

## PROGRAM-1

### AIM: Implement Bankers algorithm (using Python /Java)

#### Banker's Algorithm:

In Java, Banker's algorithm is a deadlock avoidance and resource allocation algorithm. This algorithm tests for security by simulating allocation for a predetermined maximum possible amount of all resources. After that, before deciding whether the allocation should be allowed to continue or not, it creates an "s-state" check for testing it to all possible activities.

Program:

```
// import required classes and packages
import java.util.*;
import java.io.*;
import java.util.Scanner;

// create BankersAlgoExample class to implement Banker's algorithm in Java
class BankersAlgoExample
{
    // create findNeedValue() method to calculate the need of each process
    static void findNeedValue(int needArray[], int maxArray[], int allocationArray[], int
totalProcess, int totalResources)
    {
        // use nested for loop to calculate Need for each process
        for (int i = 0 ; i < totalProcess ; i++){ // for each process

            for (int j = 0 ; j < totalResources ; j++){ //for each resource
                needArray[i][j] = maxArray[i][j] - allocationArray[i][j];
            }
        }
    }
    // create checkSafeSystem() method to determine whether the system is in safe state or not
    static boolean checkSafeSystem(int processes[], int availableArray[], int maxArray[], int
allocationArray[], int totalProcess, int totalResources)
    {
        for (j = 0; j < totalResources; j++)
            if (needArray[m][j] > workArray[j]) //check need of current resource for current
process with work
                break;

        // the value of J and totalResources will be equal when all the needs of current process
are satisfied
        if (j == totalResources)
        {
            for (int k = 0 ; k < totalResources ; k++)
                workArray[k] +=
                allocationArray[m][k];
        }
    }
}
```

```

int [][]needArray = new int[totalProcess][totalResources]

// call findNeedValue() method to calculate needArray
findNeedValue(needArray, maxArray, allocationArray, totalProcess, totalResources);

// all the process should be in finished in starting
boolean []finishProcesses = new boolean[totalProcess];

// initialize safeSequenceArray that store safe sequenced int
[]safeSequenceArray = new int[totalProcess];

// initialize workArray as a copy of the available resources
int []workArray = new int[totalResources];

for (int i = 0; i < totalResources ; i++) //use for loop to copy each available resource in the
workArray
    workArray[i] = availableArray[i];

// initialize counter variable whose value will be 0 when the system is not in the safe state or when
all the processes are not finished.
int counter = 0;

// use loop to iterate the statements until all the processes are not finished
while (counter < totalProcess)

{
    // find in finished process which needs can be satisfied with the current work resource.
    boolean foundSafeSystem = false;
    for (int m = 0; m < totalProcess; m++)
    {
        if (finishProcesses[m] == false)    // when process is not finished
        {
            int j;

//use for loop to check whether the need of each process for all the resources is less than

                // add current process in the safeSequenceArray
                safeSequenceArray[counter++] = m;
                // make this process finished
                finishProcesses[m] = true;

                foundSafeSystem = true;
            }
        }
    }
}

// the system will not be in the safe state when the value of the foundSafeSystem is false if

```

```

        (foundSafeSystem == false)
        {
            System.out.print("The system is not in the safe state because lack of resources");return
            false;
        }
    }
    // print the safe sequence
    System.out.print("The system is in safe sequence and the sequence is as follows: ");for
    (int i = 0; i < totalProcess ; i++)
        System.out.print("P"+safeSequenceArray[i] + " ");

    return true;
}
// main() method start
public static void main(String[] args)
{
    int numberOfProcesses, numberOfResources;
    //create scanner class object to get input from user
    Scanner sc = new Scanner(System.in);

    // get total number of resources from the user
    System.out.println("Enter total number of processes");
    numberOfProcesses = sc.nextInt();

    // get total number of resources from the user
    System.out.println("Enter total number of resources");
    numberOfResources = sc.nextInt();
    int processes[] = new int[numberOfProcesses];
    for(int i = 0; i < numberOfProcesses; i++){
        processes[i] = i;
    }

    int availableArray[] = new

    int[numberOfResources]; for( int i = 0; i <
    numberOfResources; i++){
        System.out.println("Enter the availability of resource"+ i +": ");
        availableArray[i] = sc.nextInt();
    }

    int maxArray[][] = new int[numberOfProcesses][numberOfResources];
    for( int i = 0; i < numberOfProcesses; i++){
        for( int j = 0; j < numberOfResources; j++){

            System.out.println("Enter the maximum resource"+ j +" that can be allocated to process"+ i
            maxArray[i][j] = sc.nextInt());

```

```

    }
}

int allocationArray[][] = new int[numberOfProcesses][numberOfResources];
for( int i = 0; i < numberOfProcesses; i++){
    for( int j = 0; j < numberOfResources; j++){
        System.out.println("How many instances of resource"+ j +" are allocated to process"+ i +"?

        allocationArray[i][j] = sc.nextInt();

    }
}

//call checkSafeSystem() method to check whether the system is in safe state or not
checkSafeSystem(processes, availableArray, maxArray, allocationArray, numberOfProcesses,
numberOfResources);
}
}

```

## Output:

```
Azure Cloud Shell
rohit@Azure:~$ java BankersAlgoExample
Enter total number of processes
5
Enter total number of resources
3
Enter the availability of resource0:
3
Enter the availability of resource1:
3
Enter the availability of resource2:
2
Enter the maximum resource0 that can be allocated to process0:
7
Enter the maximum resource1 that can be allocated to process0:
5
Enter the maximum resource2 that can be allocated to process0:
3
Enter the maximum resource0 that can be allocated to process1:
3
Enter the maximum resource1 that can be allocated to process1:
2
Enter the maximum resource2 that can be allocated to process1:
2
Enter the maximum resource0 that can be allocated to process2:
9
Enter the maximum resource1 that can be allocated to process2:
0
Enter the maximum resource2 that can be allocated to process2:
2
Enter the maximum resource0 that can be allocated to process3:
2
Enter the maximum resource1 that can be allocated to process3:
2
Enter the maximum resource2 that can be allocated to process3:
```

```

Enter the maximum resource0 that can be allocated to process3:
2
Enter the maximum resource1 that can be allocated to process3:
2
Enter the maximum resource2 that can be allocated to process3:
2
Enter the maximum resource0 that can be allocated to process4:
4
Enter the maximum resource1 that can be allocated to process4:
3
Enter the maximum resource2 that can be allocated to process4:
3
How many instances of resource0 are allocated to process0?
0
How many instances of resource1 are allocated to process0?
1
How many instances of resource2 are allocated to process0?
0
How many instances of resource0 are allocated to process1?
2
How many instances of resource1 are allocated to process1?
0
How many instances of resource2 are allocated to process1?
0
How many instances of resource0 are allocated to process2?
3
How many instances of resource1 are allocated to process2?
0
How many instances of resource2 are allocated to process2?
2
How many instances of resource0 are allocated to process3?
2
How many instances of resource1 are allocated to process3?
1
How many instances of resource2 are allocated to process3?
1
How many instances of resource0 are allocated to process4?
0
How many instances of resource1 are allocated to process4?
0
How many instances of resource2 are allocated to process4?
2
The system is in safe sequence and the sequence is as follows: P1 P3 P4
P0 P2 rohit@Azure:~$ |

```

## PROGRAM-2

### AIM: Implement RPC Algorithm (using Python /Java)RPC:

RPC stands for Remote Procedure Call which supports procedural programming. Using RPC, we can invoke methods in shared environments.

As an example, we can call a function in a remote machine from our local computer using RPC. We can define RPC as a communication type in distributed systems.

When we dig into RPC structure, we can identify it implements the client server model. And also, the calls are synchronous which makes the client wait till the server response.

#### Program:

```
myInterface
.java import
java.rmi.*;
public interface MyInterface extends Remote
{
    public String countInput(String input)throws RemoteException;
}

RMIServer.
.java import
java.rmi.*;
import java.rmi.server.*;
public class RMIServer extends UnicastRemoteObject implements MyInterface
{
    public RMIServer()throws RemoteException
    {
        System.out.println("Remote Server is running Now.!!");
    }
    public static void main(String arg[])
    {
        try{
            RMIServer      p=new
            RMIServer();
            Naming.rebind("rmiInterfa
            ce",p);
        }
        catch(Exception e)
        { System.out.println("Exception occurred : "+e.getMessage()); }
    }

    @Override
    public String countInput(String input) throws RemoteException
```

```

        System.out.println("Received your input "+ input+" at
        server!!");String reply;
        reply="You have typed "+ input.length() +"
        letters!!";return reply;
    }
}

```

## RMI Client.java

```

import java.rmi.*;
import java.io.*;
public class
RMIClient
{
    public static void main(String args[])
    {
        try
        {
            BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
            MyInterface p=(
            MyInterface)Naming.lookup("rmiInterface");
            System.out.println("Type something...");

            String input=br.readLine();
            System.out.println(p.countInput(input));
        }
        catch(Exception e) {
            System.out.println("Exception occurred : "+e.getMessage());
        }
    }
}

```

## Output:



```

E:\rmi>java RMIServer
Remote Server is running Now.!!

```



```

E:\rmi>java RMIClient
Type something...

```



## **PROGRAM-3**

### **AIM: Create and distribute a Torrent file to share a file in LAN Environment**

**Following are the steps to create a Torrent file:**

1. Launch your torrent client.
2. Go to File > Create New Torrent in the application.
3. Next, click on Add File to select a location from where we want to add the Target File.
4. Next, we need to add the Tracker URLs. These URLs are used to track & locate the file on the internet. By default, programs like uTorrent will add two or more URLs by default, in case it doesnot happens on its own, we need to add our own URLs. Most of the time these URLs need to be searched according to the website we will use.
5. Add any additional information related to the file like comments, RSS feeds etc. & choose theprivacy level, public or private.
6. Once done, click on Create to create the Torrent File & choose a location to save it.
7. Later on this file can be shared with others or uploaded to a Torrent network.

## PROGRAM-4

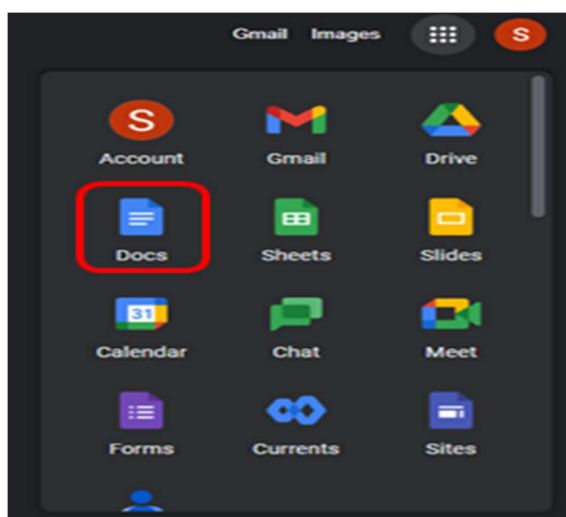
**AIM: Use Google collaboration tools: Create Google Docs, Sheets and Slides and share it with other users.**

**Google Collaboration tools:** Google collaboration tools is a robust set of applications that can help businesses boost productivity and collaboration.

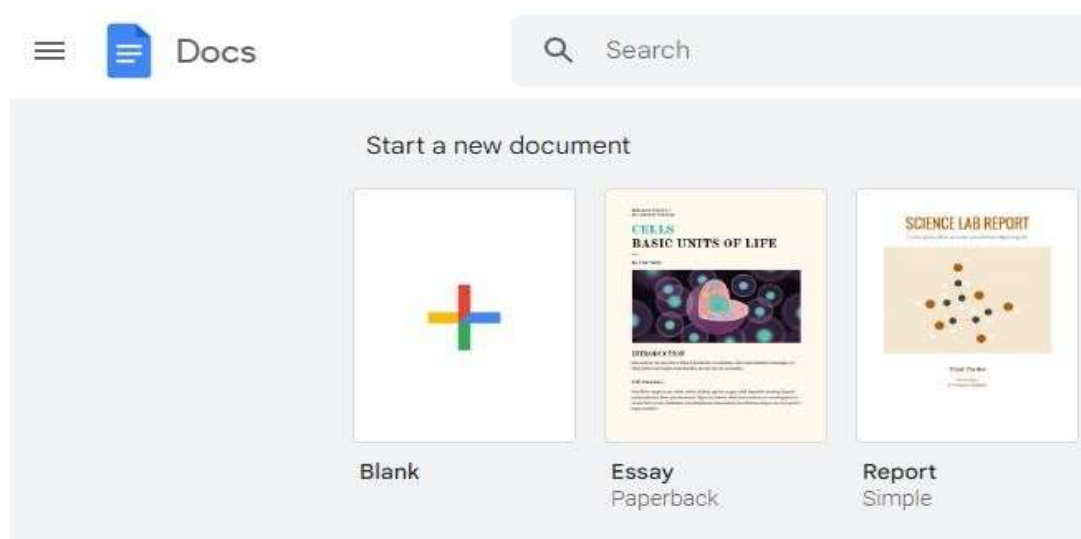
### Create Google Docs:

**Step 1:** Sign into your Google account.

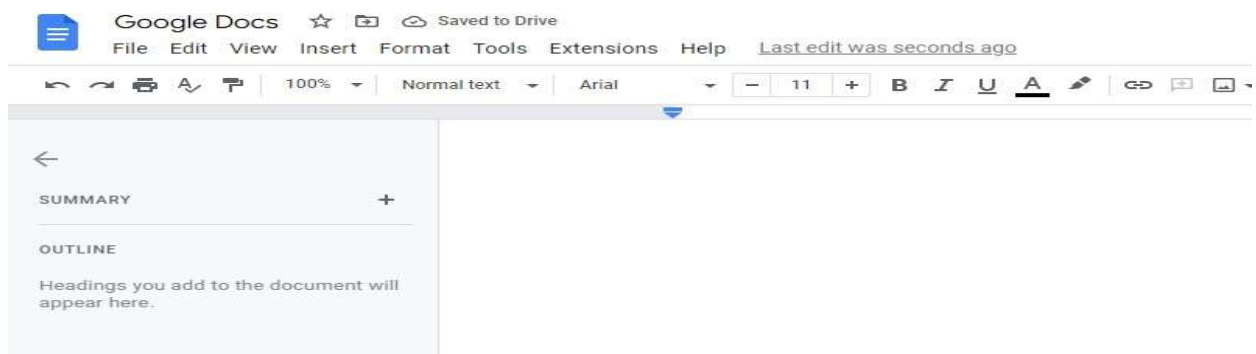
**Step 2:** Click Top right corner and click on the “Docs” icon.



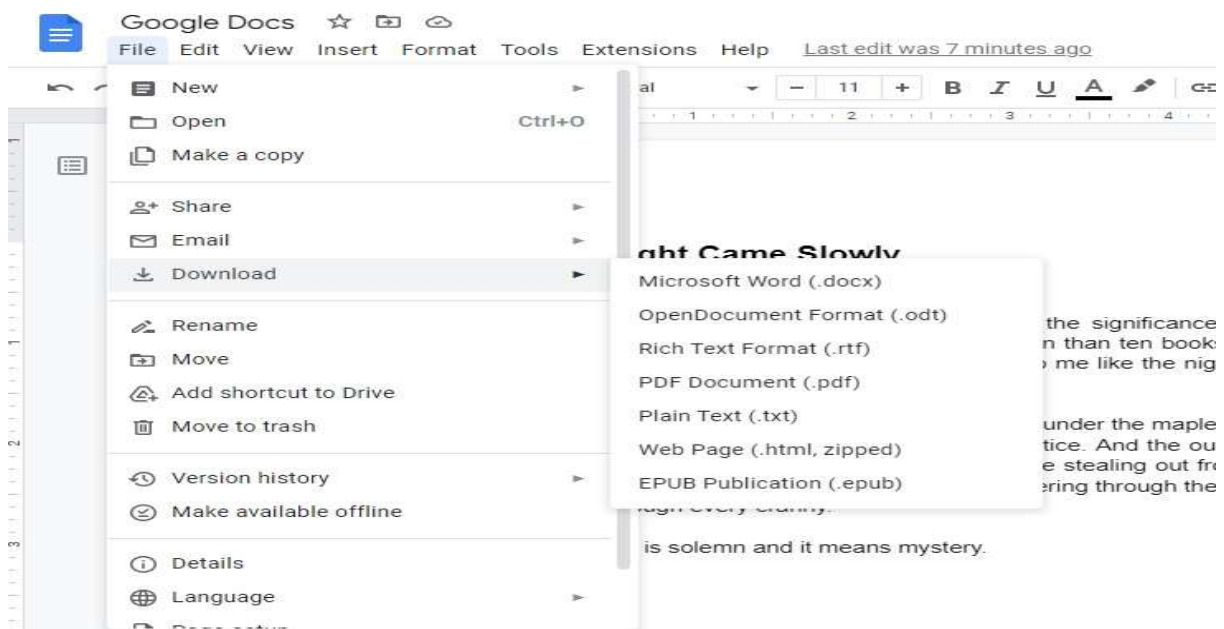
**Step 3:** Click on plus button for creating the new docs.



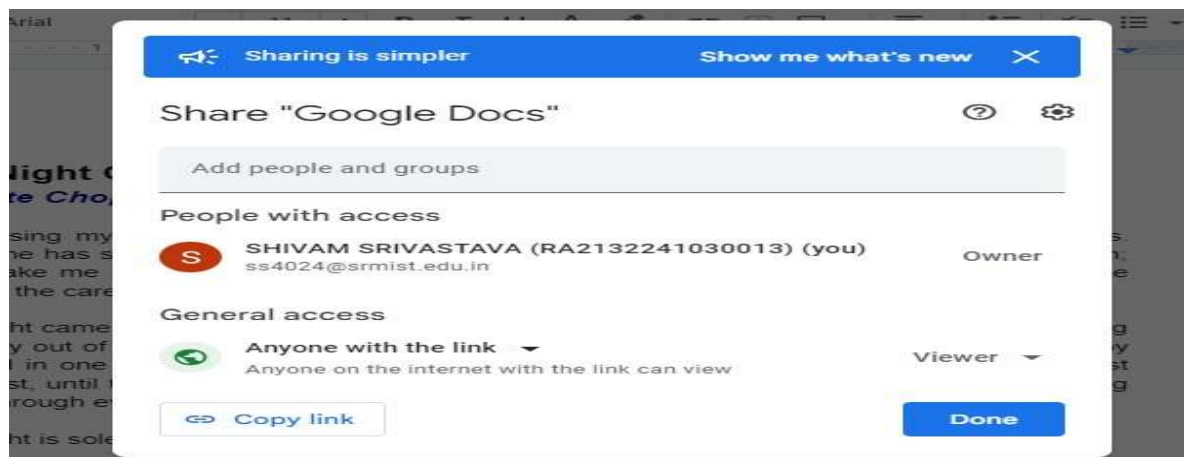
**Step 4:** Rename your google docs by clicking the top left corner



**Step 5:** You can download the file in your local computer.



**Step 6:** You can share this google doc with others by clicking the “Share” button placed in the top left corner.



## PROGRAM-5

**AIM: Explore public cloud services like Amazon, Google, Salesforce, Digital Ocean etc.**

### **Microsoft Azure:**

Azure is Microsoft's Cloud platform. It is a platform through which we can use Microsoft's resources. For example, to set up a huge server, we will require huge investment, effort, physical space and so on. In such situations, Microsoft Azure comes to our rescue. It will provide us with virtual machines, fast processing of data, analytical and monitoring tools and so on to make our work simpler. Popularly termed as "Pay as You Go", which means how much you use, pay only for that.

#### **Azure can help in the following ways-**

8. **Lesser Capital:** We don't have to worry about the capital as Azure cuts out the high cost of hardware. You simply pay as you go and enjoy a subscription-based model that's kind to your cash flow. Also, setting up an Azure account is very easy. You simply register in Azure Portal and select your required subscription and get going.
9. **Less Operational Cost:** Azure has low operational cost because it runs on its own servers whose only job is to make the cloud functional and bug-free, it's usually a whole lot more reliable than your own, on-location server.
10. **Cost Effective:** If we set up a server on our own, we need to hire a tech support team to monitor them and make sure things are working fine. Also, there might be a situation where the tech support team is taking too much time to solve the issue incurred in the server. So, in this regard it is way too pocket friendly.
11. **Easy Back Up and Recovery options:** Azure keeps backups of all your valuable data. In disaster situations, you can recover all your data in a single click without your business getting affected. Cloud-based backup and recovery solutions save time, avoid large up-front investment, and roll up third-party expertise as part of the deal.
12. **Better Security:** Azure provides more security than local servers. Be carefree about your critical data and business applications. As it stays safe in the Azure Cloud. Even in natural disasters, where the resources can be harmed, Azure is a rescue. The cloud is always on.

### **Amazon Web Services (AWS):**

The Amazon Web Services (AWS) platform provides more than 200 fully featured services from data centers located all over the world. Amazon web service is an online platform that provides scalable and cost-effective cloud computing solutions.

AWS is a broadly adopted cloud platform that offers several on-demand operations like compute power, database storage, content delivery, etc., to help corporates scale and grow.

The leading cloud provider in the marketplace is Amazon Web Services. It provides over 170 AWS services to the developers so they can access them from anywhere at the time of need.

For example, Adobe creates and updates software without depending upon the IT teams.

It uses its services by offering multi-terabyte operating environments for its clients. By deploying its services with Amazon services, Adobe integrated and operated its software in a simple manner.

#### **Advantages of AWS:**

1. AWS provides a user-friendly programming model, architecture, database as well as operating system that has been already known to employers.
2. AWS is a very cost-effective service. There is no such thing as long-term commitments for anything you would like to purchase.
3. It offers billing and management for the centralized sector, hybrid computing, and fast installation or removal of your application in any location with few clicks.
4. There is no need to pay extra money on running data servers by AWS.
5. AWS offers a total ownership cost at very reasonable rates in comparison to other private cloud servers.

#### **Google Cloud Platform:**

Starting from 1998 with the launch of Google search, Google has developed one of the largest and most Powerful IT Infrastructure in the world. Today, this infrastructure is used by billions of users to use services such as Gmail, YouTube, Google Photo and Maps. In 2008, Google decided to open its network and IT infrastructure to business customers, taking an infrastructure that was initially developed for consumers application to public service and launching google cloud platform.

All the services listed above are provided by Google hence the name Google Cloud Platform (GCP).

#### **Why choose GCP?**

- ☐ GCP allows you to choose between computing, storage, big data, machine learning, and application services for your web, mobile, analytics, and back-end solutions.
- ☐ It's global and it is cost-effective.
- ☐ It's open-source friendly.
- ☐ It's designed for security.

#### **Advantages of GCP:**

1. **Good documentation:** We are talking about many pages in total, including a reasonably detailed API Reference guide.
2. Different storage classes for every necessity: Regional (frequent use), Nearline (infrequent use), and Coldline (long-term storage).
3. **High durability:** This suggests that data survives even within the event of the simultaneous loss of two disks.
4. Many regions available to store your data: North America, South America, Europe, Asia, and Australia.
5. The "Console" tab within the documentation allows you to try for free of charge different SDKs. It's incredibly useful for developer

## PROGRAM-6

**AIM: Quizzes on different service models and deployment models. Report submission - Comparison of various services provided by different Cloud Service Providers (configuration of VM, cost, Networkbandwidth etc.).**

For the question, the top three Cloud Providers have been compared: AWS, Azure & GCP

### **AWS:**

- Amazon is an IaaS market leader, holding 31% of the cloud market share.
- AWS has over 175 cloud services for a broad range of use cases and industries. The top Amazon most used services are: Amazon EC2 (compute capacity), Amazon RDS (relational database), Amazon S3 (cloud storage), Amazon CloudFront (content delivery service) and Amazon Glacier (web storage service). EC2 allows Amazon customers to use virtual computer clusters that are available all the time.
- Currently, AWS serves 245 countries and spans 25 geographic regions: 7 in North America, 9 in Asia-Pacific, 6 in Europe, 1 in South America, 1 in the Middle East, and 1 in Africa. Every region is isolated and consists of several availability zones (AWS spans 80 availability zones in total).

### **Azure:**

- Microsoft Azure's market share among IaaS cloud providers is 20%.
- Microsoft Azure has over 600 services. Azure offers VMs as a part of its IaaS offering, Active Directory to synchronize on-premise directories, and enables single sign-on. The company also provides mobile engagement with real-time analytics and tracking of user behaviors and storage services, as well as data management tools such as Azure Data Explorer, Azure SQL Database, Serverless, CDN, Azure AI, Azure IoT and other services.
- Azure has 54 regions and is more available than any other cloud provider. Every Azure region has a minimum of three availability zones, enabling its customers to run two isolated copies of their applications.

### **GCP:**

- The market share of Google Cloud in infrastructure, as a service market, is 7% per Canalys.
- The Google Cloud platform offers 100 products that can be grouped into six categories: storage, databases, computing and hosting (servers, containers VMs), networking (VPC, load balancing, cloud DNS), big data (Big Query for data analysis, Dataflow for batch and streaming data processing), and machine learning (AI platform).
- Google Cloud is available in 200 countries. It spans 25 regions (9 in North and South America, 9 in Asia-Pacific, and 7 in Europe). Google Cloud has 76 zones and 144 network Edge locations.

## PROGRAM-7

**AIM: Create a simple web service using Python Flask/ Java/ any language: Client - Server Model should be implemented using socket/http**

The below Web Service is deployed as a Function App API on Microsoft Azure. The API is written in JavaScript which implements HTTP Calls & checks if a temperature entry collected from IoT sensor is within the prescribed range or not:

```
module.exports = function (context, req) {
    context.log('Drive Gear Temperature Service triggered');
    if (req.body && req.body.readings) {
        req.body.readings.forEach(function(reading) {

            if(reading.temperature<=25) {
                reading.status = 'OK';
            } else if (reading.temperature<=50)
                {reading.status = 'CAUTION';
            } else {
                reading.status = 'DANGER'
            }
            context.log('Reading is ' + reading.status);
        });

        context.res = {
            // status: 200, /* Defaults to 200 */
            body: {
                "readings": req.body.readings
            }
        };
    }
    else {
        context.res = {
            status: 400,
            body: "Please send an array of readings in the request body"
        };
    }
    context.done();
};
```

## Output:

Home > Microsoft.Web-FunctionApp-Portal-d144b4b7-b0d6 > escalator-funcs > HttpTrigger1

HttpTrigger1 | Code + Test ...

Function

Search (Ctrl+/)



Save



Discard



Refresh



Test/Run



Upload



Get function URL

Overview

Developer

Code + Test

Integration

Monitor

Function Keys

escalator-funcs \ HttpTrigger1 \ function.json

```
1 {
2   "bindings": [
3     {
4       "authLevel": "function",
5       "type": "httpTrigger",
6       "direction": "in",
7       "name": "req",
8       "methods": [
9         "get",
10        "post"
11      ]
12    },
13    {
14      "type": "http",
15      "direction": "out",
16      "name": "res"
17    }
18  ]
19 }
```

Filesystem Logs Log Level Stop Copy Clear Maximize

Connected!  
2021-10-21T19:31:43 Welcome, you are now connected to log-streaming service.  
The default timeout is 2 hours. Change the timeout with the App Setting  
SCH\_LOGSTREAM\_TIMEOUT (in seconds).

Input Output

Provide parameters to test the HTTP request. Results can be found in the Output tab.

HTTP method

POST

Key

master (Host key)

Query

Name Value

Enter a name

Enter a value

+ Add parameter

Headers

+ Add header

Body

```
1 {
2   "name": "Azure"
3 }
```

Run

Close



## PROGRAM-8

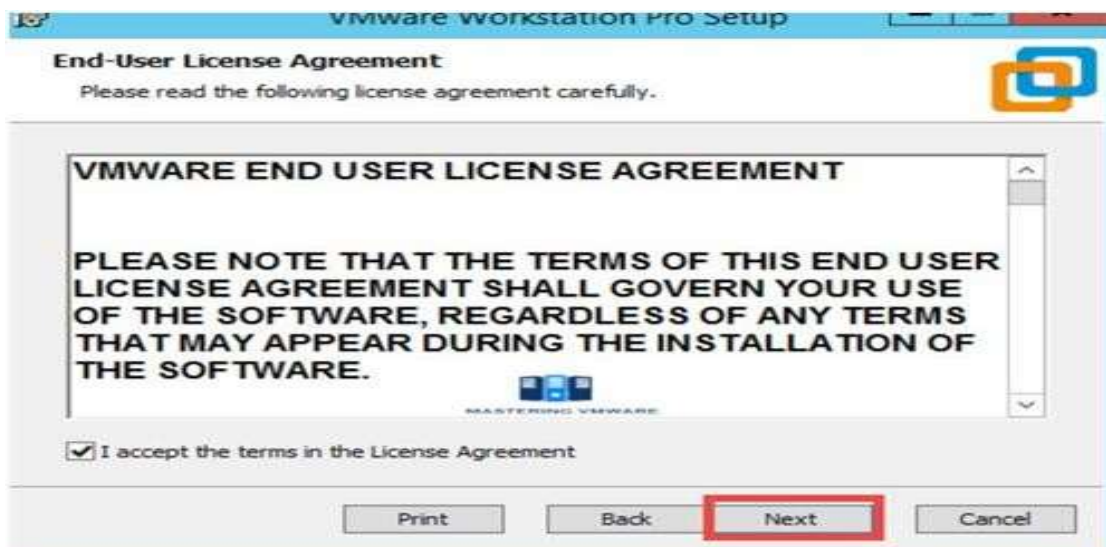
**AIM: Install Oracle Virtual Box / VMware Workstation & create a chat application.**

Steps for installing VMware workstation:

1. To install VMware workstation from the official website of VMware (<https://www.vmware.com>).
2. Download & setup and run it.
3. Click on VMware Software and Click on Next to the Installation wizard.



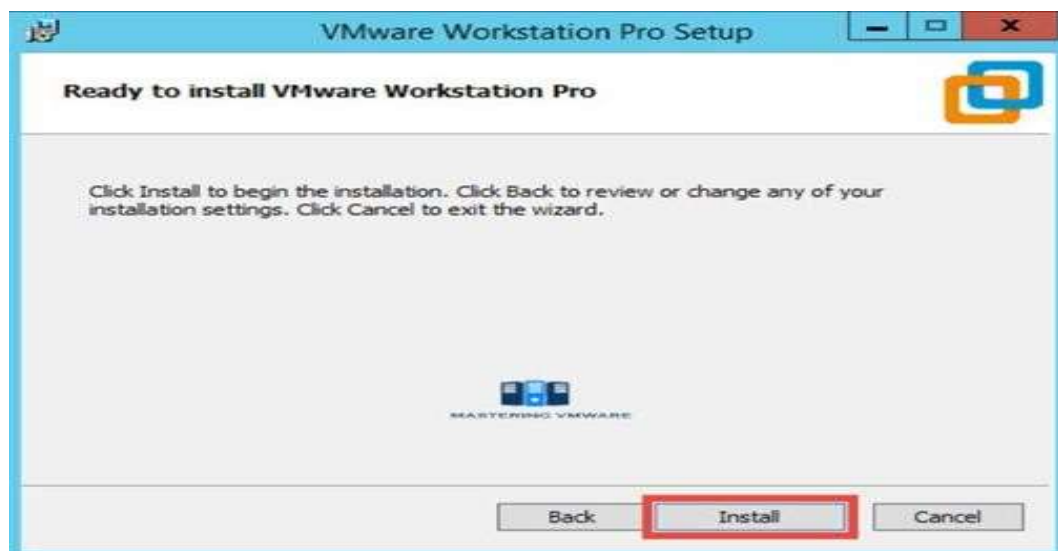
4. Read and accept the VMware End User license agreement & click next to continue.



5. Specify the Installation directory. You can also enable Enhance keyboard driver here. And then Click next to continue.



6. Click Install button to start the installation.



7. Installation will take just few seconds to complete.

If you have the license key then click on License to enter the license or you can also click Finish to exit the Installer. That's it we have successfully installed VMware Workstation.



### Create a chat application:

#### Server-side script:

**File name: server.py**

```
import socket
new_socket = socket.socket()
host_name = socket.gethostname()
s_ip = socket.gethostbyname(host_name)
port = 8080
new_socket.bind((host_name, port))
print("~~~ WELCOME TO THE CHAT APPLICATION ~~~")
name = input("Enter your name: ")
new_socket.listen(1)
conn, add = new_socket.accept()
print("Received connection from ", add[0])
client = (conn.recv(1024)).decode()
print(client + ' has connected...')
conn.send(name.encode())
while True:
    message = input('Me : ')
    conn.send(message.encode())
    message = conn.recv(1024)
    message = message.decode()
    print(client, ':', message)
```

## Output on computer 1:

## Client-side script:

**File name: client.py**

```
import socket
socket_server = socket.socket()
server_host = socket.gethostname()
ip = socket.gethostbyname(server_host)
sport = 8080
print("~~~ WELCOME TO THE CHAT APPLICATION ~~~")
server_host = '192.168.1.5'
name = input('Enter your name: ')
socket_server.connect((server_host, sport))
socket_server.send(name.encode())
server_name = socket_server.recv(1024)
server_name = server_name.decode()
print(server_name, ' has connected...') while
True:
    message = (socket_server.recv(1024)).decode()
    print(server_name, ":", message)

    message = input("Me : ")
    socket_server.send(message.encode())
```

### Output on computer 2:



A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Windows\py.exe' and standard window controls (close, maximize, minimize). The window content displays the output of a chat application. It starts with a welcome message, followed by a series of messages between 'Jenny' and 'Jack'. The messages include greetings, status updates, and plans for the next day. The text is displayed in a monospaced font on a dark background.

```
WELCOME TO THE CHAT APPLICATION
Enter your name: Jenny
Jack has connected...
Jack : Hi, Jenny!
Me : Hello, How r u?
Jack : ya, I'm fine. and u?
Me : I'm great.
Jack : nice
Me : what r u doing?
Jack : nothing, watching movie at home.
Me : ohh! nice..
Jack : hmm
Me : What are you doing tomorrow?
Jack : nothing.
Me : So tomorrow you and me go on coffee?
Jack : ya sure
Me : okk done.. 8'o clock tomorrow?
Jack : Okay then I'll pick you up from your home.
Me : okk.. done.
Jack : hmm
Me : ok... bye see you then
Jack : ok bye..
Me :
```

## PROGRAM-9

**AIM: Review web services implementation –Proper Connection should be established between the client and server to make use of the service offered by the Server.**

### **Review the working of application in virtual environment.**

In a project that I deployed, I hosted it on Microsoft Azure as a Flask App as a Web App in Azure App Service. While deploying the web app, the following points were taken care of to ensure that the application works as expected and proper connection is established between the client app or browser & the App Service instance:

- **Firewall:** The web app uses Firewall services provided by Azure to ensure that no unauthorized traffic flows through the web app instance. All outbound IPs are blocked, & for inbound communication, only the TCP Port is open.
- **Stack:** In terms of stack, everything has been verified to be performing optimally & as per the designated requirements.
- **Ports:** Port rules are enforced by the Firewall Implementation by Azure. The following rules are applied:
  - **Outbound traffic:** allowed only on HTTP Port 80 for API Calls & Port 22 for SSH.
  - **Inbound Traffic:** only allowed on HTTP Port 80 for serving Web content.
- **Connection:** For reliable & concrete connection, TCP Protocol is utilized. TCP ensures reliable transmission of data back & forth between the web server & multiple clients.
- **Certificate:** The use of SSL Certificate here is made to facilitate secure transmission. The certificate verifies the authenticity of the web app & the underlying platform.
- **Domain:** The application runs on default domain names provided by Azure & is formed as a genuine domain name in order to engrave trust in the user.
- **Backend API Calls:** The backend API Calls are passed using HTTP Requests, allowing for smooth & secure transmission of data between the API Server & application.
- **Virtual Environment Testing:** The application with all the parameters was tested on localhost & then in a staging environment on Azure, before deploying it to production.

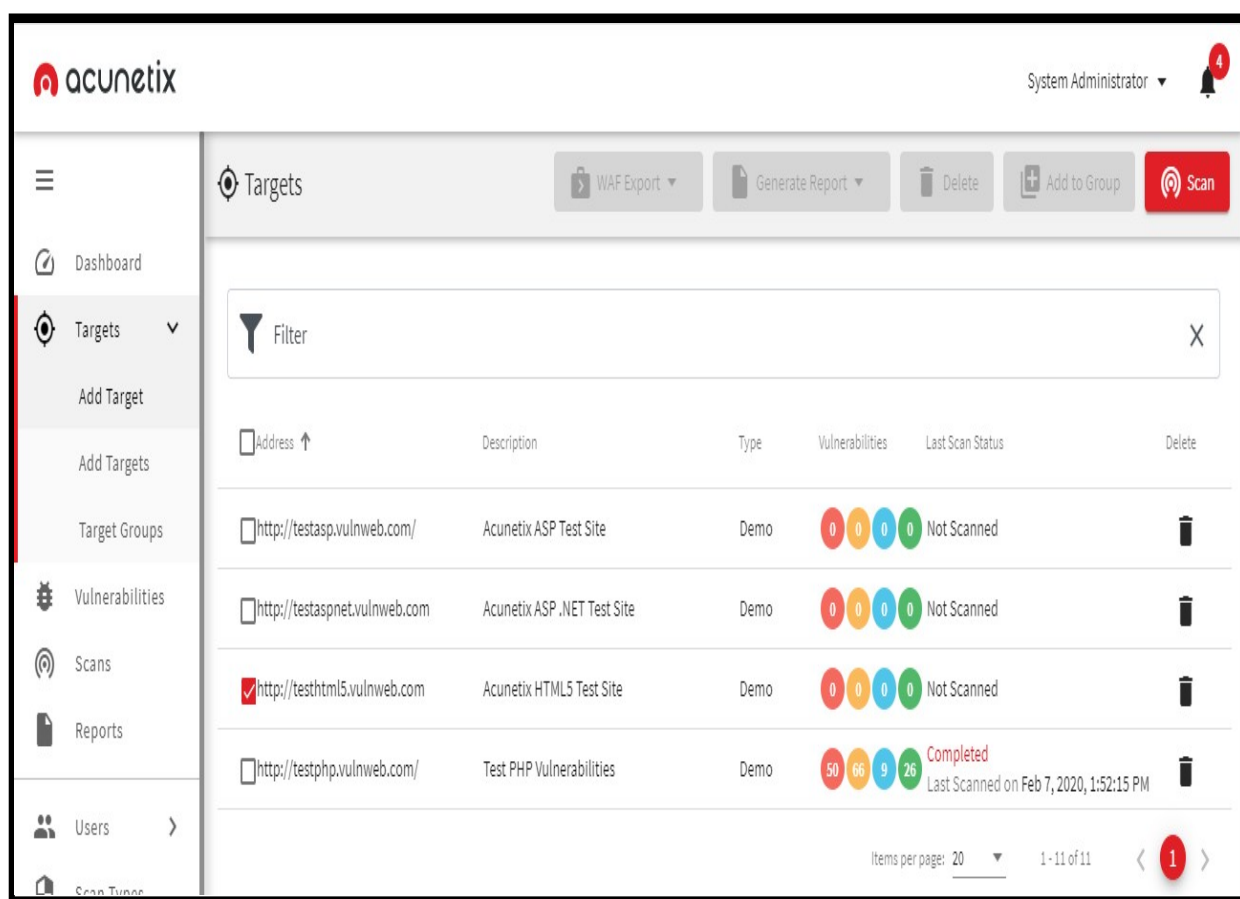
## PROGRAM-10

### AIM: Use Security tools like ACUNETIX, ETTERCAP to scan Web Applications on the Cloud.

The web server logs will show your IP address and all the attacks made by Acunetix. If you are not the sole administrator of the website or web application, please make sure to warn other administrators before performing a scan. Some scans might cause a website to crash, requiring a restart of the website.

After configuring your Targets, you are ready to launch Scans and start identifying any vulnerabilities that exist in the web applications. There are multiple ways to start a Scan, which include:

- From the Targets list, select the Targets to scan, and click the Scan button



- From within the Scanning Options dialog, configure the options to be used for the scan, then click the "Create Scan" button.

- **Scan Type** - Choose between Full Scan or a scanning profile which will scan for specific vulnerabilities, such as High Risk Vulnerabilities only.
  - **Report** - You can request that a report is automatically generated after the scan is completed.
  - **Schedule** - Select if the scan should start instantly, or if the scan should be scheduled for a future date / time. You can also configure recurrent scans.
- Click the "Create Scan" button to launch the scan.

### Choose Scanning Options

Scan Type

Full Scan

Report

None

Schedule

Instant

Cancel

Create Scan



## PROGRAM-11

**AIM: Cloud Networks for finding vulnerabilities, verifying leakage of information to an unauthorized third party.**

### - ACUNETIX

This information gathering tool scans web applications in the cloud and lists possible vulnerabilities that might be present in the given web application. Most of the scanning is focused on finding SQL injection and cross site scripting vulnerabilities. After scanning, it generates a detailed report describing vulnerabilities along with the suitable action that can be taken to remedy the loophole.

This tool can be used for scanning cloud applications. Beware: there is always a chance of false positives. Any security flaw, if discovered through scanning, should be verified. The latest version of this software, Acunetix WVS version 8, has a report template for checking compliance with ISO 27001, and can also scan for HTTP DDOS Attacks.

### - CAIN & ABEL

This is a password recovery tool. Cain is used by penetration testers for recovering passwords by sniffing networks, brute forcing and decrypting passwords. This also allows pen testers to intercept VoIP conversations that might be occurring through clouds. This multi functionality tool can decode Wi-Fi network keys, unscramble passwords, discover cached passwords, etc. The feature that separates Cain from similar tools is that it identifies security flaws in protocol standards rather than exploiting software vulnerabilities. This tool is very helpful for recovering lost passwords.

In the latest version of Cain, the 'sniffer' feature allows for analyzing encrypted protocols such as SSH-1 and HTTPS. This tool can be utilized for ARP cache poisoning, enabling sniffing of switched LAN devices, thereby performing Man in the Middle (MITM) attacks. Further functionalities have been added in the latest version, including authentication monitors for routing protocols, brute-force for most of the popular algorithms and cryptanalysis attacks.

### - ETTERCAP

Ettercap is a free and open-source tool for network security, designed for analyzing computer network protocols and detecting MITM attacks. This tool can be used for pen testing cloud networks and verifying leakage of information to an unauthorized third party. It has four methods of functionality:

- **IP-based Scanning** – Network security is scanned by filtering IP based packets.
- **Mac-based Scanning** – Here packets are filtered based on MAC addresses. This is used for sniffing connections through channels.
- **ARP-based functionality** – ARP poisoning is used for sniffing into switched LAN through an MITM attack operating between two hosts (full duplex)

## PROGRAM-12

### AIM: Install and configure OpenStack all-in-one using Dev Stack / Pack stack.

OpenStack is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service, whereby virtual servers and other resources are made available to customers.

Packstack is mostly suitable for Red Hat Distribution Linux like CentOS and Fedora. It basically uses puppet modules to deploy various part of Openstack Components through ssh.

**Hardware:** Minimum 16GB of RAM, Processor with hardware virtualization extension and at least 1 network adapter.

We are Using Oracle VM Box for Host Operating System.

**Step 1:** Before starting installation process, you have update and upgrade your system

**Run Command:**

- ☐ sudo apt-get update
- ☐ sudo apt-get upgrade

**Step 2:** Create a new user and give permission to start the openstack installation

**Run Command:**

- ☐ sudo adduser stack
- ☐ sudo-I
- ☐ cho "stack ALL-(ALL) NOPASSWD: ALL" > >/etc/sudoers

**Step 3:** Download the deystack from github.com

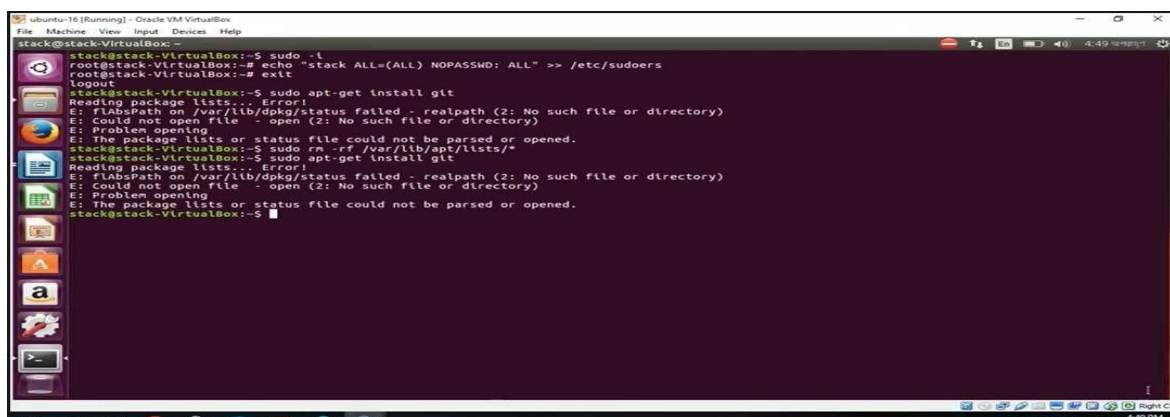
**Run Command:**

- ☐ sudo apt-get install git
- ☐ git clone <https://git.openstack.org/openstack-dev/devstack>

**Step 4:** Run the following commands to avoid errors before installation for lock error

**Run Command:**

- ☐ sudo rm /var/lib/dpkg/lock
- ☐ sudo rm /var/lib/apt/lists/lock
- ☐ sudo rm /var/cache/apt/archives/lock
- ☐ sudo rm -rf /var/lib/apt/list/\*



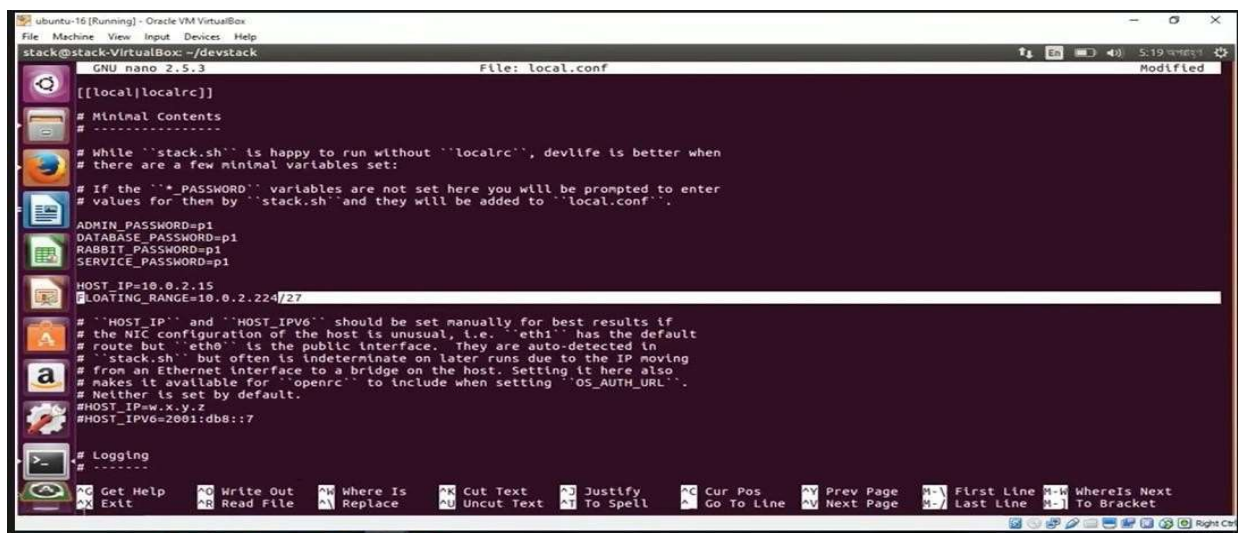
### Step 5: Configure local.conf fileRun Command:

- ☐ cd devstack/
- ☐ cd samples
- ☐ cp local.conf ../
- ☐ cd..
- ☐ Sudo nano local.conf

(We can set any password as of our choice) ADMIN\_PASSWORD=pass1  
DATABASE\_PASSWORD=pass1 RABBIT\_PASSWORD=pass1 SERVICE\_PASSWORD=pass1  
(Host IP is the IP of our system)

(We enter these values to configure local.conf file) HOST\_IP=10.0.2.15  
FLOATING\_RANGE=10.0.2.222/27

To save the file use command: **CTRL+X** and press **Y** for Confirmation.



```
[[local|localrc]]
# Minimal Contents
# -----
# While 'stack.sh' is happy to run without 'localrc', devlife is better when
# there are a few minimal variables set:
# If the 'ADMIN_PASSWORD' variables are not set here you will be prompted to enter
# values for them by 'stack.sh' and they will be added to 'local.conf'.
ADMIN_PASSWORD=p1
DATABASE_PASSWORD=p1
RABBIT_PASSWORD=p1
SERVICE_PASSWORD=p1
HOST_IP=10.0.2.15
FLOATING_RANGE=10.0.2.224/27
# 'HOST_IP' and 'HOST_IPV6' should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. 'eth1' has the default
# route but 'eth0' is the public interface. They are auto-detected in
# 'stack.sh' but often is indeterminate on later runs due to the IP moving
# from an Ethernet interface to a bridge on the host. Setting it here also
# makes it available for 'openrc' to include when setting 'OS_AUTH_URL'.
# Neither is set by default.
#HOST_IP=x.x.x.x
#HOST_IPV6=2001:db8::7
# Logging
# -----
Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  Prev Page  First Line  WhereIs Next
Exit      Read File  Replace  Uncut Text  To Spell  Go To Line  Next Page  Last Line  To Bracket
```

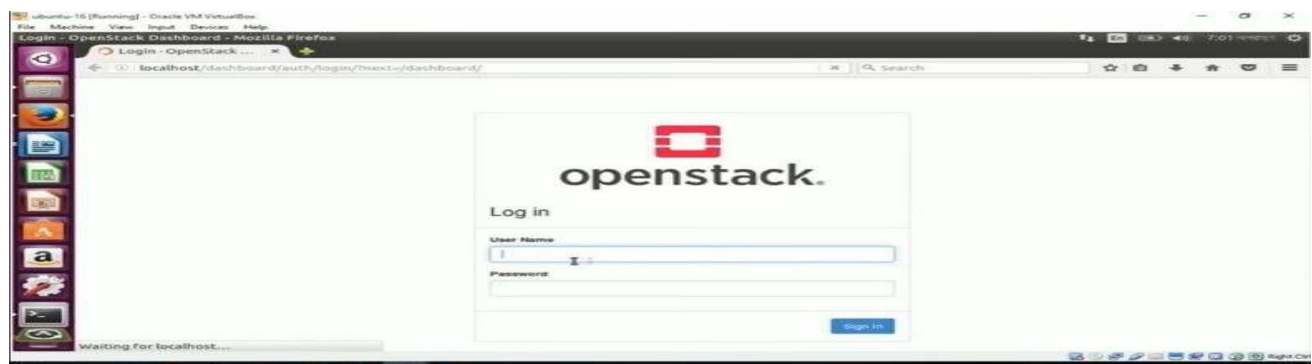
### Step 6: Now we can start Installation Process

#### Run Command:

- ☐ ./stack.sh

This Installation will take some time depending on internet speed.

After completion of Installation, we can check if the openstack is installed correctly by typing **localhost** in web Brouser, if Openstack login page Appears it means it is successfully installed.



## PROGRAM-13

### AIM: Launch VMs in OpenStack through Dashboard.

**Step 1:** Launch Openstack from browser & Login.

**Step 2:** Now, Go to Project -> Compute -> Instances.

Launch Instance ✕

Details

Source \*

Flavor \*

Networks \*

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name \*

vm01

Availability Zone

nova

Count \*

1

Total Instances (41 Max)

17%

6 Current Usage

1 Added

34 Remaining

✕ Cancel

< Back

Next >

Launch Instance

**Step 3:** Select Instance Boot Source (eg. "Image"), and choose desired image (eg. "Ubuntu 16.04 LTS") by clicking on arrow. If you do not need to have the system disk bigger than the size defined in a chosen flavor, we recommend setting "Create New Volume" feature to "No" state.

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume. ?

Select Boot Source

Image

Create New Volume

Yes

No

Allocated

Name	Updated	Size	Type	Visibility
Select an item from Available items below				

▼ Available 16

Select one

Q

Click here for filters.

✕

Name	Updated	Size	Type	Visibility	
> Ubuntu 20.04 LTS	10/20/20 9:37 AM	2.94 GB	raw	Public	↑
> Ubuntu 18.04 LTS	10/16/20 6:21 AM	3.17 GB	raw	Public	↑
> Ubuntu 16.04 LTS	8/7/20 3:46 PM	2.69 GB	raw	Public	↑
> Ubuntu 18.04 + QGIS	8/7/20 12:36 AM	15.20 GB	raw	Public	↑

**Step 4: Choose Flavor (eg. eo1.xsmall).**

Flavors manage the sizing for the compute, memory and storage capacity of the instance.



Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
> eo1.xsmall	1	1 GB	8 GB	8 GB	0 GB	Yes	↓

▼ Available 22

Select one

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
> ds.large.nvme	40	125 GB	64 GB	64 GB	0 GB	Yes	↑
> eo1.small	2	2 GB	16 GB	16 GB	0 GB	Yes	↑
> eo1.xmedium	1	2 GB	8 GB	8 GB	0 GB	Yes	↑
> eo1.medium	2	4 GB	16 GB	16 GB	0 GB	Yes	↑
> eo1.large	4	8 GB	32 GB	32 GB	0 GB	Yes	↑

**Step 5: Choose Flavor (eg. eo1.xsmall).**

Networks provide the communication channels for instances in the cloud.



▼ Allocated 2

Select networks from those listed below.

	Network	Subnets Associated	Shared	Admin State	Status	
1	> private_network_09064	private_subnet_09064	No	Up	Active	↓
2	> eodata	eodata	Yes	Up	Active	↓

▼ Available 0

Select at least one network

Network	Subnets Associated	Shared	Admin State	Status
No available items				

**Step 6:** Open "Security Groups" After that, choose "allow\_ping\_ssh\_rdp" and "default"

Select the security groups to launch the instance in. ?

▼ Allocated **2**

Name	Description	
> default	Default security group	↓
> allow_ping_ssh_rdp		↓

▼ Available **0** Select one or more

×

Name	Description
No available items	

**Step 7:** Choose or generate SSH keypair for your VM. Next, launch your instance by clicking on blue button.

A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair. ?

Allocated

Displaying 1 item

Name	Fingerprint	
> ubuntu	7d:72:67:74:f9:ab:08:68:26:cd:73:e6:0c:8e:9b:ba	↓

Displaying 1 item

▼ Available **2** Select one

×

Displaying 2 items

Name	Fingerprint	
> geolive	c7:9c:12:05:9f:af:3d:40:73:18:9d:11:c8:02:96:47	↑
> rh	2e:6d:48:46:d9:22:3a:b1:a4:2b:88:b0:06:41:f1:e7	↑

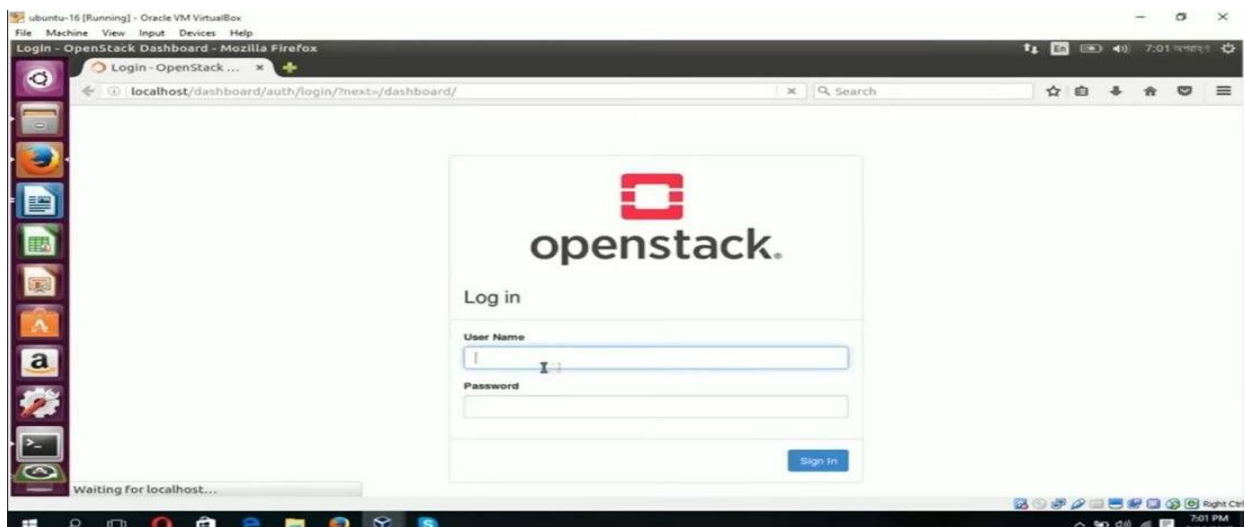
Displaying 2 items

## PROGRAM-14

**AIM: OpenStack Dashboard should be accessed through Web Browser. Verify the working of the instance by logging or pinging the instance.**

After Successful Installation Openstack Dashboard is accessible through Web browser.

We can access it by entering “localhost” into browser and Login with our saved Username & Password.

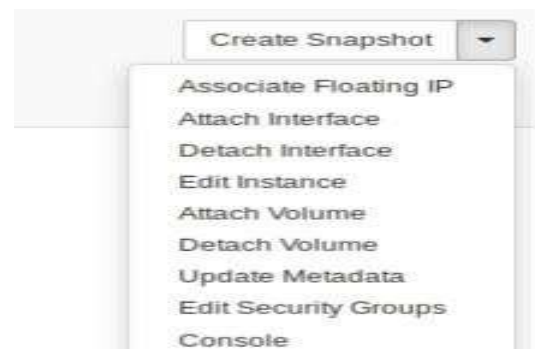


**To verify the working of Instance :**

**Step 1:** we can see "Instances" menu with our newly created VM.

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State
<input type="checkbox"/>	vm01	Ubuntu 16.04 LTS	10.111.0.142 private_network_09064 192.168.0.8	eo1.xsmall	ubuntu	Active	nova	None	Running

**Step 2:** Open the drop-down menu and choose "Console".



After opening Terminal Run Command: eoconsole

**Step 3:** Now we Need to Configure Password.

```
Ubuntu 16.04.6 LTS vm01 tty1

vm01 login: eoconsole
You are required to change your password immediately (root enforced)
Enter new UNIX password:
Retype new UNIX password:
```

After Re-entering password, we can use commands, The instance is created Successfully.

```
Ubuntu 16.04.6 LTS vm01 tty1

vm01 login: eoconsole
You are required to change your password immediately (root enforced)
Enter new UNIX password:
Retype new UNIX password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-165-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

eoconsole@vm01:~$ _
```

After finishing we will enter “exit” command into terminal to close VM