2-medium –zone $ZONE

```
Created [https://www.googleapis.com/compute/v1/projects/deft-grammar-361707/zones/asia-east1-a/instances/gcelab2].
NAME: gcelab2
ZONE: asia-east1-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.140.0.2
EXTERNAL_IP: 34.81.74.145
STATUS: RUNNING
```

- gcloud compute allows you to manage your Compute Engine resources in a format that's simpler than the Compute Engine API.
- instances create creates a new instance.
- gcelab2 is the name of the VM.
- The --machine-type flag specifies the machine type as *e2-medium*.
- The --zone flag specifies where the VM is created.
- If you omit the --zone flag, the gcloud tool can infer your desired zone based on your default properties. Other required instance settings, such as machine type and image, are set to default values if not specified in the create command.

Test completed task

Click **Check my progress** to verify your performed task. If you have successfully created a virtual machine with the gcloud tool, an assessment score is displayed.

Create a virtual machine with gcloud

Check my progress

- To open help for the create command, run the following command:

gcloud compute instances create --help
Copied!

content_copy

Note: Press **ENTER** or the spacebar to scroll through the help content. To exit the content, type **Q**.

Task 2. Filtering command line output

The gcloud CLI is a powerful tool for working at the command line. You may want specific information to be displayed.

1. List the compute instance available in the project:

   gcloud compute instances list

```
NAME: gcelab2
ZONE: asia-east1-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.140.0.2
EXTERNAL_IP: 34.81.74.145
STATUS: RUNNING
```

2. List the gcelab2 virtual machine:

   gcloud compute instance list --filter="name=('gcelab2')"

```
NAME: gcelab2
ZONE: asia-east1-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.140.0.2
EXTERNAL_IP: 34.81.74.145
STATUS: RUNNING
```

3. List the Firewall rules in the project:

   gcloud compute firewall-rules list

```
NAME: default-allow-icmp
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: icmp
DENY:
DISABLED: False

NAME: default-allow-internal
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:0-65535,udp:0-65535,icmp
DENY:
DISABLED: False

NAME: default-allow-rdp
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:3389
DENY:
DISABLED: False

NAME: default-allow-ssh
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:22
DENY:
DISABLED: False

To show all fields of the firewall, please show in JSON format: --format=json
To show all fields in table format, please see the examples in --help.
```

4. List the Firewall rules for the default network:

   gcloud compute firewall-rules list --filter=”network=’default’”

```
NAME: default-allow-icmp
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: icmp
DENY:
DISABLED: False

NAME: default-allow-internal
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:0-65535,udp:0-65535,icmp
DENY:
DISABLED: False

NAME: default-allow-rdp
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:3389
DENY:
DISABLED: False

NAME: default-allow-ssh
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:22
DENY:
DISABLED: False

To show all fields of the firewall, please show in JSON format: --format=json
To show all fields in table format, please see the examples in --help.
```

5. List the Firewall rules for the default network where the allow rule matches an ICMP rule:

   gcloud compute firewall-rules list –filter=”NETWORK: ‘defaule’ AND ALLOW: ‘icmp’”

```
NAME: default-allow-icmp
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: icmp
DENY:
DISABLED: False

NAME: default-allow-internal
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 65534
ALLOW: tcp:0-65535,udp:0-65535,icmp
DENY:
DISABLED: False

To show all fields of the firewall, please show in JSON format: --format=json
To show all fields in table format, please see the examples in --help.
```

Task 3. Connecting to your VM instance

gcloud compute makes connecting to your instances easy. The gcloud compute ssh command provides a wrapper around SSH, which takes care of authentication and the mapping of instance names to IP addresses.

1. To connect to your VM with SSH, run the following command:
   gcloud compute ssh gcelab2 --zone $ZONE

   Output**:**

```
WARNING: The public SSH key file for gcloud does not exist.
WARNING: The private SSH key file for gcloud does not exist.
WARNING: You do not have an SSH key for gcloud.
WARNING: [/usr/bin/ssh-keygen] will be executed to generate a key.
This tool needs to create the directory
[/home/gcpstaging306_student/.ssh] before being able to generate SSH Keys.
Do you want to continue? (Y/n)
```

2. To continue, type **Y**.

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase)
```

3. To leave the passphrase empty, press **ENTER** twice.
4. Install nginx web server on to virtual machine:

   sudo apt install -y nginx

5. You don't need to do anything here, so to disconnect from SSH and exit the remote shell, run the following command:

   exit

   You should be back at your project's command prompt.

Task 4. Updating the Firewall

When using compute resources such as virtual machines, it's important to understand the associated firewall rules.

1. List the firewall rules for the project:

   gcloud compute firewall-rules list

Output:

```
NAME                    NETWORK     DIRECTION  PRIORITY  ALLOW
DENY  DISABLED
default-allow-icmp      default     INGRESS    65534     icmp
False
default-allow-internal  default     INGRESS    65534     tcp:0-65535,udp:0-
65535,icmp  False
default-allow-rdp       default     INGRESS    65534     tcp:3389
False
default-allow-ssh       default     INGRESS    65534     tcp:22
False
```

```
dev-net-allow-ssh          dev-network  INGRESS    1000      tcp:22
False
serverless-to-vpc-connector  dev-network  INGRESS    1000      icmp,udp:665-
666,tcp:667          False
vpc-connector-egress        dev-network  INGRESS    1000      icmp,udp,tcp
False
vpc-connector-health-check   dev-network  INGRESS    1000      tcp:667
False
vpc-connector-to-serverless  dev-network  EGRESS     1000      icmp,udp:665-
666,tcp:667          False
```

2. Try to access the nginx service running on the gcelab2 virtual machine.
3. Add a tag to the virtual machine:

   gcloud compute instances add-tags gcelab2 --tags http-server,https-server

4. Update the firewall rule to allow:

   gcloud compute firewall-rules create default-allow-http --direction=INGRESS --priority=1000 --network=default --action=ALLOW --rules=tcp:80 --source- ranges=0.0.0.0/0 --target-tags=http-server

5. List the firewall rules for the project:

   gcloud compute firewall-rules list --filter=ALLOW:'80'

   Output**:**

```
NAME              NETWORK  DIRECTION  PRIORITY  ALLOW   DENY
DISABLED
default-allow-http  default  INGRESS    1000      tcp:80        False
```

6. Verify communication is possible for http to the virtual machine:

   curl http://$(gcloud compute instances list --filter=name:gcelab2 --format='value(EXTERNAL_IP)')

Task 5. Viewing the system logs

Viewing logs is essential to understanding the working of your project. Use gcloud to access the different logs available on Google Cloud.

1. View the available logs on the system:

   gcloud logging logs list

   Output**:**

```
NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/GCEGuestAgent
NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/OSConfigAgent
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/autoscaler.googleapis.com%2Fstatus_change
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/cloudaudit.googleapis.com%2Factivity
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/cloudaudit.googleapis.com%2Fdata_access
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/cloudaudit.googleapis.com%2Fsystem_event
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/compute.googleapis.com%2Fautoscaler
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/compute.googleapis.com%2Finstance_group_manager_events
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/compute.googleapis.com%2Fshielded_vm_integrity
NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/run.googleapis.com%2Fstderr
NAME: projects/qwiklabs-gcp-01-4b75909db302/logs/run.googleapis.com%2Fstdout
```

2. View the logs that relate to compute resources:

   gcloud logging logs list --filter="compute"

   Output:

```
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/compute.googleapis.com%2Fautoscaler
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/compute.googleapis.com%2Finstance_group_manager_events
NAME: projects/qwiklabs-gcp-01-
4b75909db302/logs/compute.googleapis.com%2Fshielded_vm_integrity
```

3. Read the logs related to the resource type of gce_instance:

   gcloud logging read "resource.type=gce_instance" --limit 5

4. Read the logs for a specific viítual machine:

   gcloud logging read "resource.type=gce_instance AND labels.instance_name='gcelab2'" --limit 5

CONCLUSION:

In this Practical We are Learn How to run or use the google cloud shell.

PRACTICAL: 6

AIM:

Perform Cluster orchestration with Google Kubernetes Engine.

THEORY:

Active Cloud Shell:

1. Click **Activate Cloud Shell** at the top of the Google Cloud console.

2. Click **Continue**.

Task 1: Set a default compute zone

1. **Set the default computer zone**
   a. gcloud config set compute/region us-west4
2. **Set the default compute zone**
   a. gcloud config set compute/zone us-west4-c

```
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ gcloud config list project
[core]
project = qwiklabs-gcp-02-a0271246c06e

Your active configuration is: [cloudshell-6294]
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ gcloud config set compute/region us-west4
Updated property [compute/region].
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ gcloud config set compute/zone us-west4-c
Updated property [compute/zone].
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$
```

Task 2: Create a GKE Cluster

A cluster consists of at least one **cluster master** machine and multiple worker machines called **nodes**. Nodes are Compute Engine virtual machine (VM) instances that run the Kubernetes processes necessary to make them part of the cluster.

1. Create a cluster:
   a. gcloud container clusters create --machine-type=e2-medium --zone=us-west4-c lab-cluster

```
Updated property [compute/zone].
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ gcloud container clusters create --machine-type=e2-medium --zone=us-west4-c lab-cluste
r
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based clusters,
 please pass the `--no-enable-ip-alias` flag
Default change: During creation of nodepools or autoscaling configuration changes for cluster versions greater than 1.24.1-gke.800 a default location polic
y is applied. For Spot and PVM it defaults to ANY, and for all other VM kinds a BALANCED policy is used. To change the default values use the `--location-p
olicy` flag.
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster lab-cluster in us-west4-c... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/qwiklabs-gcp-02-a0271246c06e/zones/us-west4-c/clusters/lab-cluster].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-west4-c/lab-cluster?project=qwiklabs-gcp-02
-a0271246c06e
kubeconfig entry generated for lab-cluster.
NAME: lab-cluster
LOCATION: us-west4-c
MASTER_VERSION: 1.22.12-gke.2300
MASTER_IP: 34.125.31.44
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.22.12-gke.2300
NUM_NODES: 3
STATUS: RUNNING
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$
```

Task 3: Get authentication credential for the cluster

After creating your cluster, you need authentication credentials to interact with it.

1. Authenticate with the cluster:
   a. gcloud container clusters get-credentials lab-cluster

```
STATUS: RUNNING
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ gcloud container clusters get-credentials lab-cluster
Fetching cluster endpoint and auth data.
kubeconfig entry generated for lab-cluster.
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$
```

Task 4: Deploy an application to the cluster

GKE uses Kubernetes objects to create and manage your cluster's resources. Kubernetes provides the Deployment object for deploying stateless applications like web servers. Service objects define rules and load balancing for accessing your application from the internet.

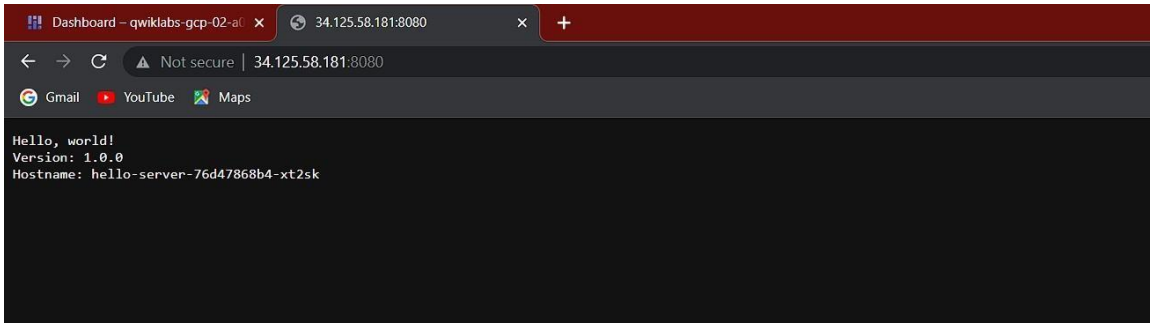
1. To **create a new Deployment** `hello-server` from the `hello-app` container image, run the following kubectl create command:
   a. kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:1.0

2. To **create a Kubernetes Service**, which is a Kubernetes resource that lets you expose your application to external traffic, run the following kubectl expose command:
   a. kubectl expose deployment hello-server --type=LoadBalancer --port 8080

3. To **inspect** the `hello-server` Service, run kubectl get:
   a. kubectl get service

4. To view the application from your web browser, open a new tab and enter the following address, replacing `[EXTERNAL IP]` with the `EXTERNAL-IP` for `hello-server`.
   a. http://[EXTERNAL-IP]:8080

```
kubeconfig entry generated for lab-cluster.
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:1.0
deployment.apps/hello-server created
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ kubectl expose deployment hello-server --type=LoadBalancer --port 8080
service/hello-server exposed
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ kubectl get service
NAME           TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
hello-server   LoadBalancer   10.0.8.2      <pending>     8080:30713/TCP   6s
kubernetes     ClusterIP      10.0.0.1      <none>        443/TCP          6m30s
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ kubectl get service
NAME           TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
hello-server   LoadBalancer   10.0.8.2      <pending>     8080:30713/TCP   24s
kubernetes     ClusterIP      10.0.0.1      <none>        443/TCP          6m48s
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ kubectl get service
NAME           TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)          AGE
hello-server   LoadBalancer   10.0.8.2      34.125.58.181   8080:30713/TCP   65s
kubernetes     ClusterIP      10.0.0.1      <none>          443/TCP          7m29s
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ http://34.125.58.181:8080
```



Task 5: Deleting the cluster

1. To delete the cluster run the following command:
   a. gcloud container clusters delete lab-cluster
2. When prompted type Y to conform:

```
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$ gcloud container clusters delete lab-cluster
The following clusters will be deleted.
 - [lab-cluster] in [us-west4-c]

Do you want to continue (Y/n)?  y

Deleting cluster lab-cluster...done.
Deleted [https://container.googleapis.com/v1/projects/qwiklabs-gcp-02-a0271246c06e/zones/us-west4-c/clusters/lab-cluster].
student_00_4186e6c66c91@cloudshell:~ (qwiklabs-gcp-02-a0271246c06e)$
```

Conclusion:

In this Practical, we have deployed and deleted containerized application to Kubernetes Engine.

PRACTICAL: 7

AIM:

Set Up Network and HTTP Load Balancers on Google Cloud Platform.

THEORY:

Active Cloud Shell:

1. Click **Activate Cloud Shell** at the top of the Google Cloud console.

2. Click **Continue**.

Task 1: Set the default region and zone for all resources

1. In cloud shell write following command to set default zone:
   a. **gcloud config set compute/zone**

2. After that set default region by following command:
   a. **gcloud config set compute/region**

```
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud auth list
Credentialed Accounts

ACTIVE: *
ACCOUNT: student-00-988ea34bc163@qwiklabs.net

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud config list project
[core]
project = qwiklabs-gcp-01-e763adff4182

Your active configuration is: [cloudshell-3634]
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud config set compute/zone us-west3-a
Updated property [compute/zone].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud config set compute/region us-west3
Updated property [compute/region].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$
```

Task 2: Create multiple web server instance

For this load balancing scenario, create three Compute Engine VM instances and install Apache on them, then add a firewall rule that allows HTTP traffic to reach the instances.

The code provided sets the zone to <filled in at lab start>. Setting the tags field lets you reference these instances all at once, such as with a firewall rule. These commands also install Apache on each instance and give each instance a unique home page.

1. Create a Virtual machine www1 in your default zone

```
gcloud compute instances create www1 \
  --zone= \
  --tags=network-lb-tag \
  --machine-type=e2-medium \
  --image-family=debian-11 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
    apt-get update
    apt-get install apache2 -y
    service apache2 restart
    echo "
<h3>Web Server: www1</h3>" | tee /var/www/html/index.html'
```

2. Create a virtual machine www2 in your default zone

```
gcloud compute instances create www2 \
  --zone= \
  --tags=network-lb-tag \
  --machine-type=e2-medium \
  --image-family=debian-11 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
    apt-get update
    apt-get install apache2 -y
    service apache2 restart
    echo "
<h3>Web Server: www2</h3>" | tee /var/www/html/index.html'
```

3. Create a virtual machine www3 in your default zone.

```
gcloud compute instances create www3 \
  --zone= \
  --tags=network-lb-tag \
  --machine-type=e2-medium \
  --image-family=debian-11 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
    apt-get update
    apt-get install apache2 -y
    service apache2 restart
    echo "
<h3>Web Server: www3</h3>" | tee /var/www/html/index.html'
```

4. Create a firewall rule to allow external traffic to the VM instances:

```
gcloud compute firewall-rules create www-firewall-network-lb \
  --target-tags network-lb-tag --allow tcp:80
```

5. Run the following to list your instances. You'll see their IP addresses in the `EXTERNAL_IP` column:

   a. gcloud compute instances list

6. Verify that each instance is running with `curl`, replacing **[IP_ADDRESS]** with the IP address for each of your VMs:

   a. curl http://[IP_ADDRESS]

```
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$   gcloud compute instances create www3 \
    --zone=us-west3-a \
    --tags=network-lb-tag \
    --machine-type=e2-medium \
    --image-family=debian-11 \
    --image-project=debian-cloud \
    --metadata=startup-script='#!/bin/bash
      apt-get update
      apt-get install apache2 -y
      service apache2 restart
      echo "
<h3>Web Server: www3</h3>" | tee /var/www/html/index.html'
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/zones/us-west3-a/instances/www3].
NAME: www3
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.4
EXTERNAL_IP: 34.106.85.38
STATUS: RUNNING
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute firewall-rules create www-firewall-network-lb \
    --target-tags network-lb-tag --allow tcp:80
Creating firewall...working..Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/firewalls/www-firewall-network-lb].
Creating firewall...done.
```

```
STATUS: RUNNING
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute firewall-rules create www-firewall-network-lb \
    --target-tags network-lb-tag --allow tcp:80
Creating firewall...working..Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/firewalls/www-firewall-network-lb].
Creating firewall...done.
NAME: www-firewall-network-lb
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 1000
ALLOW: tcp:80
DENY:
DISABLED: False
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute instances list
NAME: www1
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.2
EXTERNAL_IP: 34.106.208.119
STATUS: RUNNING

NAME: www2
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.3
EXTERNAL_IP: 34.106.134.212
STATUS: RUNNING

NAME: www3
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.4
EXTERNAL_IP: 34.106.85.38
STATUS: RUNNING
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$
```

```
DISABLED: False
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute instances list
NAME: www1
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.2
EXTERNAL_IP: 34.106.208.119
STATUS: RUNNING

NAME: www2
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.3
EXTERNAL_IP: 34.106.134.212
STATUS: RUNNING

NAME: www3
ZONE: us-west3-a
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.180.0.4
EXTERNAL_IP: 34.106.85.38
STATUS: RUNNING
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ curl http://10.180.0.4
^C
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ curl http://34.106.85.38

<h3>Web Server: www3</h3>
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$
```

Task 3: Configure the load balancing service

1. Create a static external IP address for your load balancer:
   gcloud compute addresses create network-lb-ip-1 \ --region

2. Add a legacy HTTP health check resource
   gcloud compute http-health-checks create basic-check

3. Add a target pool in the same region as your instances. Run the following to create the target pool and use the health check, which is required for the service to function:
   gcloud compute target-pools create www-pool \ --region --http-health-check basic-check

4. Add the instances to the pool:
   gcloud compute target-pools create www-pool \ --region --http-health-check basic-check

5. Add a forwarding rule:
   gcloud compute target-pools create www-pool \ --region --http-health-check basic-check

```
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ curl http://10.180.0.4
^C
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ curl http://34.106.85.38

<h3>Web Server: www3</h3>
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$    gcloud compute addresses create network-lb-ip-1 \
    --region us-west3
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/regions/us-west3/addresses/network-lb-ip-1].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute http-health-checks create basic-check
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/httpHealthChecks/basic-check].
NAME: basic-check
HOST:
PORT: 80
REQUEST_PATH: /
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$   gcloud compute target-pools create www-pool \
    --region us-west3 --http-health-check basic-check
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/regions/us-west3/targetPools/www-pool].
NAME: www-pool
REGION: us-west3
SESSION_AFFINITY: NONE
BACKUP:
HEALTH_CHECKS: basic-check
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute target-pools add-instances www-pool \
    --instances www1,www2,www3
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/regions/us-west3/targetPools/www-pool].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute forwarding-rules create www-rule \
    --region  us-west3 \
    --ports 80 \
    --address network-lb-ip-1 \
    --target-pool www-pool
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/regions/us-west3/forwardingRules/www-rule].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$
```

Task 4: Sending traffic to your instance

1. Enter the following command to view the external IP address of the www-rule forwarding rule used by the load balancer:
   gcloud compute forwarding-rules describe www-rule --region

2. Access the external IP address
   IPADDRESS=$(gcloud compute forwarding-rules describe www-rule --region --format="json" | jq -r .IPAddress)

3. Show the external IP address
   echo $IPADDRESS

4. Use `curl` command to access the external IP address, replacing `IP_ADDRESS` with an external IP address from the previous command:
   while true; do curl -m1 $IPADDRESS; done

5. Use **Ctrl** + **c** to stop running the command.

```
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ IPADDRESS=$(gcloud compute forwarding-rules describe www-rule --region us-west3 --format="json" | jq -r .IPAddress)
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ echo $IPADDRESS
34.106.62.128
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ while true; do curl -m1 $IPADDRESS; done

<h3>Web Server: www1</h3>

<h3>Web Server: www1</h3>

<h3>Web Server: www3</h3>

<h3>Web Server: www2</h3>

<h3>Web Server: www3</h3>

<h3>Web Server: www1</h3>

<h3>Web Server: www1</h3>

<h3>Web Server: www3</h3>
```

Task 5: create an HTTP load balancer

HTTP(S) Load Balancing is implemented on Google Front End (GFE). GFEs are distributed globally and operate together using Google's global network and control plane. You can configure URL rules to route some URLs to one set of instances and route other URLs to other instances.

Requests are always routed to the instance group that is closest to the user, if that group has enough capacity and is appropriate for the request. If the closest group does not have enough capacity, the request is sent to the closest group that **does** have capacity.

To set up a load balancer with a Compute Engine backend, your VMs need to be in an instance group. The managed instance group provides VMs running the backend servers of an external HTTP load balancer. For this lab, backends serve their own hostnames.

1. First, create the load balancer template:

```
gcloud compute instance-templates create lb-backend-template \
  --region= \
  --network=default \
  --subnet=default \
  --tags=allow-health-check \
  --machine-type=e2-medium \
  --image-family=debian-11 \
  --image-project=debian-cloud \
  --metadata=startup-script='#!/bin/bash
   apt-get update
   apt-get install apache2 -y
   a2ensite default-ssl
   a2enmod ssl
   vm_hostname="$(curl -H "Metadata-Flavor:Google" \
   http://169.254.169.254/computeMetadata/v1/instance/name)"
   echo "Page served from: $vm_hostname" | \
   tee /var/www/html/index.html
   systemctl restart apache2'
```

2. Create a managed instance group based on the template:

```
gcloud compute instance-groups managed create lb-backend-group \ --template=lb-backend-template --size=2 --zone=
```

3. Create the `fw-allow-health-check` firewall rule.

```
gcloud compute firewall-rules create fw-allow-health-check \
  --network=default \
  --action=allow \
  --direction=ingress \
  --source-ranges=130.211.0.0/22,35.191.0.0/16 \
```

```
  --target-tags=allow-health-check \
  --rules=tcp:80
```

4. Now that the instances are up and running, set up a global static external IP address that your customers use to reach your load balancer

```
gcloud compute addresses create lb-ipv4-1 \
  --ip-version=IPV4 \
  --global
gcloud compute addresses describe lb-ipv4-1 \
  --format="get(address)" \
  --global
```

5. Create a health check for the load balancer:

```
gcloud compute health-checks create http http-basic-check \ --port 80
```

6. Create a backend service:

```
gcloud compute backend-services create web-backend-service \
  --protocol=HTTP \
  --port-name=http \
  --health-checks=http-basic-check \
  --global
```

7. Add your instance group as the backend to the backend service:

```
gcloud compute backend-services add-backend web-backend-service \
  --instance-group=lb-backend-group \
  --instance-group-zone= \
  --global
```

8. Create a URL map to route the incoming requests to the default backend service:

```
gcloud compute backend-services add-backend web-backend-service \
  --instance-group=lb-backend-group \
  --instance-group-zone= \
  --global
```

9. Create a target HTTP proxy to route requests to your URL map:

```
gcloud compute target-http-proxies create http-lb-proxy \
    --url-map web-map-http
```

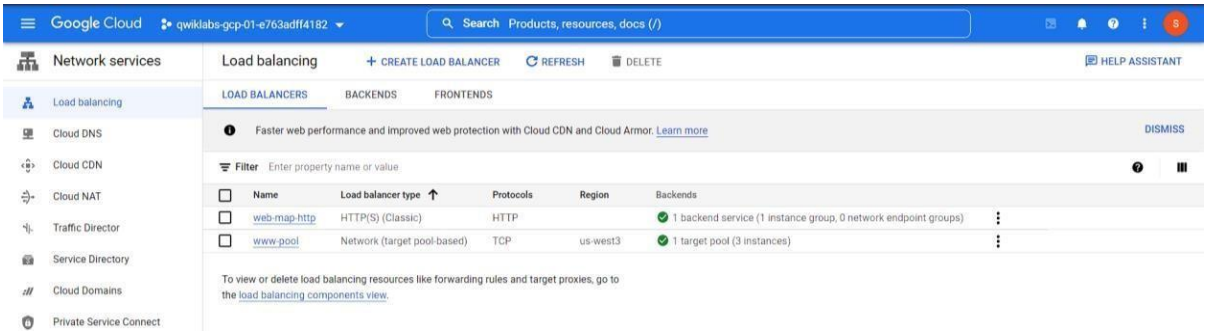
10. Create a global forwarding rule to route incoming requests to the proxy:

```
gcloud compute forwarding-rules create http-content-rule \
   --address=lb-ipv4-1\
   --global \
   --target-http-proxy=http-lb-proxy \
   --ports=80
```

```
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute instance-templates create lb-backend-template \
   --region=us-west3 \
   --network=default \
   --subnet=default \
   --tags=allow-health-check \
   --machine-type=e2-medium \
   --image-family=debian-11 \
   --image-project=debian-cloud \
   --metadata=startup-script='#!/bin/bash
     apt-get update
     apt-get install apache2 -y
     a2ensite default-ssl
     a2enmod ssl
     vm_hostname="$(curl -H "Metadata-Flavor:Google" \
     http://169.254.169.254/computeMetadata/v1/instance/name)"
     echo "Page served from: $vm_hostname" | \
     tee /var/www/html/index.html
     systemctl restart apache2'
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/instanceTemplates/lb-backend-template].
NAME: lb-backend-template
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
CREATION_TIMESTAMP: 2022-10-09T04:44:45.665-07:00
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute instance-groups managed create lb-backend-group \
   --template=lb-backend-template --size=2 --zone=us-west3-a
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/zones/us-west3-a/instanceGroupManagers/lb-backend-group].
NAME: lb-backend-group
LOCATION: us-west3-a
SCOPE: zone
BASE_INSTANCE_NAME: lb-backend-group
SIZE: 0
TARGET_SIZE: 2
INSTANCE_TEMPLATE: lb-backend-template
AUTOSCALED: no
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute firewall-rules create fw-allow-health-check \
```

```
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute firewall-rules create fw-allow-health-check \
  --network=default \
  --action=allow \
  --direction=ingress \
  --source-ranges=130.211.0.0/22,35.191.0.0/16 \
  --target-tags=allow-health-check \
  --rules=tcp:80
Creating firewall...working..Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/firewalls/fw-allow-health-check].
Creating firewall...done.
NAME: fw-allow-health-check
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 1000
ALLOW: tcp:80
DENY:
DISABLED: False
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute addresses create lb-ipv4-1 \
  --ip-version=IPV4 \
  --global
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/addresses/lb-ipv4-1].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute addresses describe lb-ipv4-1 \
  --format="get(address)" \
  --global
34.110.174.114
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute health-checks create http http-basic-check \
  --port 80
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/healthChecks/http-basic-check].
NAME: http-basic-check
PROTOCOL: HTTP
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute backend-services create web-backend-service \
  --protocol=HTTP \
  --port-name=http \
  --health-checks=http-basic-check \
  --global
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/backendServices/web-backend-service].
```

```
PROTOCOL: HTTP
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute backend-services create web-backend-service \
  --protocol=HTTP \
  --port-name=http \
  --health-checks=http-basic-check \
  --global
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/backendServices/web-backend-service].
NAME: web-backend-service
BACKENDS:
PROTOCOL: HTTP
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute backend-services add-backend web-backend-service \
  --instance-group=lb-backend-group \
  --instance-group-zone=us-west3-a \
  --global
Updated [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/backendServices/web-backend-service].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute url-maps create web-map-http \
    --default-service web-backend-service
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/urlMaps/web-map-http].
NAME: web-map-http
DEFAULT_SERVICE: backendServices/web-backend-service
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute target-http-proxies create http-lb-proxy \
    --url-map web-map-http
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/targetHttpProxies/http-lb-proxy].
NAME: http-lb-proxy
URL_MAP: web-map-http
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$ gcloud compute forwarding-rules create http-content-rule \
    --address=lb-ipv4-1\
    --global \
    --target-http-proxy=http-lb-proxy \
    --ports=80
Created [https://www.googleapis.com/compute/v1/projects/qwiklabs-gcp-01-e763adff4182/global/forwardingRules/http-content-rule].
student_00_988ea34bc163@cloudshell:~ (qwiklabs-gcp-01-e763adff4182)$
```



Task 6: Testing traffic sent to your instance

1. In the Cloud Console, from the **Navigation menu**, go to **Network services** > **Load balancing**.

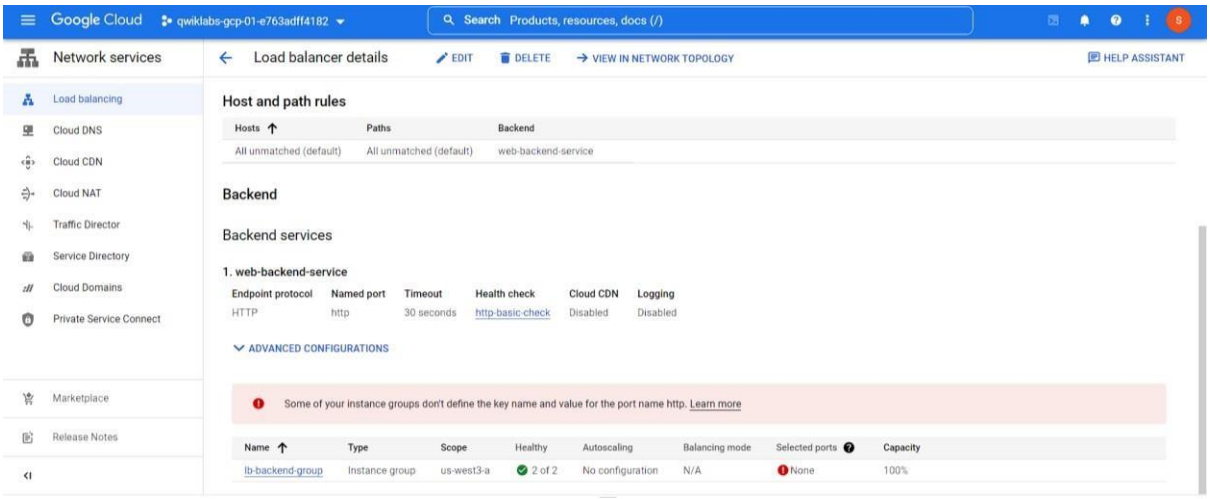
2. Click on the load balancer that you just created (`web-map-http`).

3. In the **Backend** section, click on the name of the backend and confirm that the VMs are **Healthy**. If they are not healthy, wait a few moments and try reloading the page.

4. When the VMs are healthy, test the load balancer using a web browser, going to `http://IP_ADDRESS/`, replacing `IP_ADDRESS` with the load balancer's IP address.

This may take three to five minutes. If you do not connect, wait a minute, and then reload the browser.

Your browser should render a page with content showing the name of the instance that served the page, along with its zone (for example, `Page served from: lb-backend-group-xxxx`).



Conclusion:

In this Practical, we have built a network load balancer and an HTTP(S) load balancer and practiced using instance template and managed instance group.

PRACTICAL: 8

AIM:

Create and Manage Cloud Resources: Challenge Lab on Google cloud Platform.

THEORY:

Active Cloud Shell:

1. Click **Activate Cloud Shell** at the top of the Google Cloud console.

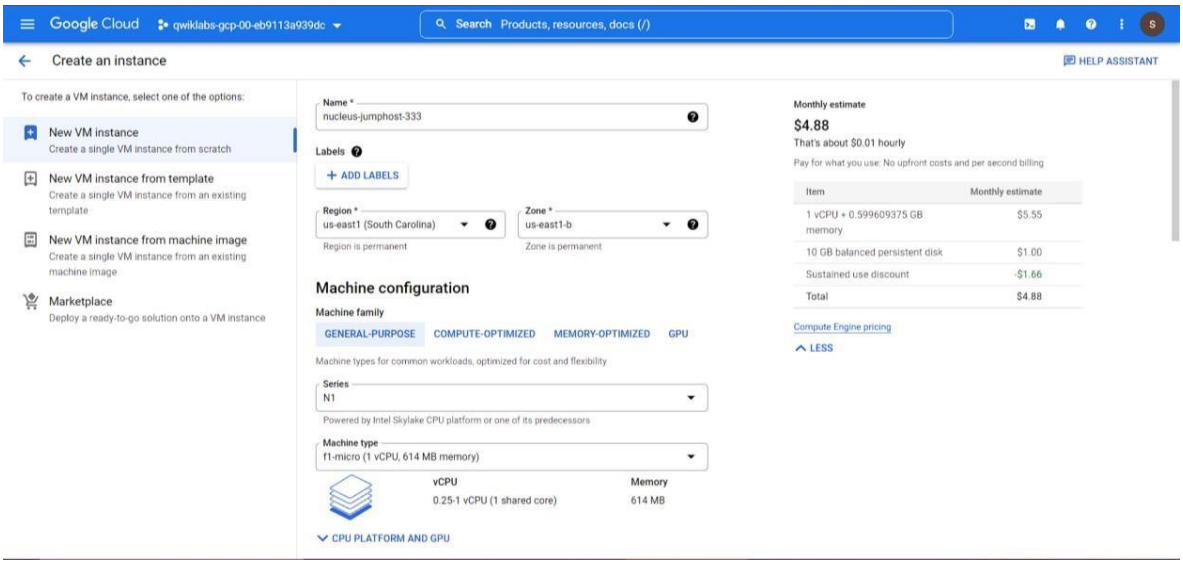
2. Click **Continue**.

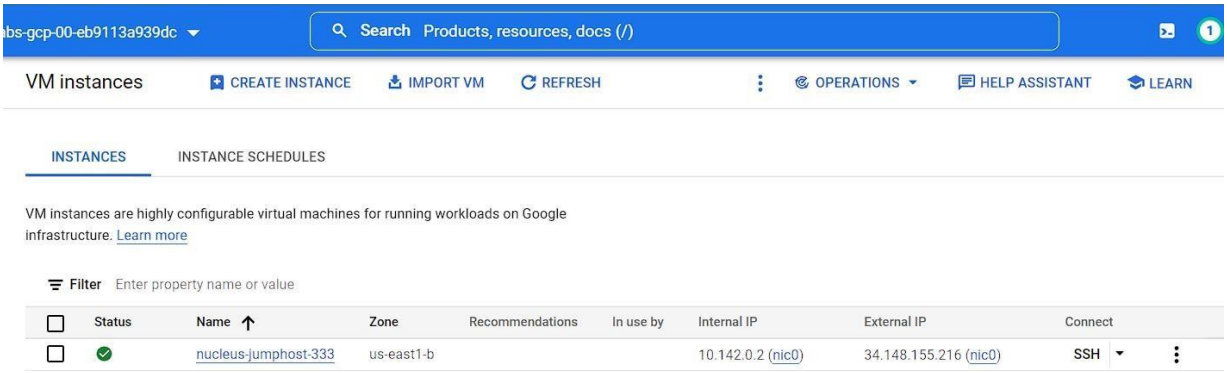
Task 1: Create a project jumphost instance

In the Cloud Console, on the top left of the screen, select Navigation menu > Compute Engine > VM Instances:

Enter details as following to create a VM Instance:

- Name for the VM instance : `nucleus-jumphost`
- Region : leave Default Region
- Zone : leave Default Zone
- Machine Type : `f1-micro` (Series - N1)
- Boot Disk : use the default image type (Debian Linux)
- Create

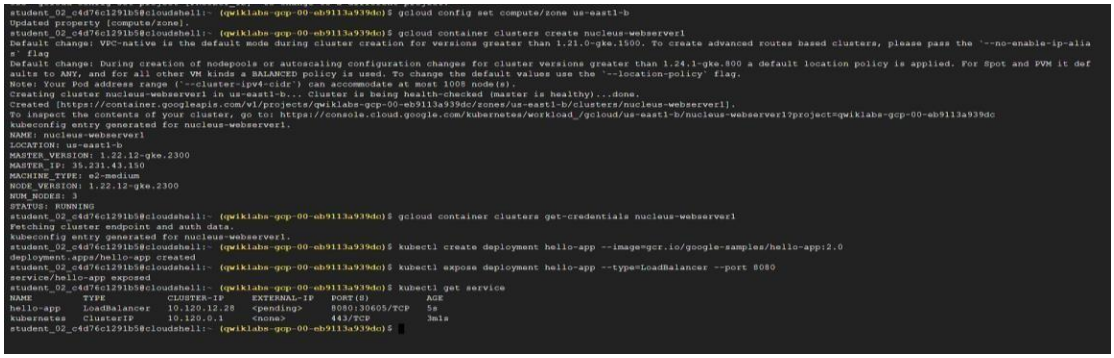




Task 2: Create a Kubernetes service cluster

Run Following command to create a cluster to host service

```
gcloud config set compute/zone us-east1-b
gcloud container clusters create nucleus-webserver1
gcloud container clusters get-credentials nucleus-webserver1
kubectl create deployment hello-app --image=gcr.io/google-samples/hello-app:2.0
kubectl expose deployment hello-app --type=LoadBalancer --port 8080
kubectl get service
```



Task 3: Set up an HTTP load balancer

You will serve the site via nginx web servers, but you want to ensure that the environment is fault-tolerant. Create an HTTP load balancer with a managed instance group of **2 nginx web servers**. Use the following code to configure the web servers; the team will replace this with their own configuration later.

After that we need to perform following task:

- Create an instance template.
- Create a target pool.
- Create a managed instance group.

- Create a firewall rule named as Firewall rule to allow traffic (80/tcp).
- Create a health check.
- Create a backend service, and attach the managed instance group with named port (http:80).
- Create a URL map, and target the HTTP proxy to route requests to your URL map.
- Create a forwarding rule.

```
cat << EOF > startup.sh
#! /bin/bash
apt-get update
apt-get install -y nginx
service nginx start
sed -i -- 's/nginx/Google Cloud Platform - "\$HOSTNAME"/' /var/www/html/index.nginx-
debian.html
EOF

gcloud compute instance-templates create web-server-template \
        --metadata-from-file startup-script=startup.sh \
        --network nucleus-vpc \
        --machine-type g1-small \
        --region us-east1

gcloud compute instance-groups managed create web-server-group \
        --base-instance-name web-server \
        --size 2 \
        --template web-server-template \
        --region us-east1

gcloud compute firewall-rules create web-server-firewall \
        --allow tcp:80 \
        --network nucleus-vpc

gcloud compute http-health-checks create http-basic-check

gcloud compute instance-groups managed \
        set-named-ports web-server-group \
        --named-ports http:80 \
        --region us-east1

gcloud compute backend-services create web-server-backend \
        --protocol HTTP \
        --http-health-checks http-basic-check \
        --global

gcloud compute backend-services add-backend web-server-backend \
        --instance-group web-server-group \
        --instance-group-region us-east1 \
        --global

gcloud compute url-maps create web-server-map \
```

```
    --default-service web-server-backend

gcloud compute target-http-proxies create http-lb-proxy \
    --url-map web-server-map

gcloud compute forwarding-rules create http-content-rule \
    --global \
    --target-http-proxy http-lb-proxy \
    --ports 80

gcloud compute forwarding-rules list
```

```
-bash: set-named-ports: command not found
student_02_c4d76c1291b5@cloudshell:~ (qwiklabs-gcp-00-eb9113a939dc)$ cat << EOF > startup.sh
#! /bin/bash
apt-get update
apt-get install -y nginx
service nginx start
sed -i -- 's/nginx/Google Cloud Platform - '"\$HOSTNAME"'/' /var/www/html/index.nginx-debian.html
EOF

gcloud compute instance-templates create web-server-template \
          --metadata-from-file startup-script=startup.sh \
          --network nucleus-vpc \
          --machine-type g1-small \
          --region us-east1

gcloud compute instance-groups managed create web-server-group \
          --base-instance-name web-server \
          --size 2 \
          --template web-server-template \
          --region us-east1

gcloud compute firewall-rules create web-server-firewall \
          --allow tcp:80 \
          --network nucleus-vpc

gcloud compute http-health-checks create http-basic-check

gcloud compute instance-groups managed \
          set-named-ports web-server-group \
          --named-ports http:80 \
          --region us-east1

gcloud compute backend-services create web-server-backend \
          --protocol HTTP \
          --http-health-checks http-basic-check \
          --global
```

```
NAME: a7a800636d2604805ba6d9baa7689fd1
REGION: us-east1
IP_ADDRESS: 34.74.119.68
IP_PROTOCOL: TCP
TARGET: us-east1/targetPools/a7a800636d2604805ba6d9baa7689fd1
student_02_c4d76c1291b5@cloudshell:~ (qwiklabs-gcp-00-eb9113a939dc)$
```

Conclusion:

In this Practical, we have learnt how to create an Instance, Kubernetes cluster and setup an HTTP load balancer.

PRACTICAL: 9

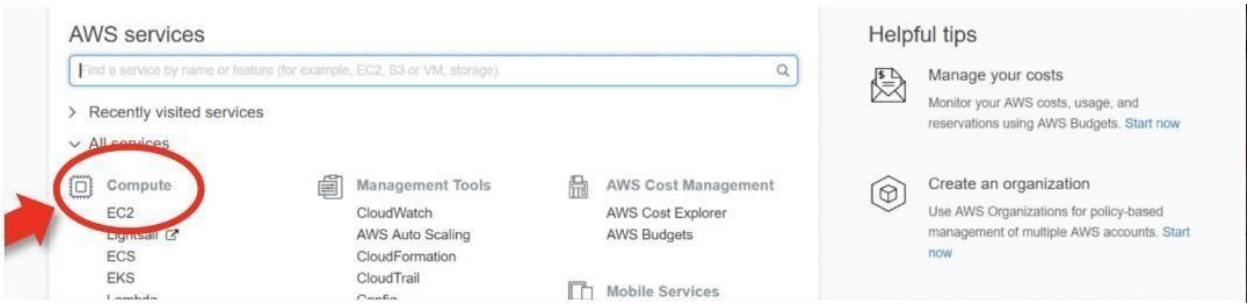
AIM:

1. **Create and Setup Amazon Elastic Compute Cloud (EC2) on Amazon cloud Platform.**
2. **Create and setup monitoring service for AWS cloud resources and the applications you run on AWS (Amazon CloudWatch).**
3. **Create an AWS Identity and Access Management (IAM) group and user, attach a policy and add a user to a group.**

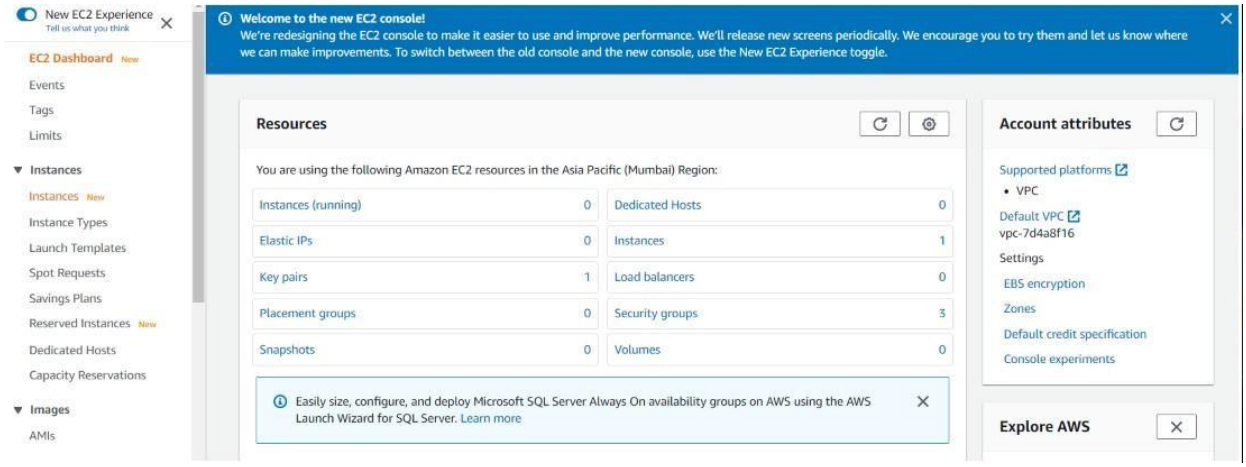
THEORY:

Task 1: Create and Setup Amazon Elastic Compute Cloud (EC2) on Amazon cloud Platform.

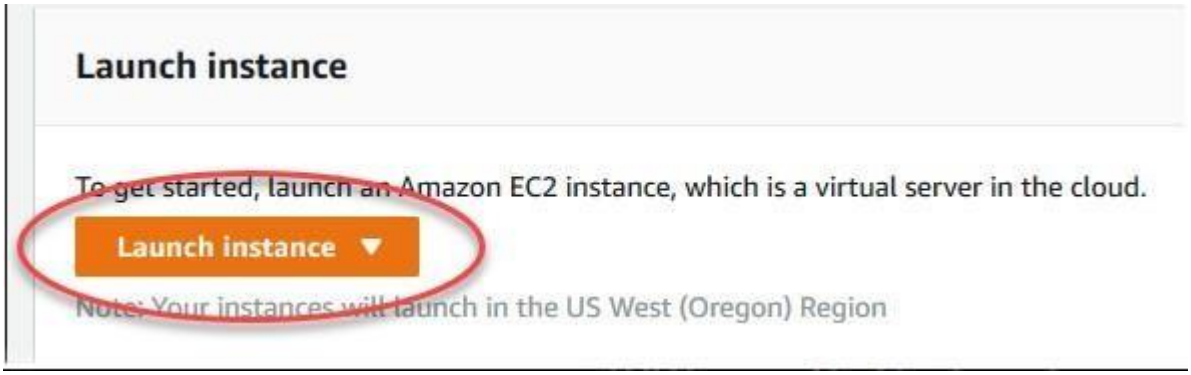
1. First, login into your AWS account and click on “services” present on the left of the AWS management console, i.e. the primary screen. And from the drop-down menu of options, tap on “EC2”. Here is the image attached to refer to.



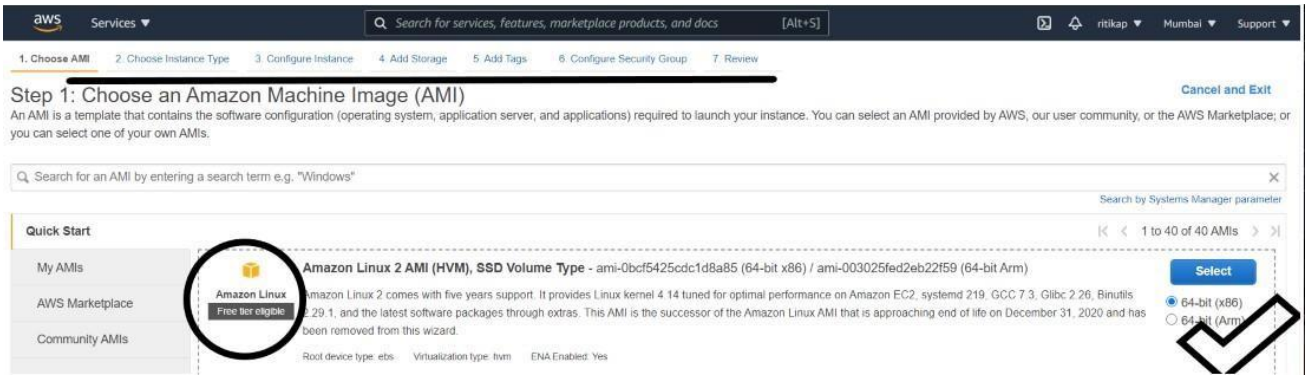
2. In a while, the EC2 console will be loaded onto your screen. Once it is done, from the list of options on the left in the navigation pane, click on “Instances”. Please refer to the image attached ahead for a better understanding.



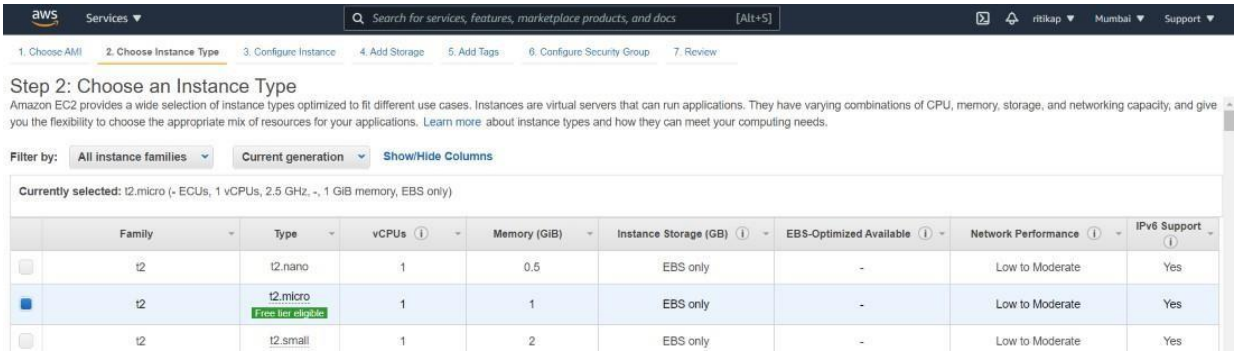
3. A new fresh screen will be loaded in a while. In the right corner, there will be an orange box named “Launch Instance”. Click on that box and wait. Here is the image to refer to.



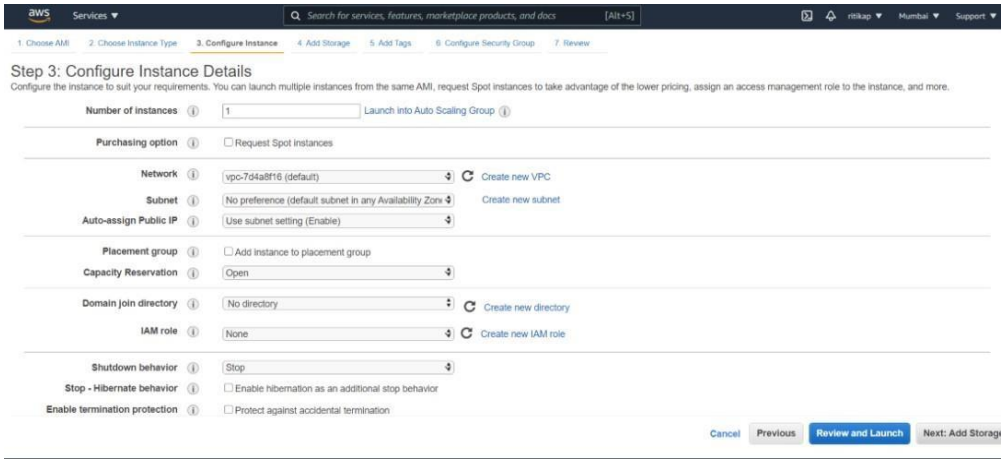
4. Now, the process of launching an EC2 instance will start. The next screen will display you a number of options to choose your AMI(Amazon Machine Image) from it. And horizontally, on the menu bar you will see, there is a 7-step procedure written to be followed for successfully launching an instance. I have chosen “Amazon Linux 2 AMI” as my AMI. And then go ahead, click “Next”. Refer to the image for any confusion.



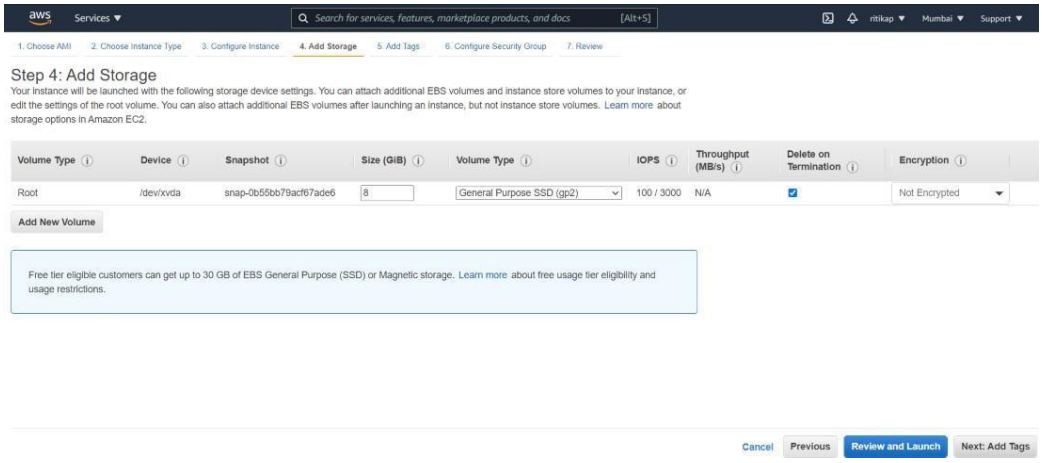
5. Now, comes sub-step-2 out of 7-steps of creating the instance, i.e. “Choose Instance Type”. I have chosen “t2 micro” as my instance type because I am a free tier user and this instance type is eligible to use for us. Then click “Next”. Refer to the image attached ahead for better understanding.



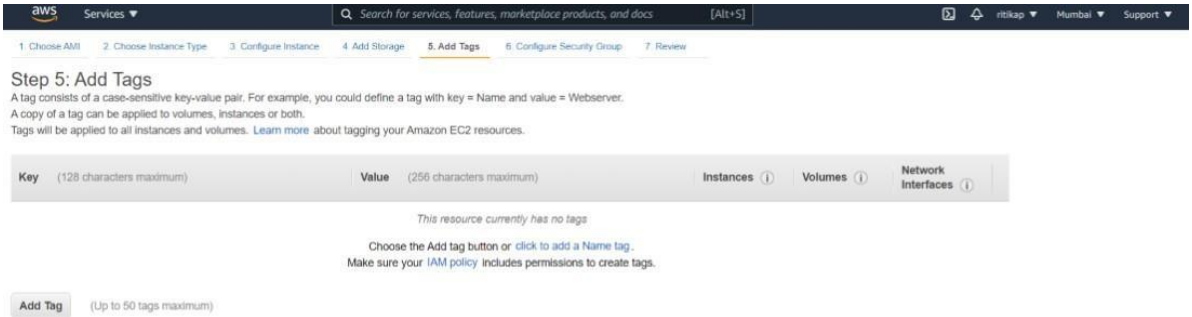
6. Further comes sub-step 3 out of the 7-step process of creating the instance, i.e. “Configure Instance”. Here we will confirm all the configurations we need for our EC2. By default, the configurations are filled, we just confirm them or alter them as per our needs and click “Next” to proceed. Here’s the image for better understanding and resolving confusion.



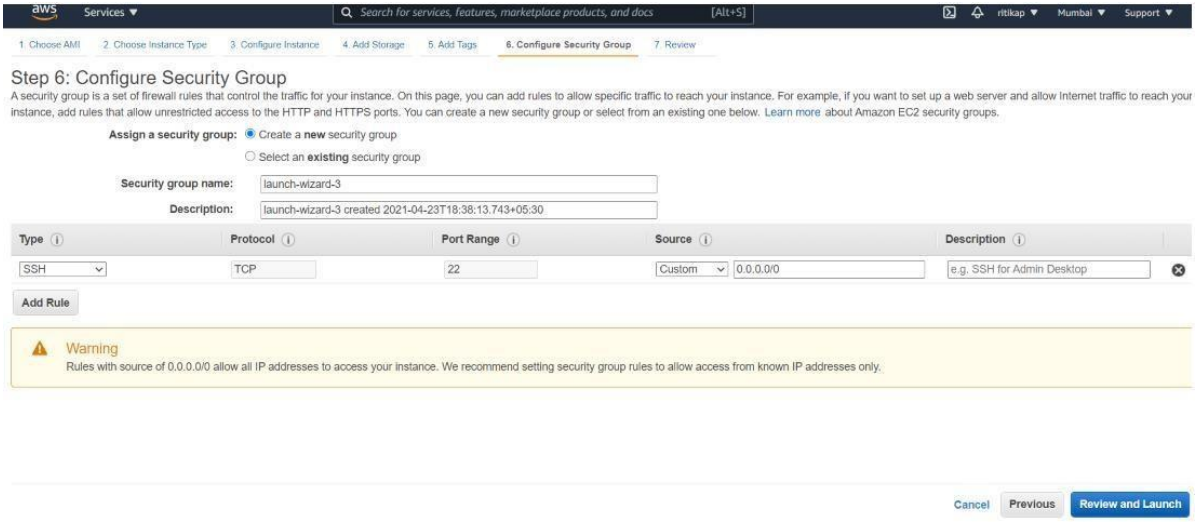
7. Next comes sub step 4 out of the 7-step process of creating the instance, i.e. “Add Storage”. Here we will look at the pre-defined storage configurations and modify them if they are not aligned as per our requirements. Then click “Next”. Here’s the image of the storage window attached ahead to understand better.



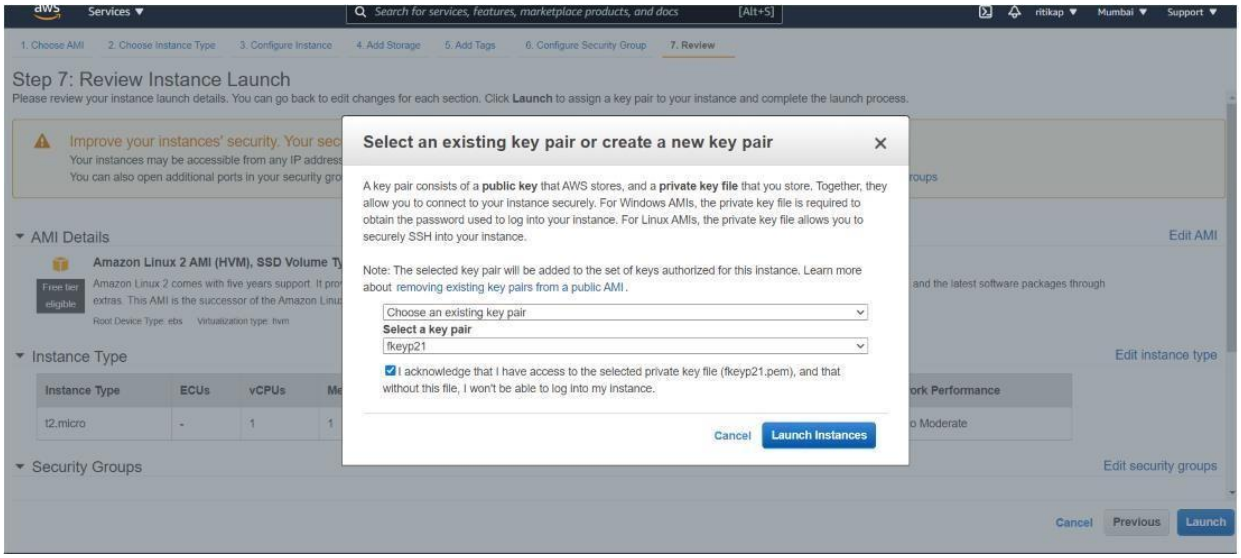
8. Next comes sub-step 5 out of the 7-step process of creating the instance, i.e. “Add Tags”. Here we will just click “Next” and proceed ahead. Here’s the image to refer to.



9. Now we will complete the 6th sub step out of the 7-step process of creating the instance, which is “Configure Security Group”. In Security Group, we have to give a group name and a group description, followed by the number and type of ports to open and the source type. In order to resolve confusion please refer to the image attached ahead.



10. Now we will complete the last step of the process of creating the instance, which is “Review”. In review, we will finally launch the instance and then a new dialog box will appear to ask for the “Key Pair”. Key pairs are used for authentication of the user when connecting to your EC2. We will be given two options to choose from, whether to choose an existing key pair or creating a new one and downloading it to launch. It is not necessary to create a new key pair every time you can use the previous one as well. Here is the image of the window attached.



Task 2: Create and setup monitoring service for AWS cloud resources and the applications you run on AWS (Amazon CloudWatch).

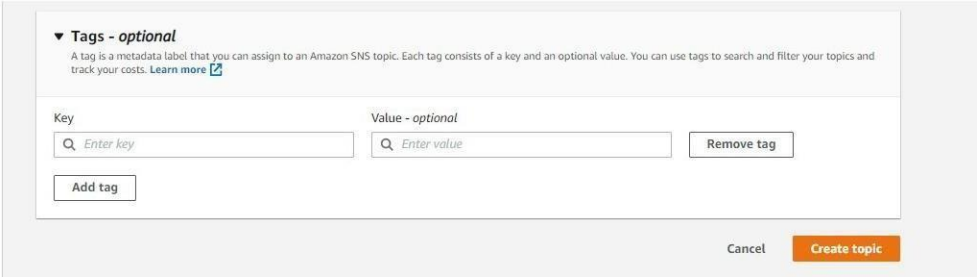
Notifying website management team when the instance on which website is hosted stops Whenever the CPU utilization of instance (on which website is hosted) goes above 80%, cloudwatch event is triggered. This cloudwatch event then activates the SNS topic which sends the alert email to the attached subscribers.

1. Let us assume that you have already launched an instance with the name tag ‘instance’.

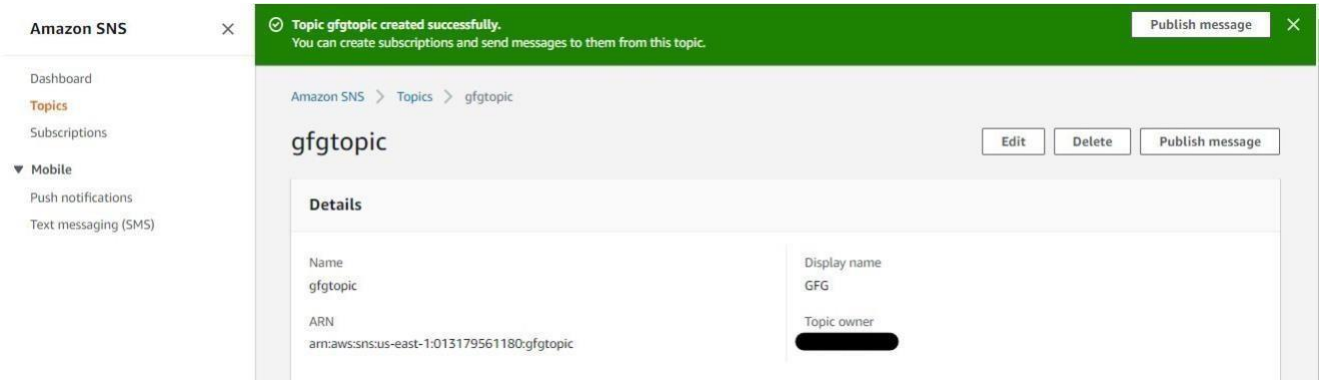
2. Go to SNS topic dashboard and click on create a topic

3. You will be directed to this dashboard. Now specify the name and display name.

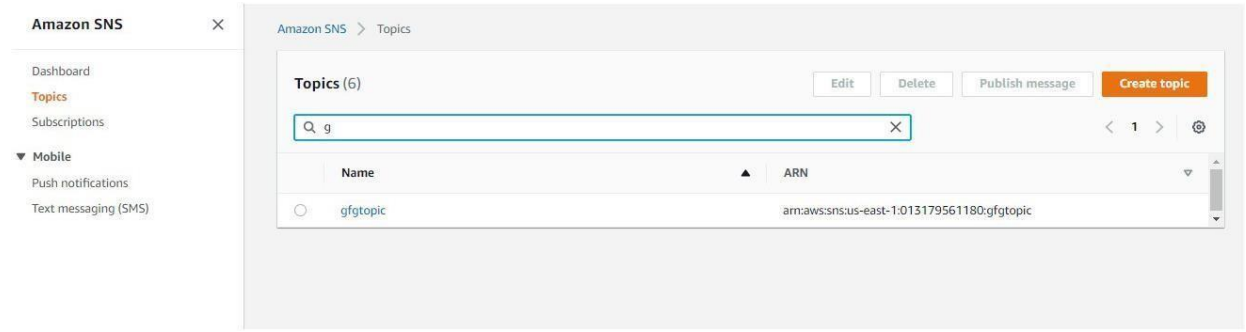
4. Scroll down and click on create the topic.



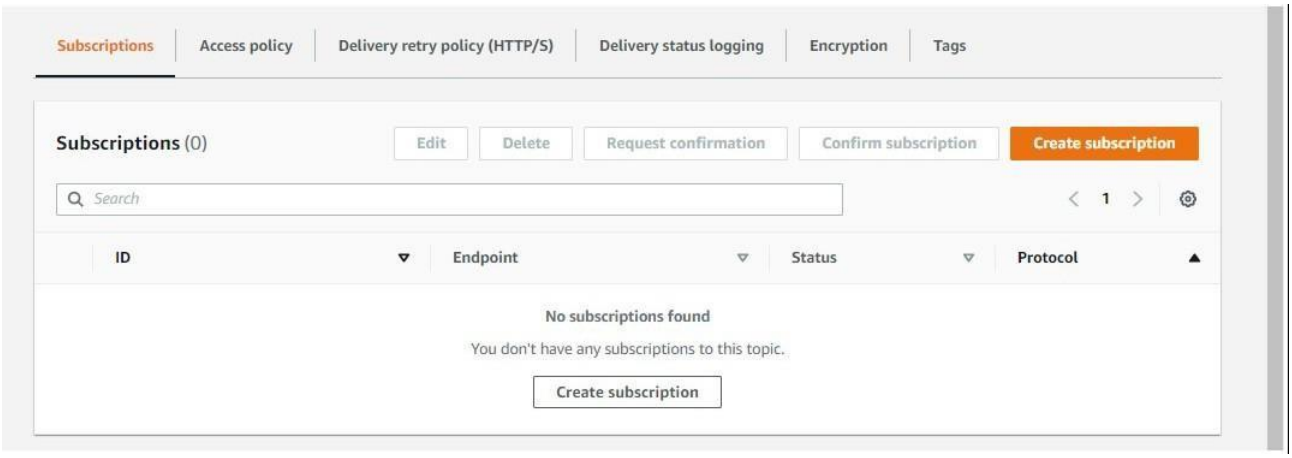
5. The SNS topic is created successfully



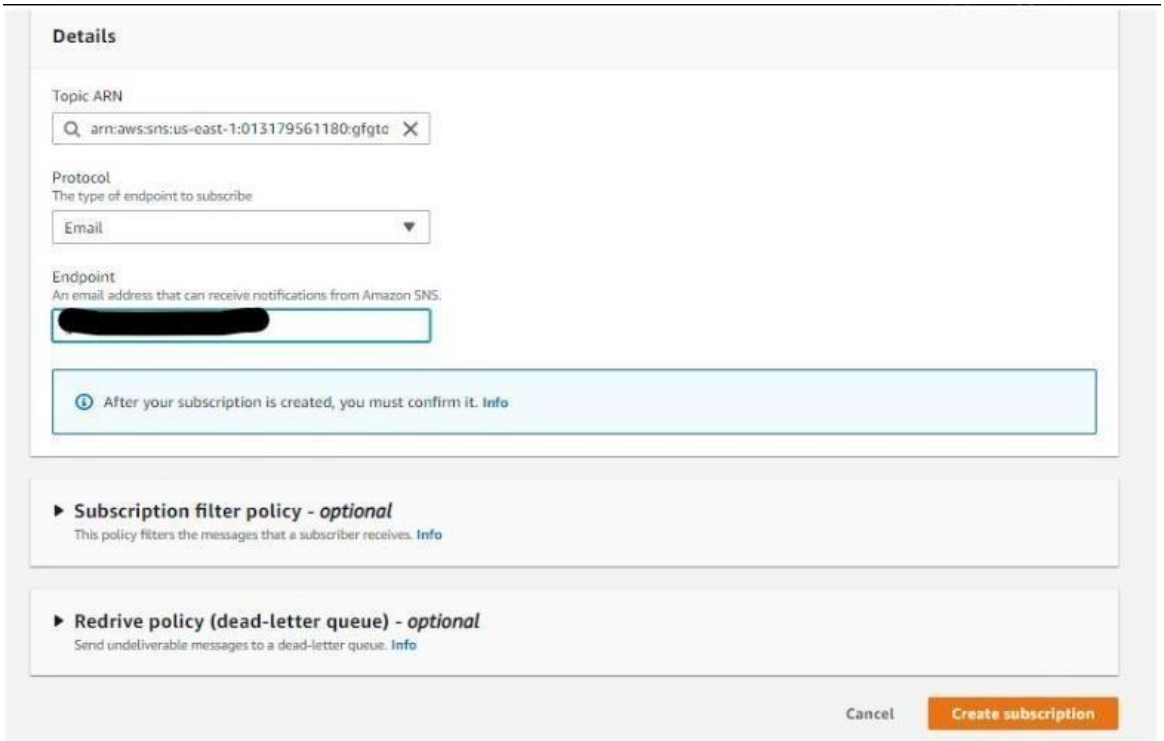
6. Go to the SNS topic dashboard and click on gfgtopic link.



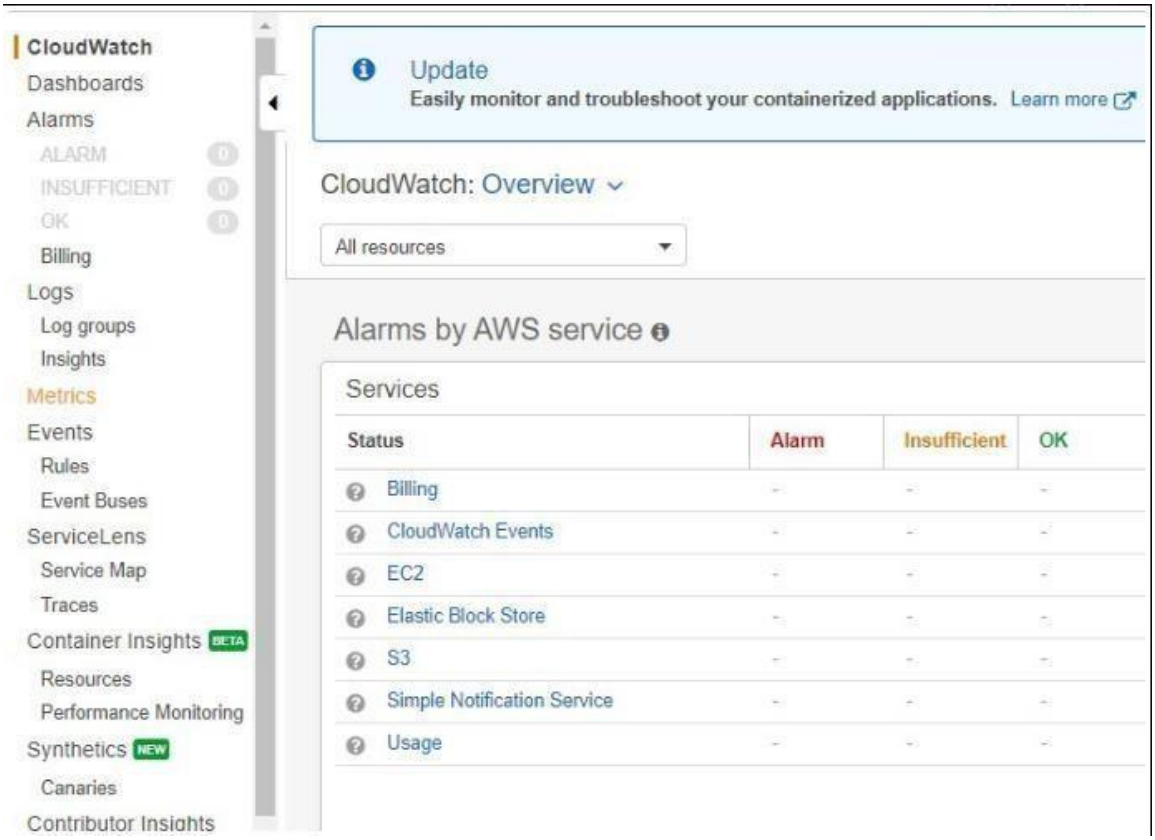
7. Under the subscriptions section, Click on Create subscription.



8. Select Email as protocol and specify the email address of subscribers in Endpoint. Click on create the subscription. Now Go to the mailbox of the specified email id and click on Subscription confirmed.



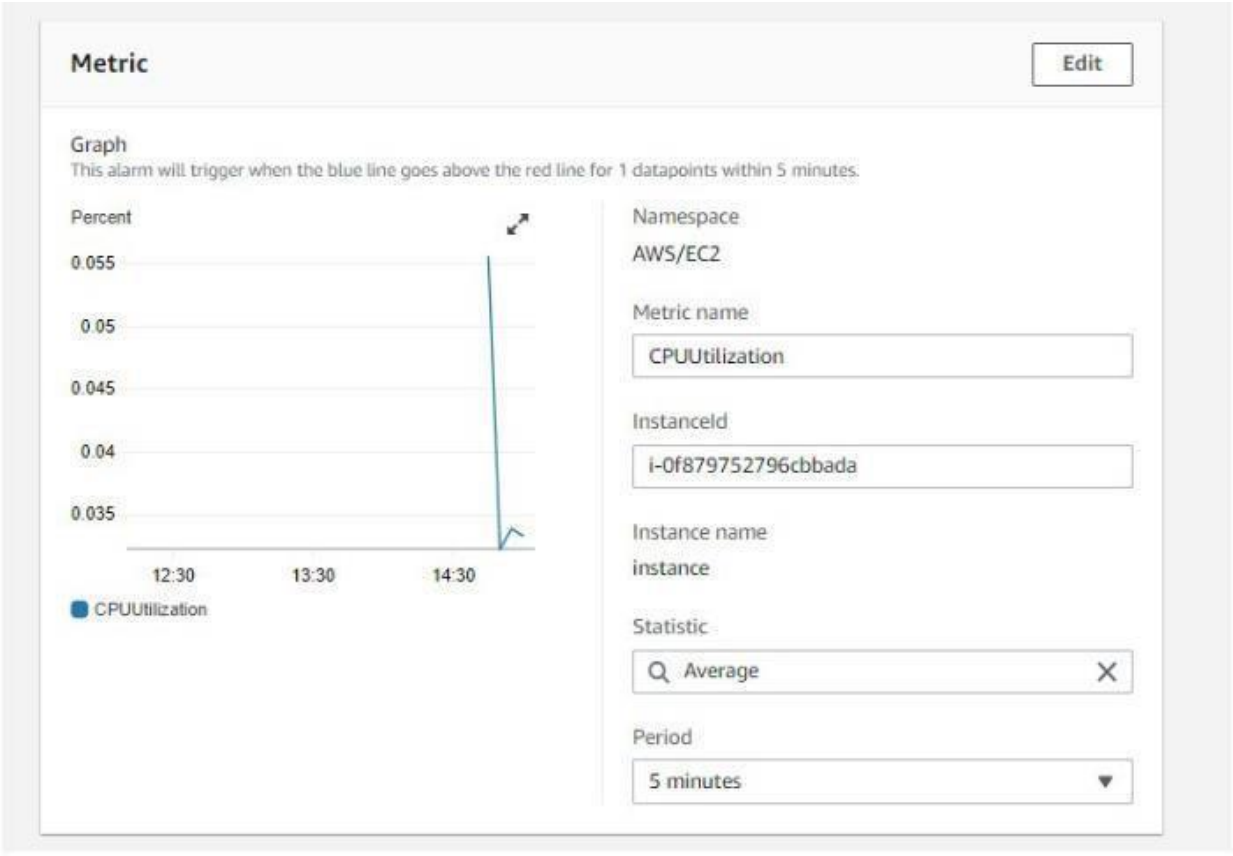
9. Go to the cloudwatch dashboard on the AWS management console. Click on Metrics in the left pane.



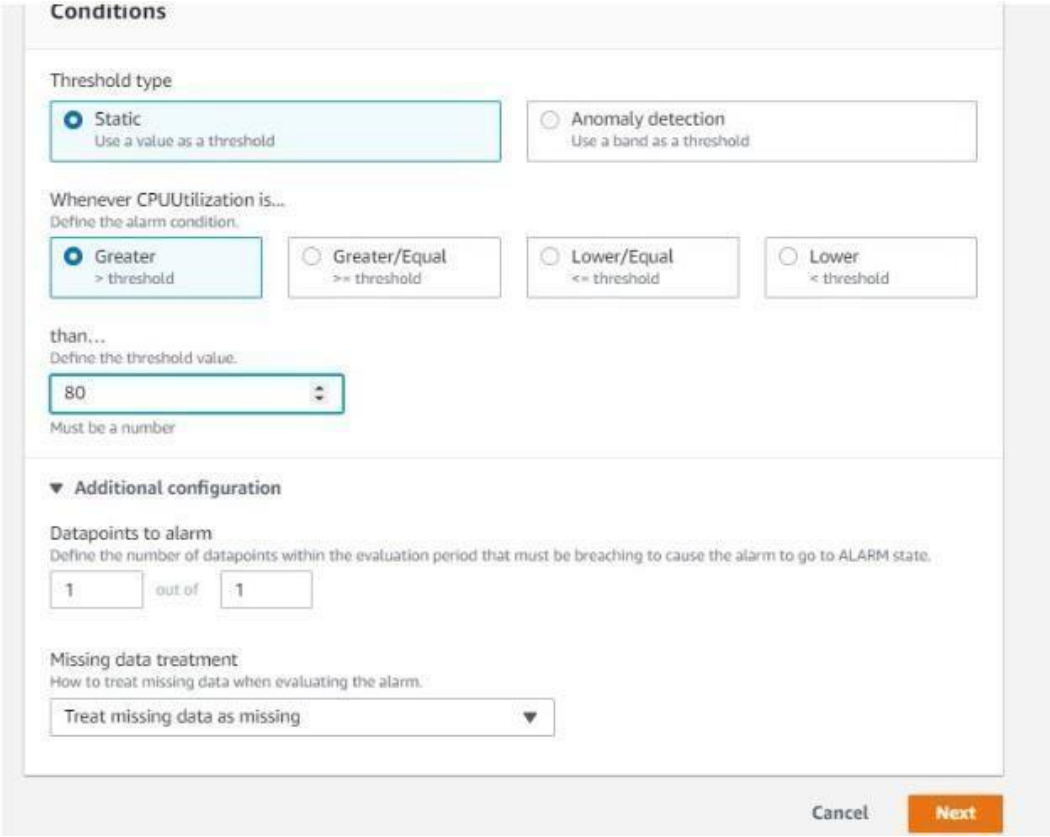
10. Select the instance you launched

11. Go to Graphed metrics, click on the bell icon

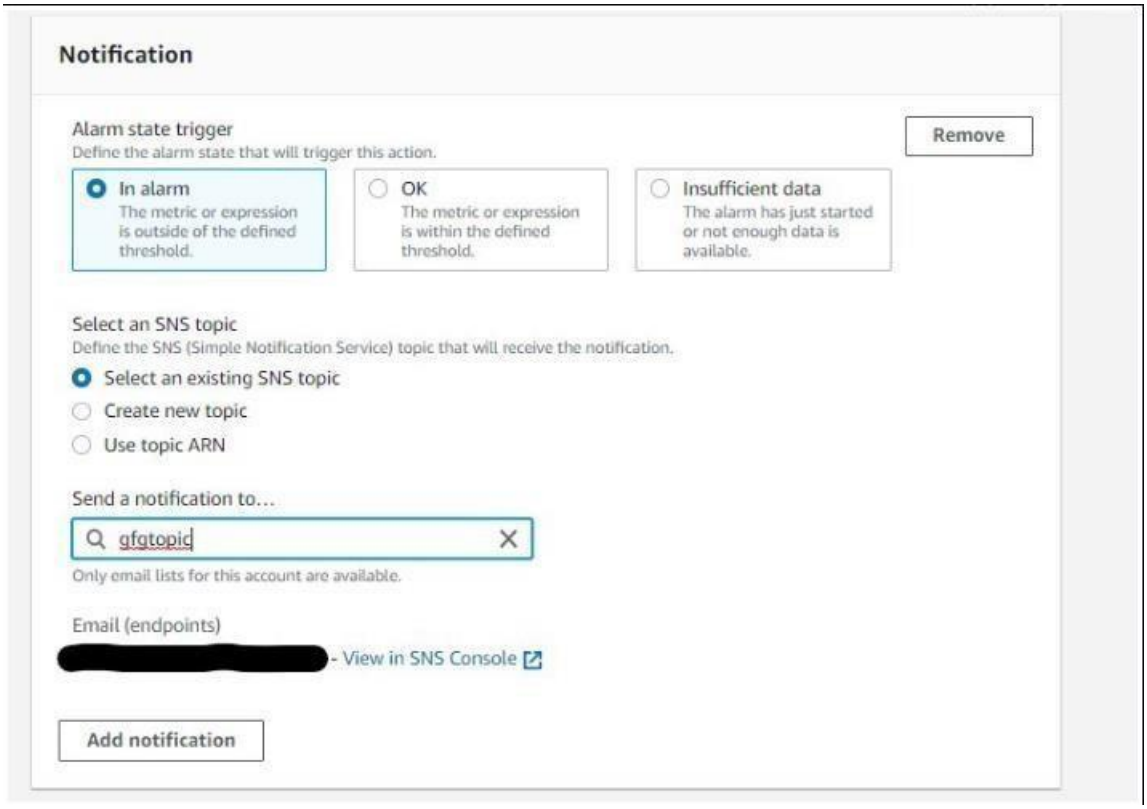
12. This dashboard shows the components of Amazon Cloudwatch such as Namespace, Metric Name, Statistics, etc



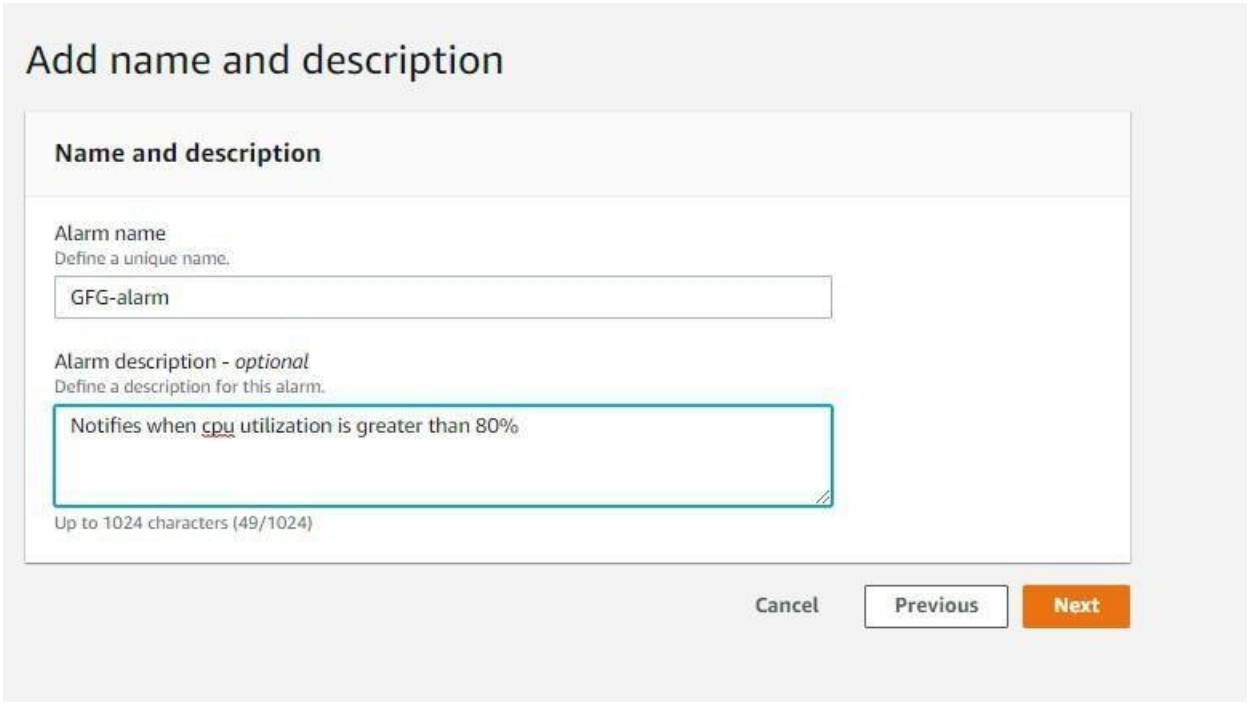
13. Select the greater threshold. Also, specify the amount(i.e 80) of the threshold value. Click on Next.



14. Click on Select an existing SNS topic, also mention the name of the SNS topic you created now.



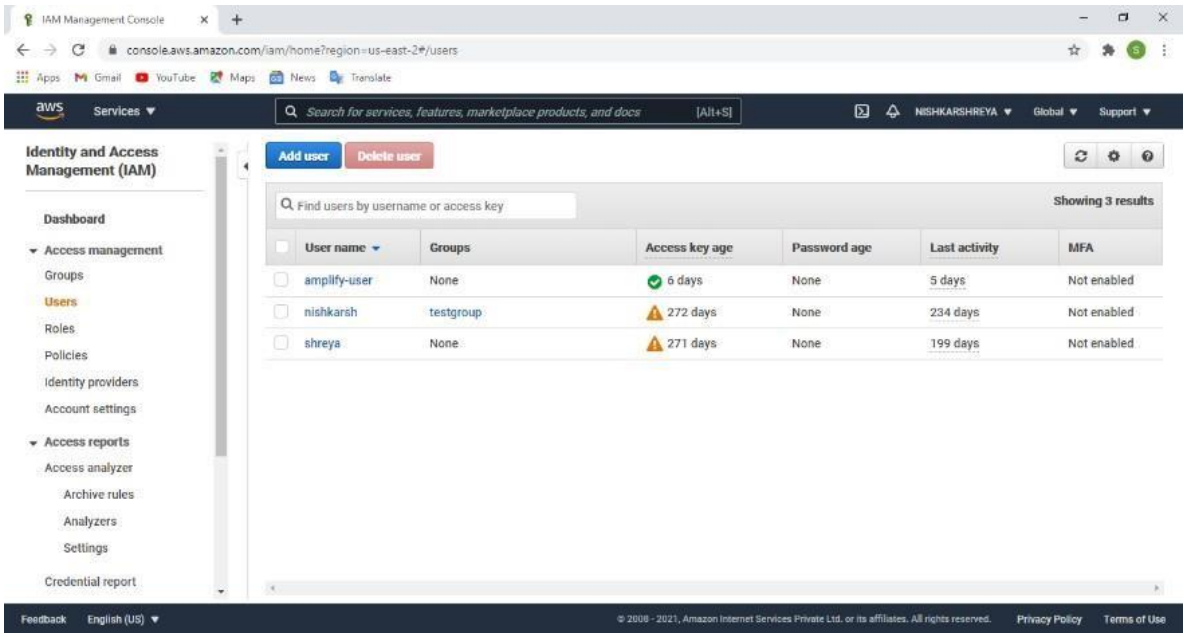
15. Specify the name of alarm and description which is completely optional. Click on Next and then click on Create alarm.



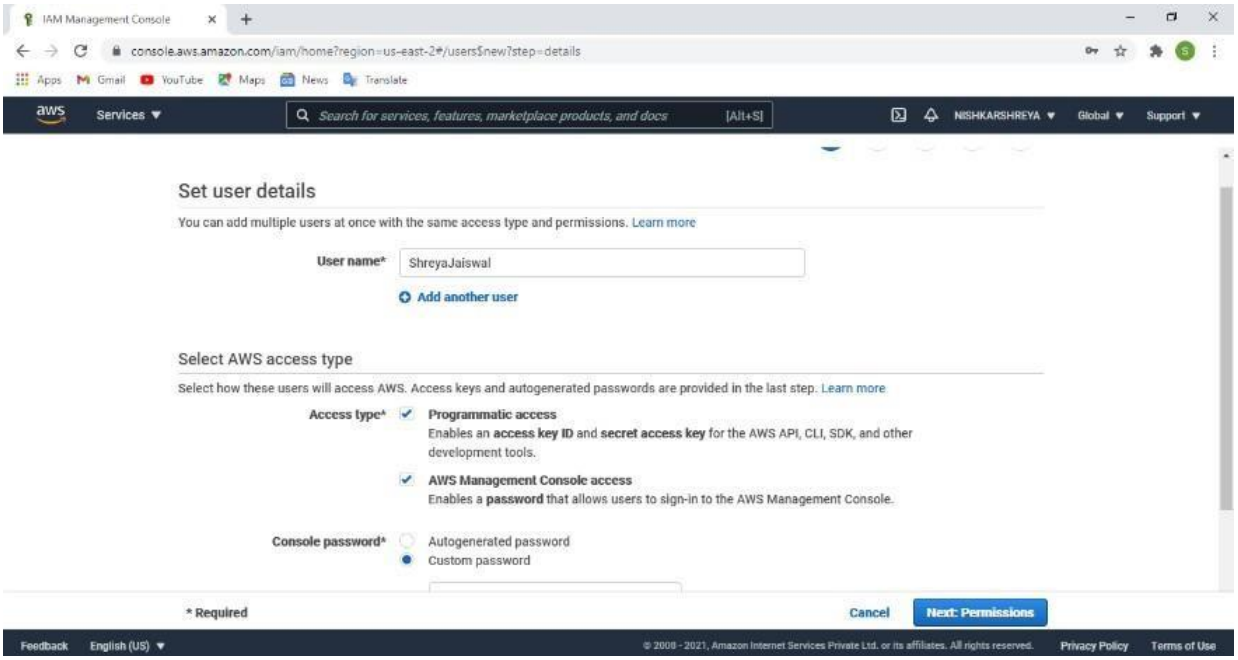
16. You can see the graph which notifies whenever CPU utilization goes above 80%.

Task 3: Create an AWS Identity and Access Management (IAM) group and user, attach a policy and add a user to a group.

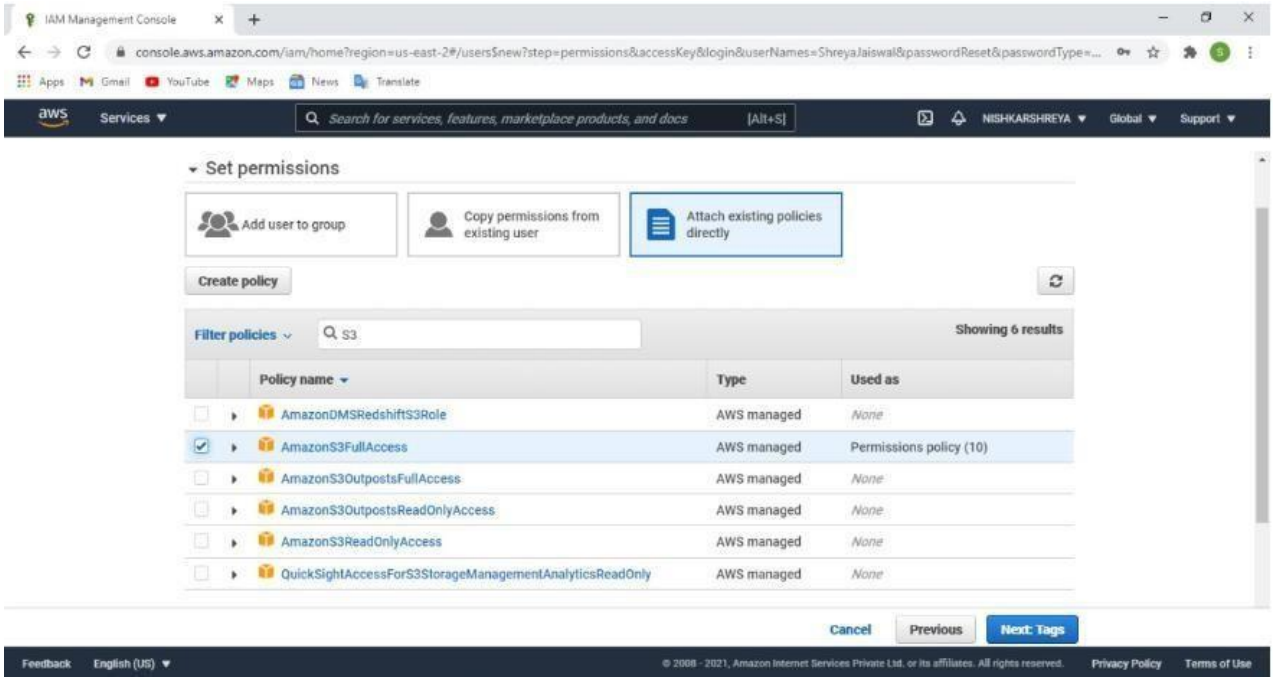
1. Steps to create an IAM user
   - You must have an AWS account, sign in as a root user to the AWS Management Console dashboard.
   - Search IAM in services goes to the IAM dashboard.



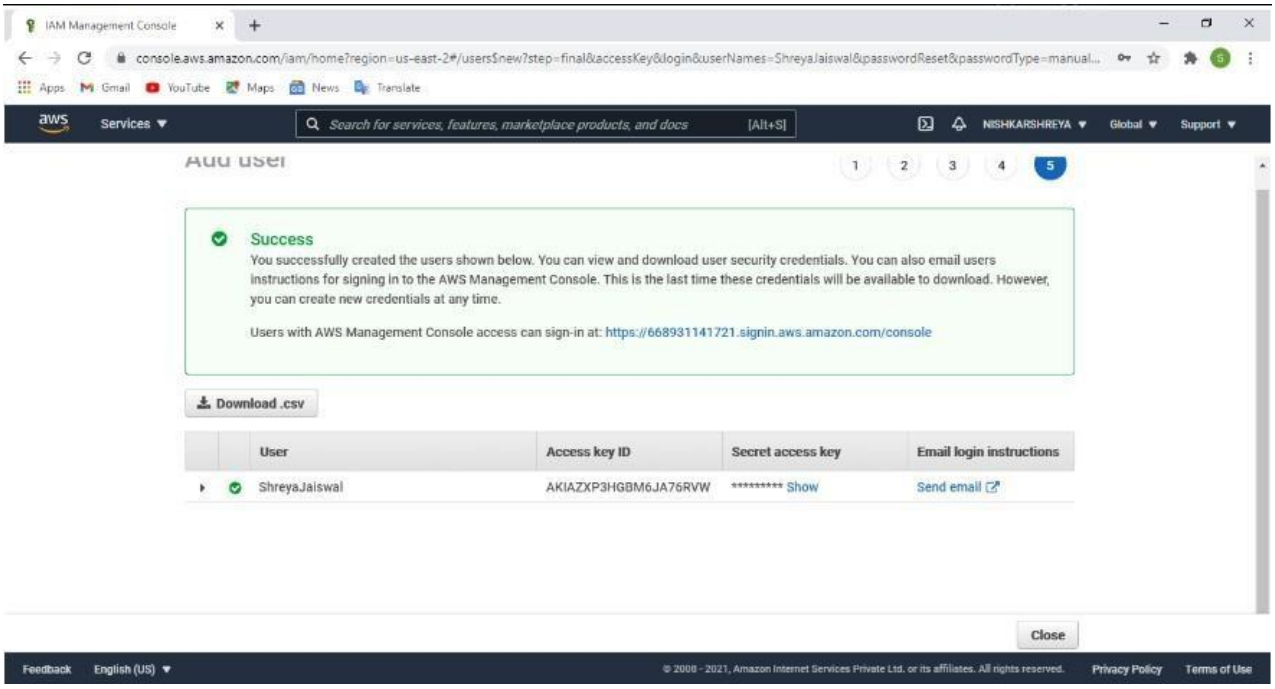
   - Select user →Click to Add user, provide a username, and select any one or both access type(programmatic access and AWS Management Console access), select auto-generated password or custom password(give your own password).



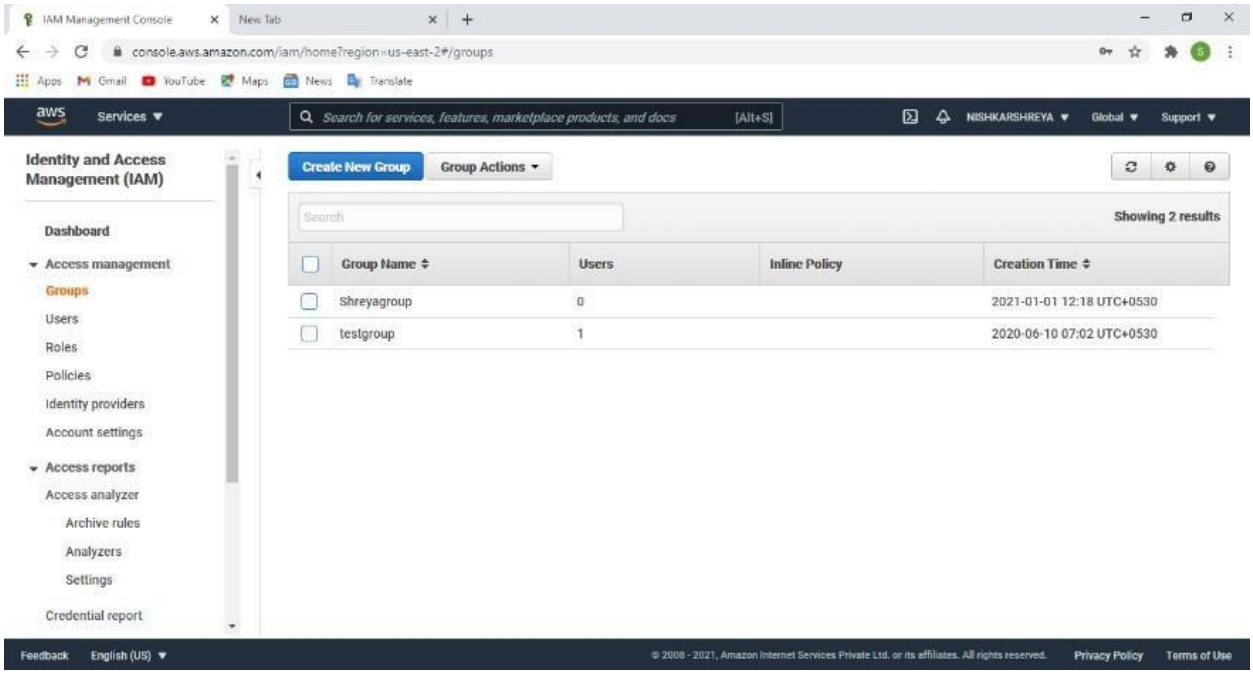
- Click on the attach polices → Next: permissions.



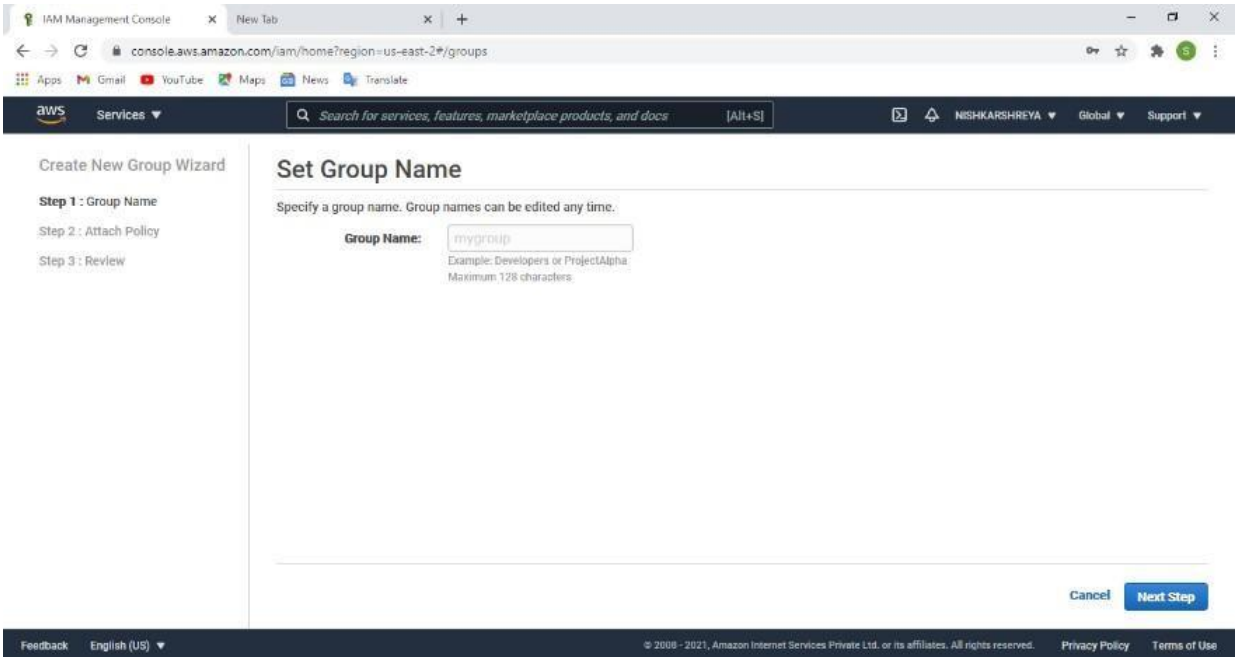
- Click on Next: Tags provide key and value to your user which will be helpful in searching when you have so many IAM users.
- Click on Reviews check all the configurations and make changes if needed.
- Click on create user and your IAM user is successfully created and as you have chosen programmatic access an Access key ID and a secret access key.



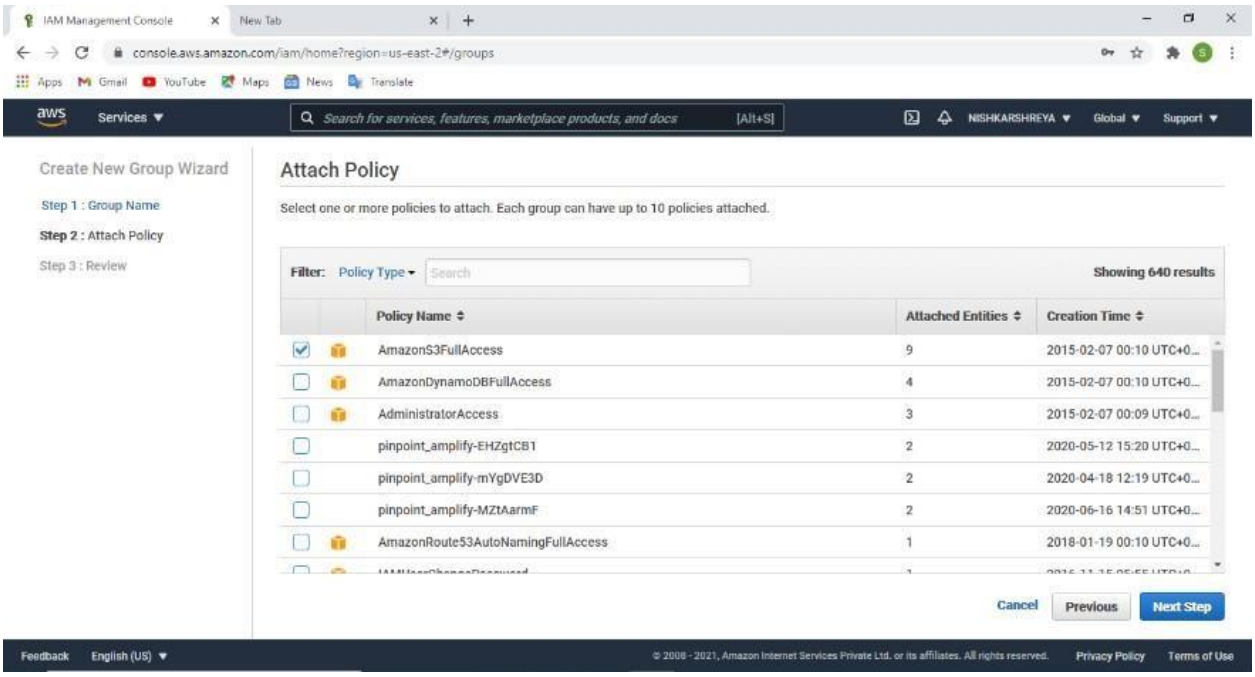
2. Steps to create an IAM group
   - Click on the Groups →Go to Create group.



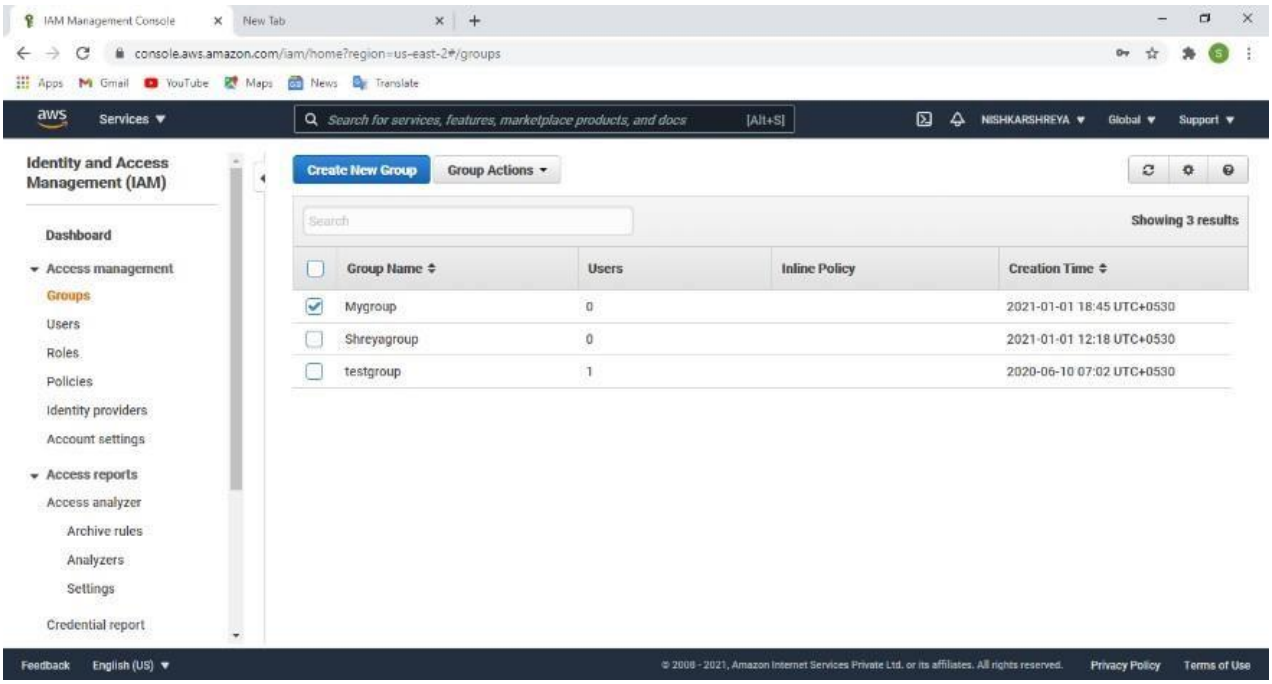
   - Give group name →next step. Give permissions /attach policies to the group.



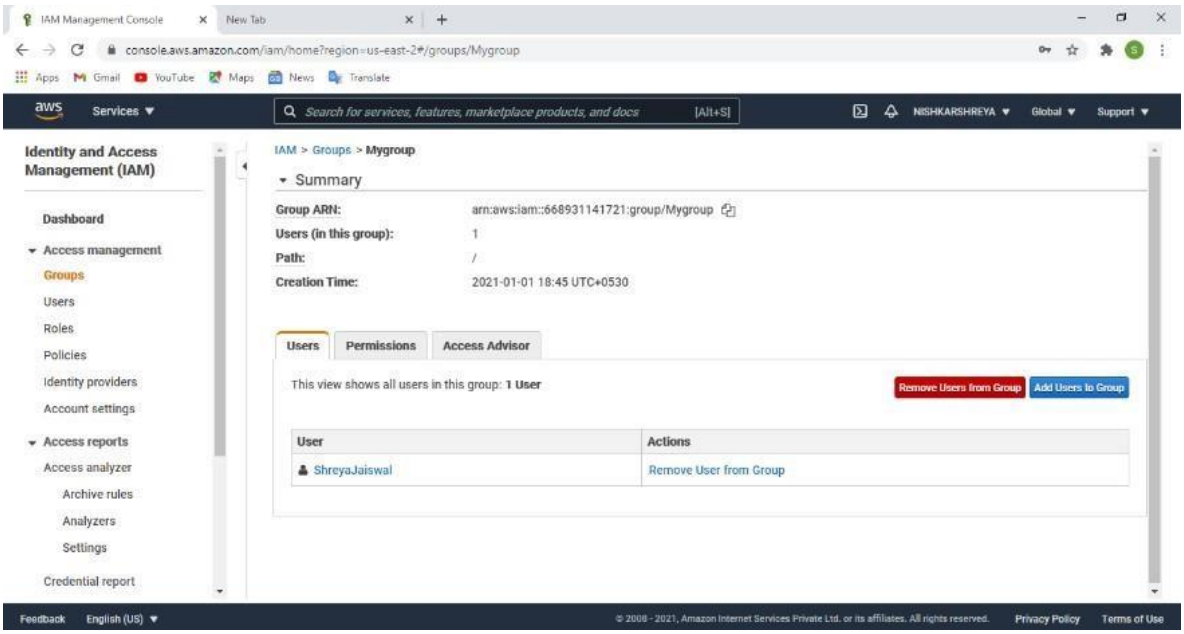
- Click on the next step (check group configuration and make changes if needed).



- Click on create group, group successfully created.



- By default, an IAM group does not have any IAM user we have to add a user to it and remove the user if required.
- To add IAM user in IAM group. Inside your IAM group that you have created →go to Users →click on Add Users to Group → click to Add User. User successfully added.



Conclusion:

In this Practical, we have learnt how to Create and Setup Amazon Elastic Compute Cloud and how to Create and setup monitoring service for AWS cloud resources and the applications you run on AWS and also Create an AWS Identity and Access Management (IAM) group and user, attach a policy and add a user to a group.

PRACTICAL: 10

AIM:

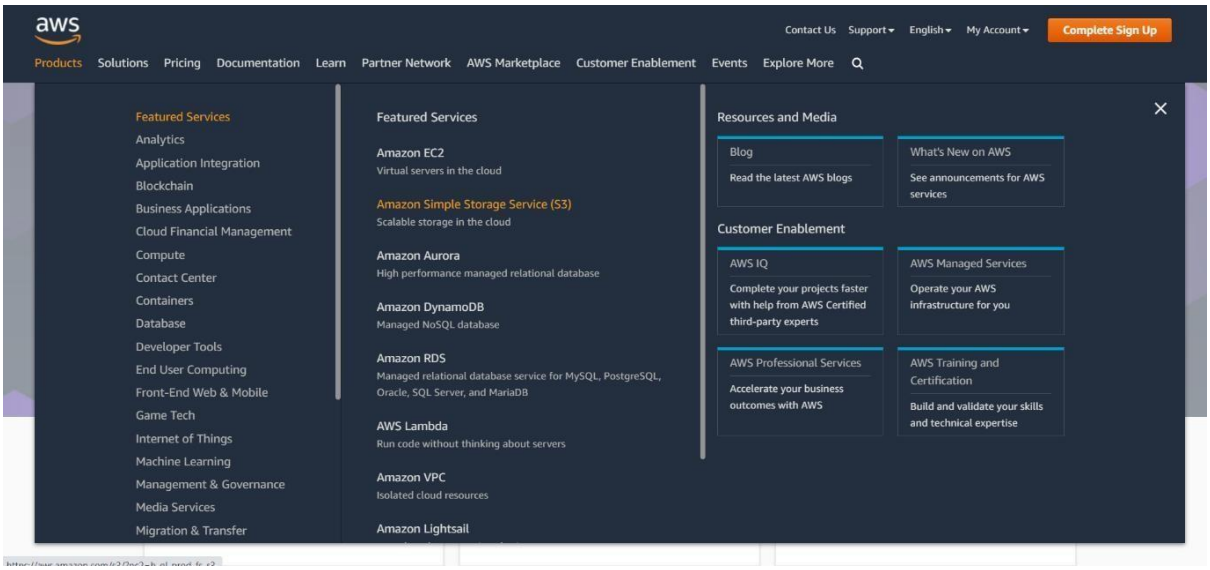
Create and setup Amazon Simple Storage Service (Amazon S3) Block Public Access on Amazon Cloud Platform.

THEORY:

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

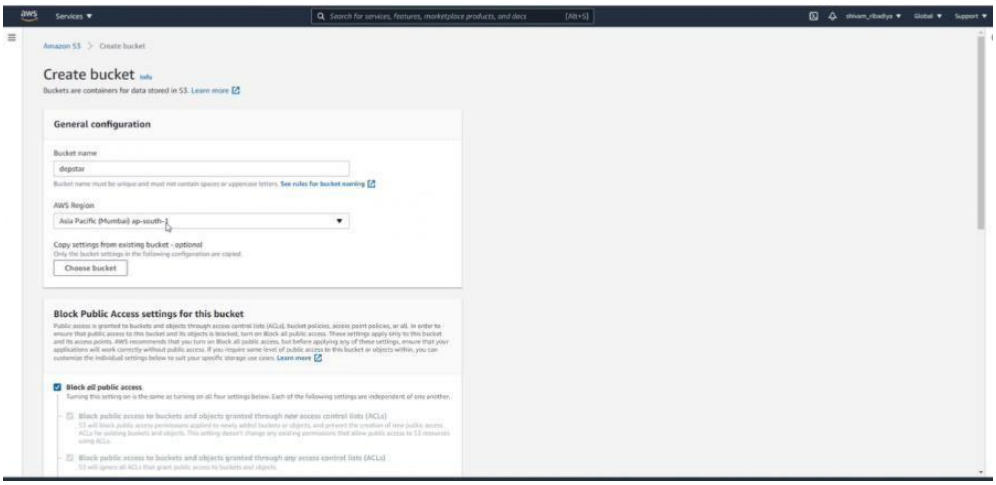
The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. As part of the subscription agreement, Amazon provides security for subscribers' systems. AWS operates from many global geographical regions including 6 in North America.

Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2017, AWS owns 33% of all cloud (IaaS, PaaS) while the next two competitors Microsoft Azure and Google Cloud have 18%, and 9% respectively, according to Synergy Group.



PRACTICAL:

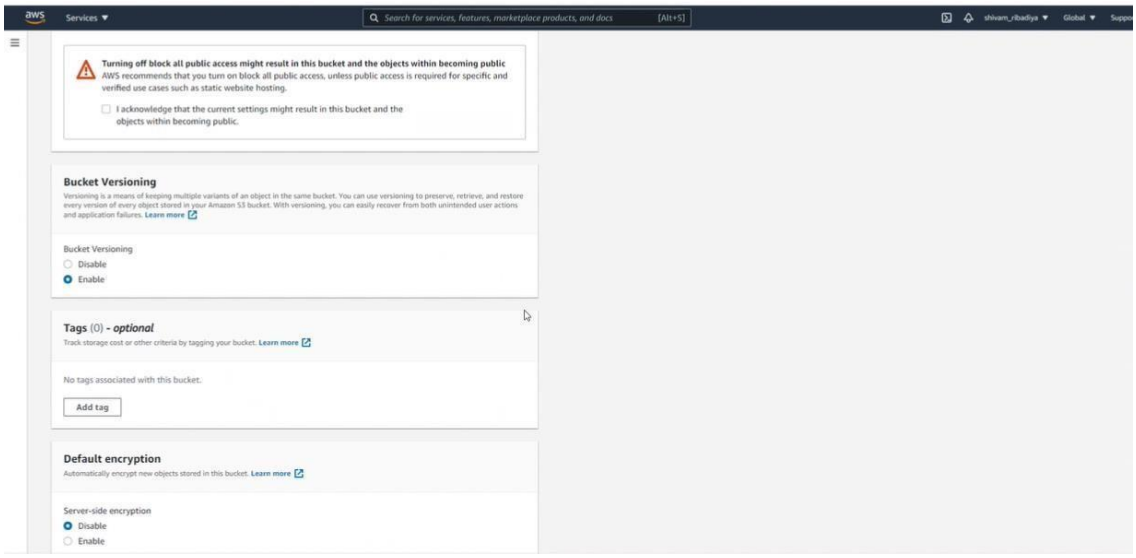
Login to your AWS account and go to products and select Amazon Simple Storage Service (S3).
Before you begin hosting your awesome static website out of S3, you need a bucket first. For this blog post, it is critical that your bucket has the same name as your domain name.
If your website domain is www.my-awesome-site.com, then your bucket name must be www.my-awesome-site.com.
The reasoning for this has to do with how requests are routed to S3. The request comes into the bucket, and then S3 uses the Host header in the request to route to the appropriate bucket.



Alright, you have your bucket. It has the same name as your domain name, yes? Time to configure the bucket for static website hosting.

- Navigate to S3 in the AWS Console.
- Click into your bucket.
- Click the "Properties" section.
- Click the "Static website hosting" option.
- Select "Use this bucket to host a website".

- Enter “index.html” as the Index document.



Your bucket is configured for static website hosting, and you now have an S3 website url like this http://www.my-awesome-site.com.s3-website-us-east-1.amazonaws.com/.

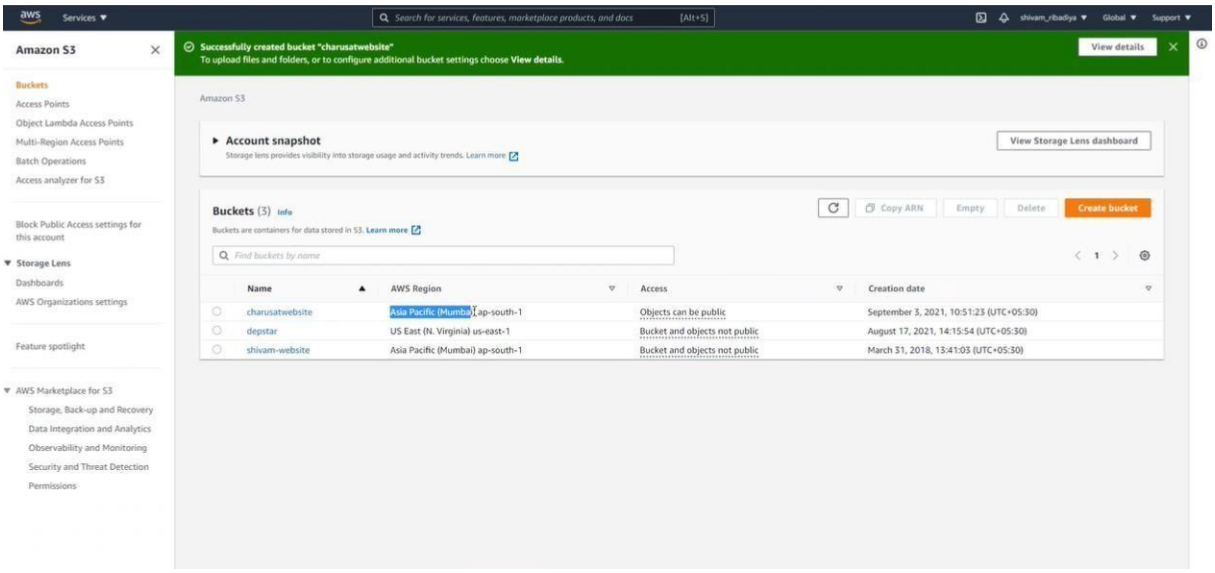
By default, any new buckets created in an AWS account deny you the ability to add a public access bucket policy. This is in response to the recent leaky buckets where private information has been exposed to bad actors. However, for our use case, we need a public access bucket policy. To allow this you must complete the following steps before adding your bucket policy.

- Click into your bucket.
- Select the “Permissions” tab at the top.
- Under “Public Access Settings” we want to click “Edit”.
- Change “Block new public bucket policies”, “Block public and cross-account access if bucket has public policies”, and “Block new public ACLs and uploading public objects” to be false and Save.

Now you must update the Bucket Policy of your bucket to have public read access to anyone in the world. The steps to update the policy of your bucket in the AWS Console are as follows:

- Navigate to S3 in the AWS Console.
- Click into your bucket.
- Click the “Permissions” section.
- Select “Bucket Policy”.
- Add the following Bucket Policy and then Save

Remember S3 is a flat object store, which means each object in the bucket represents a key without any hierarchy. While the AWS S3 Console makes you believe there is a directory structure, there isn’t. Everything stored in S3 is keys with prefixes.



Conclusion:

In this practical, we learnt about AWS and hosted our static website on AWS using S3 service.

PRACTICAL: 11

AIM:

Create and deploy project using AWS Amplify Hosting Service of AWS.

THEORY:

Amazon Web Services are some of the most useful products we have access to. One such service that is becoming increasingly popular as days go by is AWS Amplify. It was released in 2018 and it runs on Amazon's cloud infrastructure. It is in direct competition with Firebase, but there are features that set them apart.

Why is it needed?

User experience on any application is the most important aspect that needs to be taken care of. AWS Amplify helps unify user experience across platforms such as web and mobile. This makes it easier for a user to choose which one would they be more comfortable with. It is useful in case of front end development as it helps in building and deployment. Many who use it claim that it actually makes full-stack development a lot easier with its scalability.

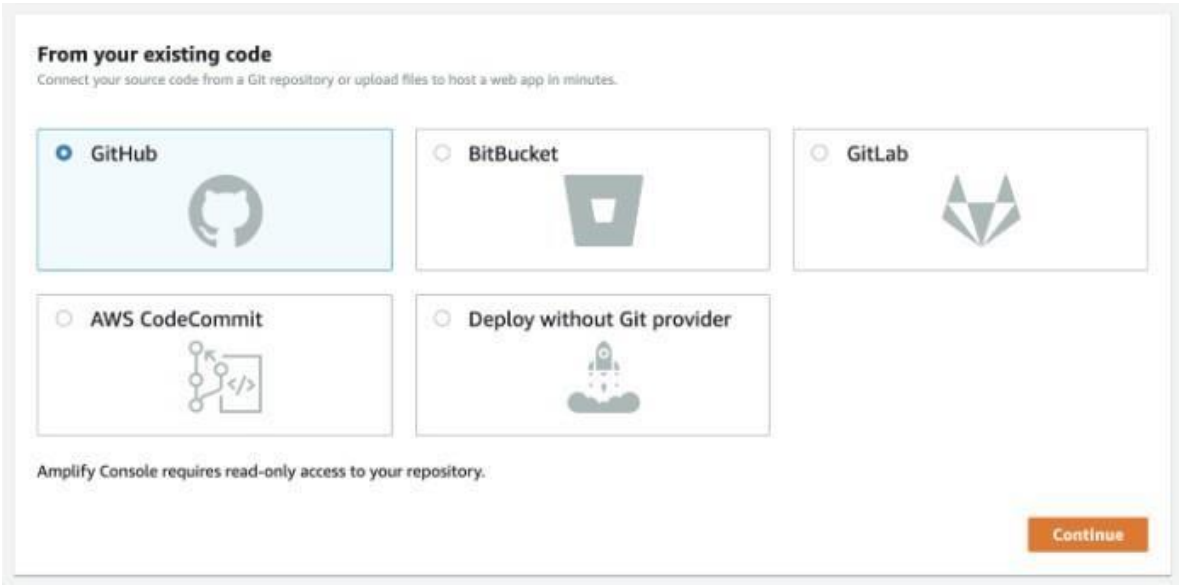
Main features:

- Can be used for authenticating users which are powered by Amazon Cognito.
- With help from Amazon AppSync and Amazon S3, it can securely store and sync data seamlessly between applications.
- As it is serverless, making changes to any back-end related cases has become simpler. Hence, less time is spent on maintaining and configuring back-end features.
- It also allows for offline synchronization.
- It promotes faster app development.
- It is useful for implementing Machine Learning and AI-related requirements as it is powered by Amazon Machine learning services.
- It is useful for continuous deployment.
- Various AWS services are used for various functionalities. AWS Amplify offers. The main components are libraries, UI components, and the CLI tool chain. It also provides static web hosting using AWS Amplify Console.

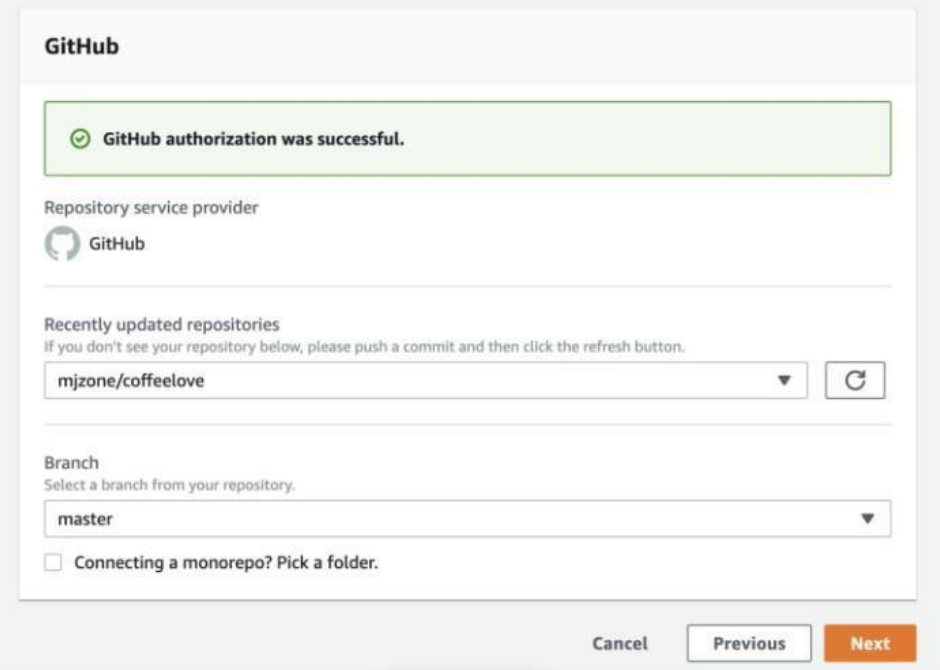
Task 1: Log in to the AWS Amplify Console and choose Get Started under Deploy.

Task 2: Connect your Code Repository

Connect a branch from your GitHub, Bitbucket, GitLab, or AWS Code Commit repository. Connecting your repository allows Amplify to deploy updates on every code commit to a branch.

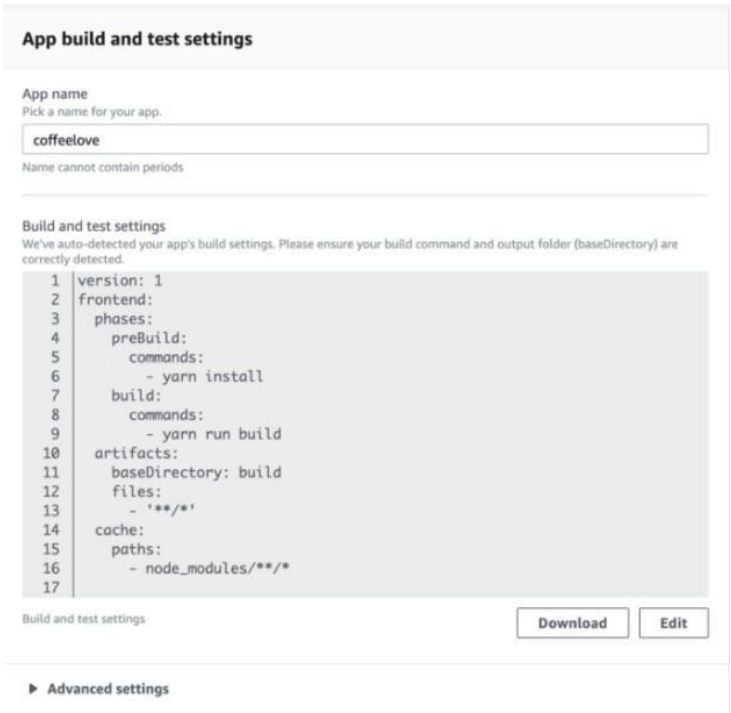


Task 3: Adding the Repo Branch



Task 4: Configure Build Settings

Accept the default build settings. Give the Amplify Console permission to deploy backend resources with your frontend with a service role. This allows the Console to detect changes to both your backend and frontend on every code commit and make updates. If you do not have a service role follow the prompts to create one, then come back to the console and pick it from the dropdown.



Task 5: Save & Deploy

Review your changes and then choose **Save and deploy**. The Amplify Console will pull code from your repository, build changes to the backend and frontend, and deploy your build artifacts at https://master.unique-id.amplifyapp.com. Bonus: Screenshots of your app on different devices to find layout issues

Conclusion:

In this Practical, we have learnt how to Create and deploy project using AWS Amplify Hosting Service of AWS.

.

PRACTICAL: 12

AIM:

Simulating networks using iFogSim.

THEORY:

iFogSim is a java programming language based API that inherits the established API of Cloudsim to manage its underlying discrete event-based simulation. It also utilizes the API of CloudsimSDN for relevant network-related workload handling.

iFogSim Simulation Toolkit is another simulator used for the implementation of the Fog computing-related research problem.

This course will help you to follow the simulation-based approach of iFogSim and can leverage various benefits like:

1. Testing of services and scenarios in a controllable environment of iFogSim.
2. Optimizing the core system performance issues with the iFogSim simulation engine before deploying on real infrastructure.
3. iFogSim allows simulating the small or large scale infrastructure to evaluate different sets of workload along with the resource performance. This facilitates the development, testing, and deployment of adaptive application provisioning techniques.

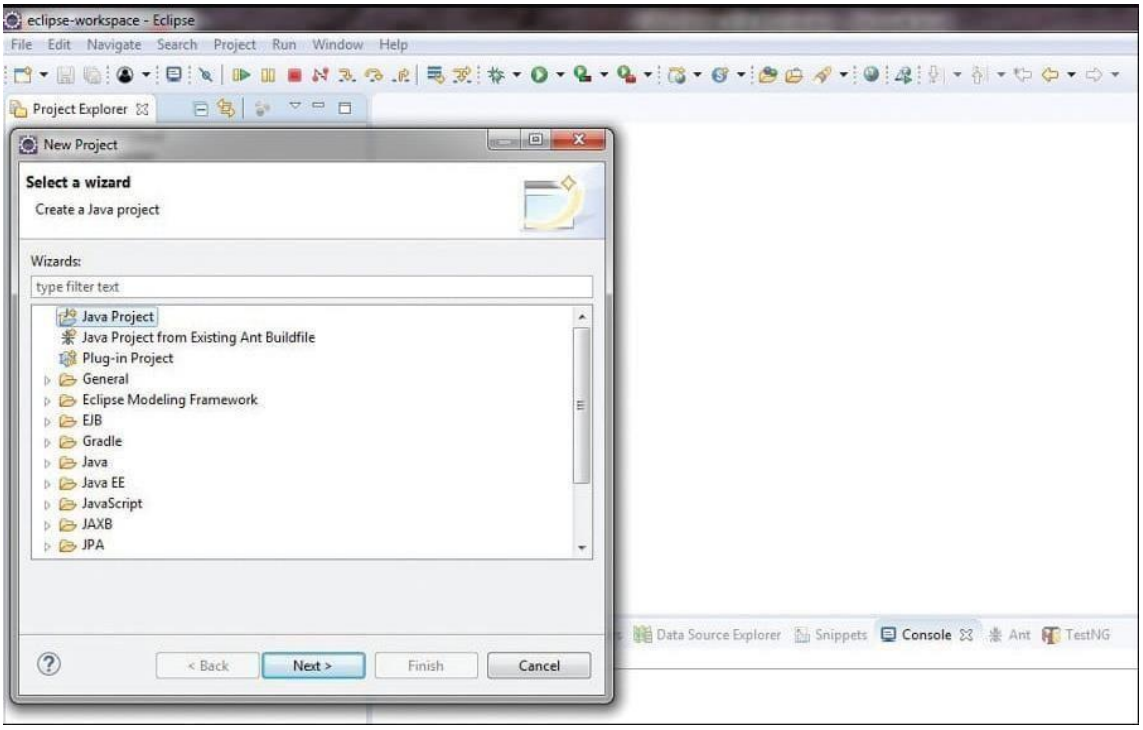
iFogSim simulator possesses a huge potential to simulate the research-based use case and then corresponding to the promising results can be deployed to the existing system with minimum cost and efforts involved.

Installing iFogSim

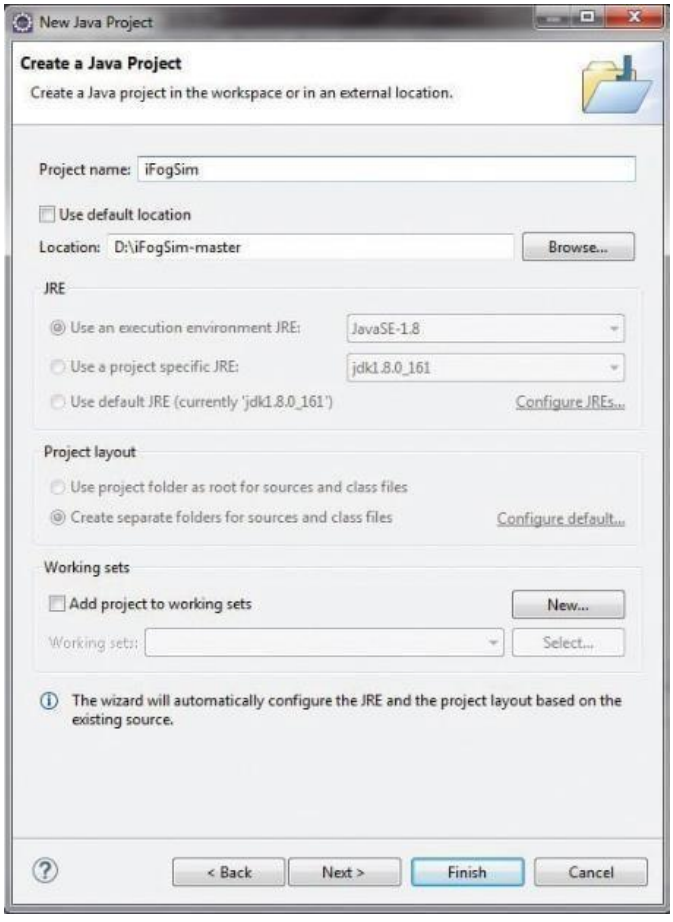
The iFogSim library can be downloaded from the URL *https://github.com/Cloudslab/iFogSim*. This library is written in Java, and therefore the Java Development Kit (JDK) will be required to customise and work with the toolkit.

After downloading the compression toolkit in the Zip format, it is extracted and a folder *iFogSim-master* is created. The iFogSim library can be executed on any Java based integrated development environment (IDE) like Eclipse, Netbeans, JCreator, JDeveloper, jGRASP, BlueJ, IntelliJ IDEA or Jbuilder.

In order to integrate iFogSim on an Eclipse ID, we need to create a new project in the IDE

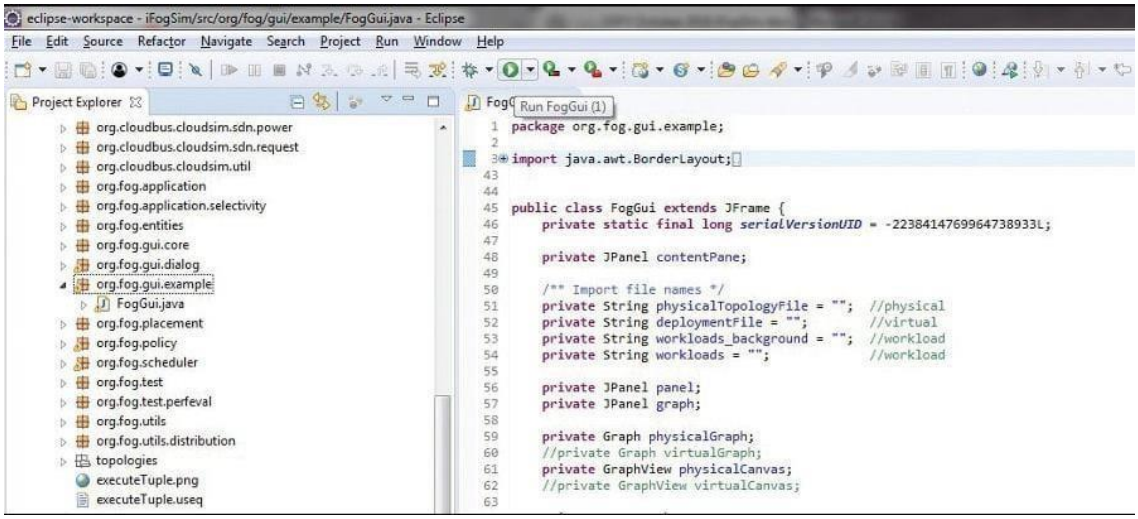


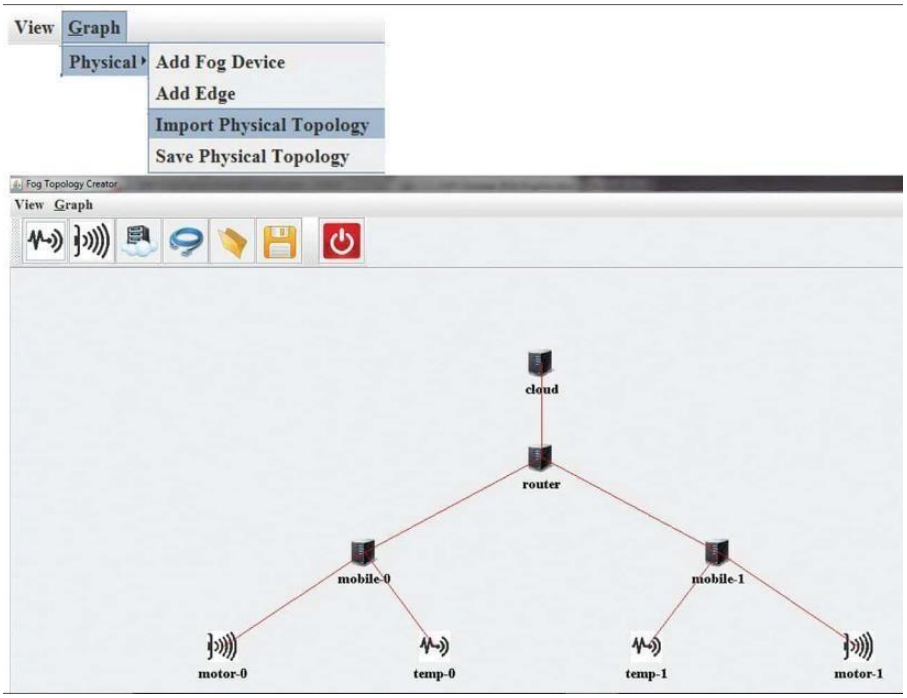
Creating a new project in the Eclipse IDE



Simulating networks using iFogSim

Once the library is set up, the directory structure of iFogSim can be viewed in the Eclipse IDE in *Project Name -> src*.
There are numerous packages with Java code for different implementations of fog computing, IoT and edge computing.
To work with iFogSim in the graphical user interface (GUI) mode, there is a file called *FogGUI.java* in *org.fog.gui.example*. This file can be directly executed in the IDE, and there aredifferent cloud and fog components that can be imported in the simulation working area as shown in Figure 3.
In *Fog Topology Creator*, there is a *Graph* menu, where there is the option to import the topology





Conclusion:

In this Practical, we have learnt how to Simulating networks using iFogSim.

PRACTICAL: 13

AIM:

A Comparative Study of Docker Engine on Windows Server vs Linux Platform Comparing the feature sets and implementations of Docker on Windows and Linux and Build and Run Your First Docker Windows Server Container Walkthrough installing Docker on Windows 10, building a Docker image and running a Windows container.

THEORY:

What does it mean to Windows community?

It means that Windows Server 2016 natively supports Docker containers now on-wards and offers two deployment options – **Windows Server Containers** and **Hyper-V Containers**, which offer an additional level of isolation for multi-tenant environments.The extensive partnership integrates across the Microsoft portfolio of developer tools, operating systems and cloud infrastructure including:

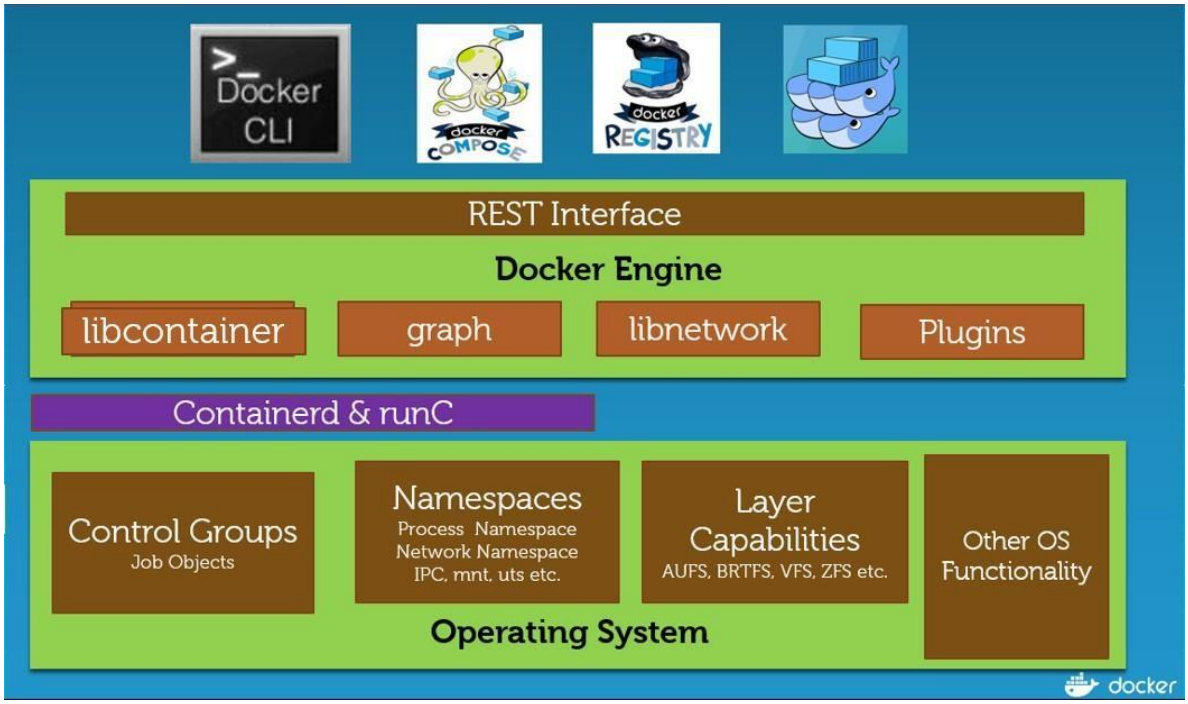
- Windows Server 2016
- Hyper-V
- Visual Studio
- Microsoft Azure

What does it mean to Linux enthusiasts?

In case you are Linux enthusiast like me, you must be curious to know how different does Docker Engine on Windows Server Platform work in comparison to Linux Platform. Under this post, I am going to spend considerable amount of time talking about architectural difference, CLI which works under both the platform and further details about Dockerfile, docker compose and the state of Docker Swarm under Windows Platform.

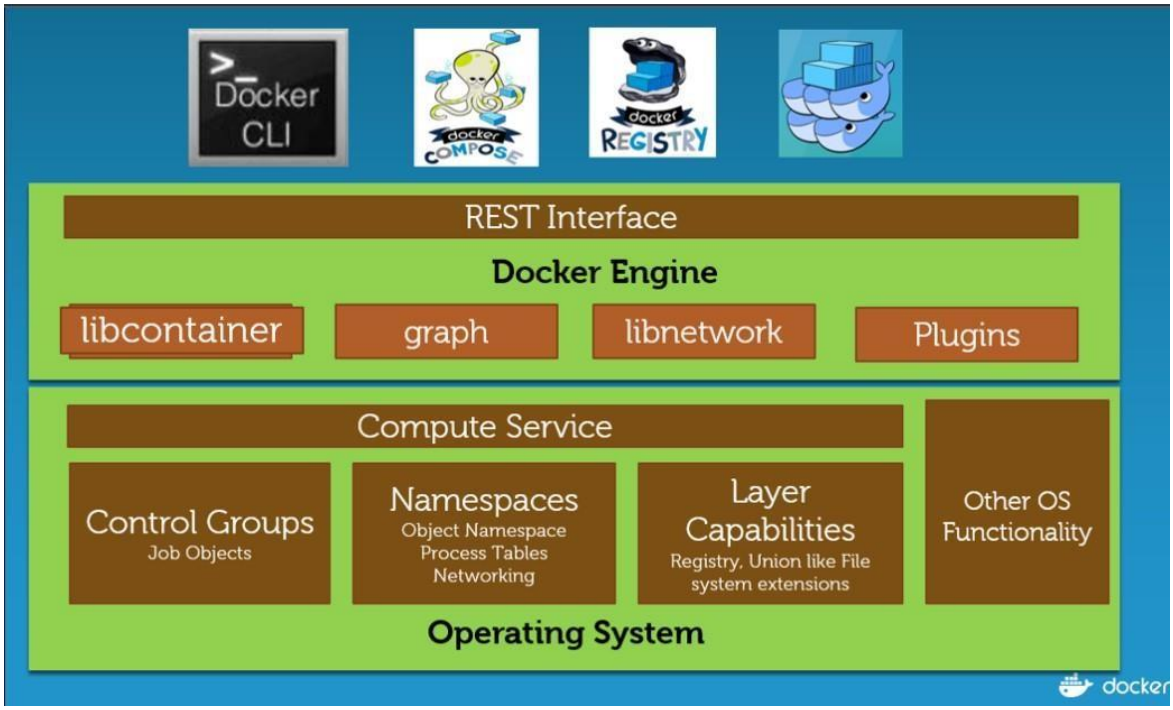
Let us first talk about architectural difference of Windows containers Vs Linux containers.

Looking at Docker Engine on Linux architecture, sitting on the top are CLI tools like Docker compose, Docker Client CLI, Docker Registry etc. which talks to Docker REST API. Users communicates and interacts with the Docker Engine and in turn, engine communicates with containerd. Containerd spins up runC or other OCI compliant run time to run containers. At the bottom of the architecture, there are underlying kernel features like namespaces which provides isolation and control groups etc. which implements resource accounting and limiting, providing many useful metrics, but they also help ensure that each container gets its fair share of memory, CPU, disk I/O; and, more importantly, that a single container cannot bring the system down by exhausting one of those resources.



Docker Engine on Linux Platform

Under Windows, it's slightly a different story. The architecture looks same for the most of the top level components like same Remote API, same working tools (Docker Compose, Swarm) but as we move down, the architecture looks different. In case you are new to Windows kernel, the Kernel within the Windows is somewhat different than that of Linux because Microsoft takes somewhat different approach to the Kernel's design. The term "Kernel mode" in Microsoft language refers to not only the Kernel itself but the HAL*(hal.dll)* and various system services as well. Various managers for Objects, processes, Memory, Security, Cache, Plug in Play (PnP), Power, Configuration and I/O collectively called Windows Executive(ntoskrnl.exe) are available. **There is no kernel feature specifically called namespace and cgroup on Windows**. Instead, Microsoft team came up with new version of Windows Server 2016 introducing "Compute Service Layer" at OS level which provides namespace, resource control and UFS like capabilities. Also, as you see below, **there is NO containerd and runC concept available under Windows Platform**. Compute Service Layer provides public interface to container and does the responsibility of managing the containers like starting and stopping containers but it doesn't maintain the state as such. In short, it replaces containerd on windows and abstracts low level capabilities which the kernel provides.



Getting Started with Docker on Windows 2016 Server

You need Windows 2016 Server Evaluation build 14393 or later to taste the newer Docker Engine on Win2k16. If you try to follow the usual Docker installation process on your old Windows 2016 TP5 system, you will get the following error

```
PS C:\Users\depstarcsed> dockerd
Error starting daemon: The docker daemon requires build 14393 or later of Windows Server 2016 or Windows 10
```

Start-Service Docker

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\depstarcsed> docker images
error during connect: Get http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.25/images/json: open //./pipe/docker_engine: The syst
em cannot find the file specified.
PS C:\Users\depstarcsed> docker version
Client:
 Version:      1.12.2-cs2-ws-beta
 API version:  1.25
 Go version:   go1.7.1
 Git commit:   050b611
 Built:        Tue Oct 11 02:35:40 2016
 OS/Arch:      windows/amd64
error during connect: Get http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.25/version: open //./pipe/docker_engine: The system c
annot find the file specified.
PS C:\Users\depstarcsed> Start-Service Docker
PS C:\Users\depstarcsed> docker version
Client:
 Version:      1.12.2-cs2-ws-beta
 API version:  1.25
 Go version:   go1.7.1
 Git commit:   050b611
 Built:        Tue Oct 11 02:35:40 2016
 OS/Arch:      windows/amd64

Server:
 Version:      1.12.2-cs2-ws-beta
 API version:  1.25
 Go version:   go1.7.1
 Git commit:   050b611
 Built:        Tue Oct 11 02:35:40 2016
 OS/Arch:      windows/amd64
```

Now you can search plenty of Windows Dockerized application using the below command:

```
PS C:\Users\depstarcsed> docker search windows
NAME                                        DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
microsoft/mssql-server-2014-express-windows Microsoft SQL Server 2014 Express installe...  37
thewtex/cross-compiler-windows-x64          64-bit Windows cross-compiler based on MXE...   20                   [OK]
thewtex/cross-compiler-windows-x86          32-bit Windows cross-compiler based on MXE...   5                    [OK]
libretro/build-windows                      Build image for libretro windows                3
stefanscherer/swarm-windows                 Swarm: a Docker-native clustering system. ...   2
bosh/windows                                                                                2
devacto/windows-machine                     From Matt Fellow's Windows machine factory...   1
msolution/repos-windows                                                                     1
stefanscherer/node-windows                  Node.js is a JavaScript-based platform for...   1
dockcross/windows-x86                                                                       1
dockcross/windows-x64                                                                       1
calvinpark/arc-toolchain-builder-windows                                                    0                    [OK]
coderobin/windows-sdk-10.1                  Windows SDK 10.1 for Windows Container (wi...   0                    [OK]
toktoknet/windows-qt5                       Windows cross compilers with Qt5 pre-built      0
stefanscherer/registry-windows              Containerized docker registry for Windows ...   0
mazzolino/twister-core-windows-docker                                                       0                    [OK]
coderobin/windows-build-uwp-461             Windows Build container for UWP 10.0 and ....   0                    [OK]
florentbenoit/che-installer-windows         Allow to get installer dependencies and co...   0                    [OK]
inginform/devenv-sammler-windows            Lädt alle Programme die von der IIF Entwic...   0                    [OK]
toktoknet/windows                           Windows cross compilers (i686 and x86_64)       0
avalverde/ghc-cross-compiler-windows-x86    i686-w64-mingw32 GHC 8.0.1 cross-compiler ...   0                    [OK]
cdrx/pyinstaller-windows                    PyInstaller for Windows inside Docker (usi...   0                    [OK]
bit3/jsass-build-windows-x64                Build jsass natives for windows x86_64 arc...   0                    [OK]
bit3/jsass-build-windows-x32                Build jsass natives for windows x386 archi...   0                    [OK]
jibyjose/swarm-windows                      swarm built for windows server 2016 tp5         0
```

```
PS C:\Users\depstarcsed> docker search microsoft
NAME                                        DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
microsoft/aspnet                            ASP.NET is an open source server-side Web ...   483                  [OK]
microsoft/dotnet                            Official images for working with .NET Core...   290                  [OK]
mono                                        Mono is an open source implementation of M...   192       [OK]
microsoft/azure-cli                         Docker image for Microsoft Azure Command L...   64                   [OK]
microsoft/nanoserver                        Windows Server 2016 Nano Server base OS im...   43
microsoft/windowsservercore                 Windows Server 2016 Server Core base OS im...   38
microsoft/mssql-server-2014-express-windows Microsoft SQL Server 2014 Express installe...   35
microsoft/iis                               Internet Information Services (IIS) instal...   35
microsoft/aspnetcore                        Official images for running compiled ASP.N...   8                    [OK]
microsoft/oms                               Monitor your containers using the Operatio...   6                    [OK]
microsoft/dotnet35                                                                          4
microsoft/sample-nginx                      Nginx installed in Windows Server Core and...   4
microsoft/applicationinsights               Application Insights for Docker helps you ...   3                    [OK]
microsoft/aspnetcore-build                  Official images for building ASP.NET Core ...   3                    [OK]
microsoft/sample-httpd                      Apache httpd installed in Windows Server C...   3
microsoft/mssql-server-2016-developer-windows Microsoft SQL Server 2016 Developer instal...  2
microsoft/dotnet-nightly                    Preview bits of the .NET Core CLI               2                    [OK]
microsoft/sample-redis                      Redis installed in Windows Server Core and...   2
5496abde368a: Extracting [==============>                                  ] 72.97 MB/242.6 MB
5496abde368a: Extracting [===>                                             ] 16.15 MB/242.6 MB
microsoft/sample-dotnet                     .NET Core running in a Nano Server container    1
microsoft/mssql-server-2016-express-windows Microsoft SQL Server 2016 Express installe...   1
microsoft/vsts-agent                                                                        0
```

Important Points:

1. Linux containers doesn't work on Windows Platform.(see below)

```
PS C:\Users\depstarcsed> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu

image operating system "linux" cannot be used on this platform
```

2. DTR is still not supported on Windows Platform

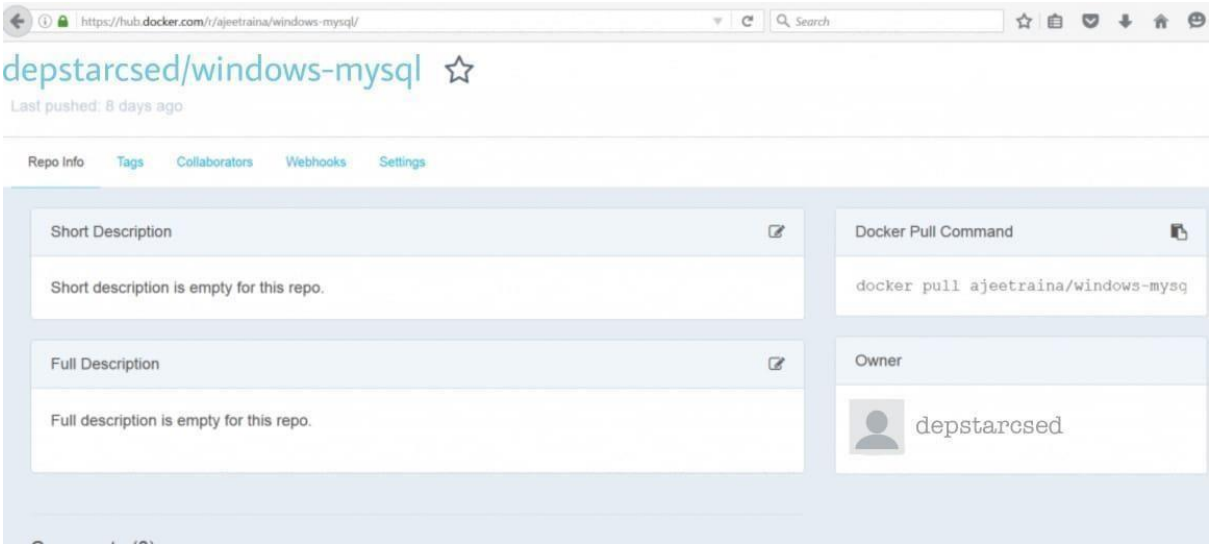
3. You can't commit a running container and build image out of it. (This is very much possible on Linux Platform.

Using Docker file for MySQL

Building containers using Dockerfile is supported on Windows server platform. Let's pick up a sample MySQL Dockerfile to build up MySQL container. I found it available on some github repository and want to see if Dockerfile is supported or not. The sample Dockerfile looks somewhat like as shown below:

```
FROM microsoft/windowsservercore
LABEL Description="MySql" Vendor="Oracle" Version="5.6.29″
RUN powershell -Command \
$ErrorActionPreference = 'Stop'; \
Invoke-WebRequest -Method Get -Uri https://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-5.6.29-winx64.zip -OutFile c:\mysql.zip ; \
Expand-Archive -Path c:\mysql.zip -DestinationPath c:\ ; \
Remove-Item c:\mysql.zip -Force
RUN SETX /M Path %path%;C:\mysql-5.6.29-winx64\bin
RUN powershell -Command \
$ErrorActionPreference = 'Stop'; \
mysqld.exe –install ; \
Start-Service mysql ; \
Stop-Service mysql ; \
Start-Service mysql
RUN type NUL > C:\mysql-5.6.29-winx64\bin\foo.mysql
RUN echo UPDATE user SET Password=PASSWORD('mysql123′) WHERE User='root'; FLUSH PRIVILEGES; .> C:\mysql-5.6.29-winx64\bin\foo.mysql
RUN mysql -u root mysql < C:\mysql-5.6.29-winx64\bin\foo.mysql
```

This just brings up the MySQL image perfectly. I had my own version of MySQL Dockerized image available which is still under progress. I still need to populate the Docker image details.



Conclusion:

In this Practical, we have learnt how to Docker Engine works on windows and linux operating system and instating also building a docker image and running a windows container.