Procedure for Installing Oracle VirtualBox, Downloading Windows 10 ISO, and Creating a Virtual Machine

---

Step 1: Download and Install Oracle VirtualBox

1. Open your web browser and go to the official Oracle VirtualBox website: https://www.virtualbox.org/
2. Click on the **"Downloads"** link on the homepage.
3. Under the section "VirtualBox binaries," click on the link for your operating system (e.g., **Windows hosts** if you are using Windows).
4. The installer file will begin downloading. Wait for the download to complete.
5. Once downloaded, locate the installer file (usually in the Downloads folder) and double-click it to run.
6. The VirtualBox Setup Wizard will open. Click **Next** to proceed.
7. Choose the installation location or leave it as default and click **Next**.
8. Select the components to install (leave the defaults checked) and click **Next**.
9. Choose whether to create shortcuts, then click **Next**.
10. A warning about network interfaces may appear, click **Yes** to continue.
11. Click **Install** to begin the installation.
12. Once the installation completes, click **Finish** to exit the wizard. Optionally, launch VirtualBox immediately.

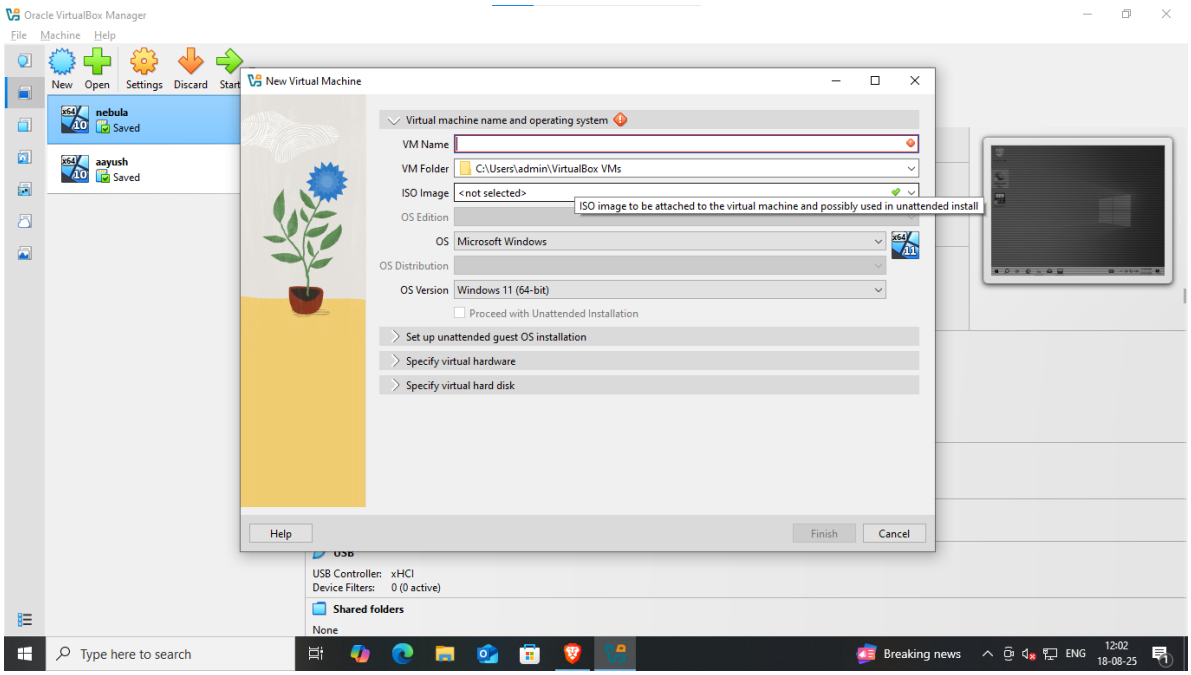
Step 2: Download Microsoft Media Creation Tool for Windows 10

1. Open your web browser and navigate to the official Microsoft Windows 10 download page: https://www.microsoft.com/software-download/windows10
2. Scroll down to the section **"Create Windows 10 installation media"** and click **Download tool now**.
3. Once downloaded, locate the file (MediaCreationTool.exe) and double-click to run it.
4. When the tool opens, accept the license terms.
5. Select **Create installation media (USB flash drive, DVD, or ISO file) for another PC** and click **Next**.
6. Choose the language, edition, and architecture (64-bit or 32-bit) according to your preference or leave it default, then click **Next**.
7. Select **ISO file** as the media to create and click **Next**.

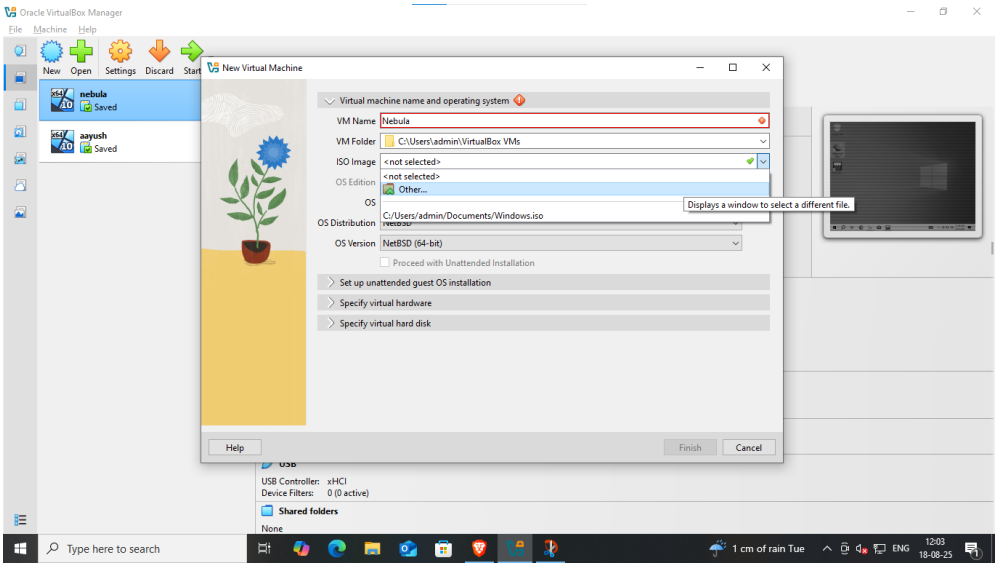
8. Choose a location to save the Windows 10 ISO file and click **Save**.
9. Wait while the tool downloads the Windows 10 ISO file.
10. Once completed, click **Finish** to close the tool.

Step 3: Create a Virtual Machine in Oracle VirtualBox Using the Windows 10 ISO

1. Open **Oracle VirtualBox** if it is not already running.
2. Click the **New** button on the top left of the VirtualBox Manager window.

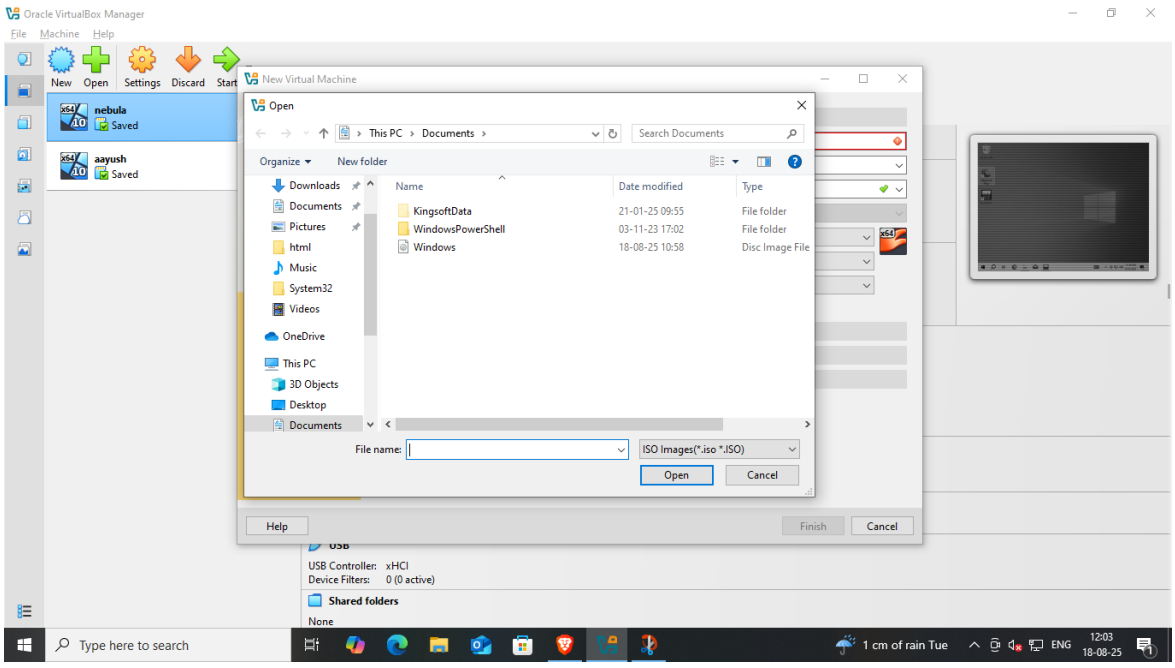


3. In the **Name and Operating System** window:
    - Enter a name for your VM (e.g., "Windows 10").
    - Select **Microsoft Windows** as the type.
    - Select **Windows 10 (64-bit)** or **Windows 10 (32-bit)** depending on the ISO you downloaded.
    - Click **Next**.

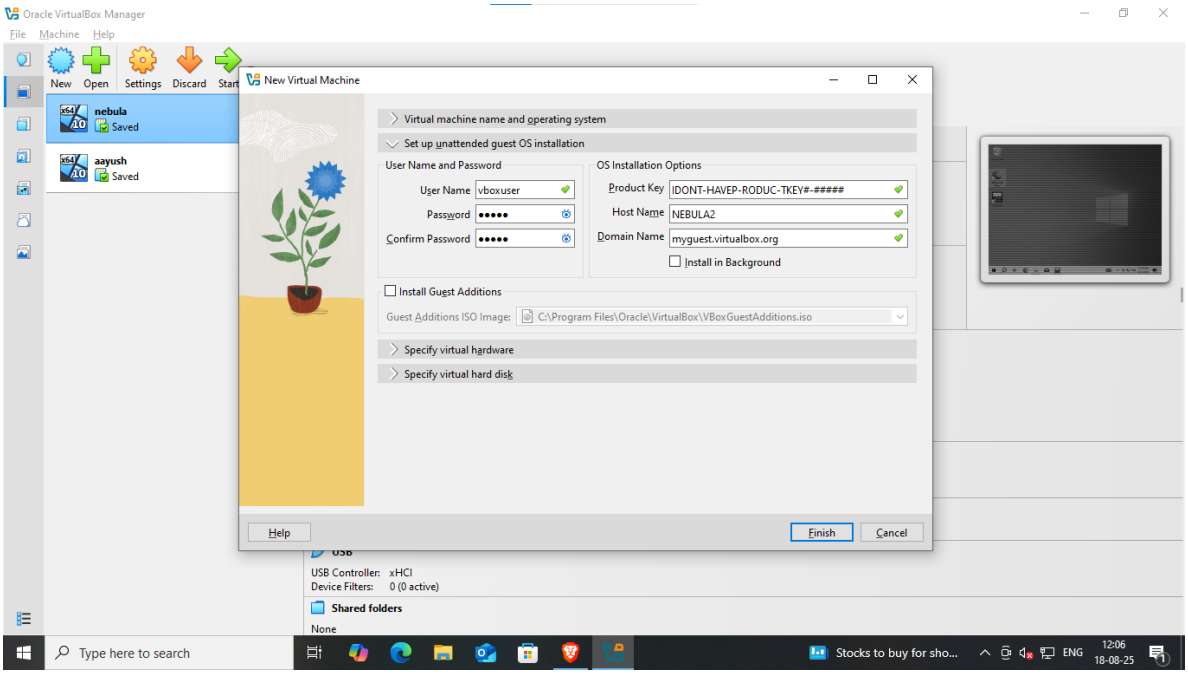


4. Navigate to ISO image and select the Windows 10 ISO file you downloaded.

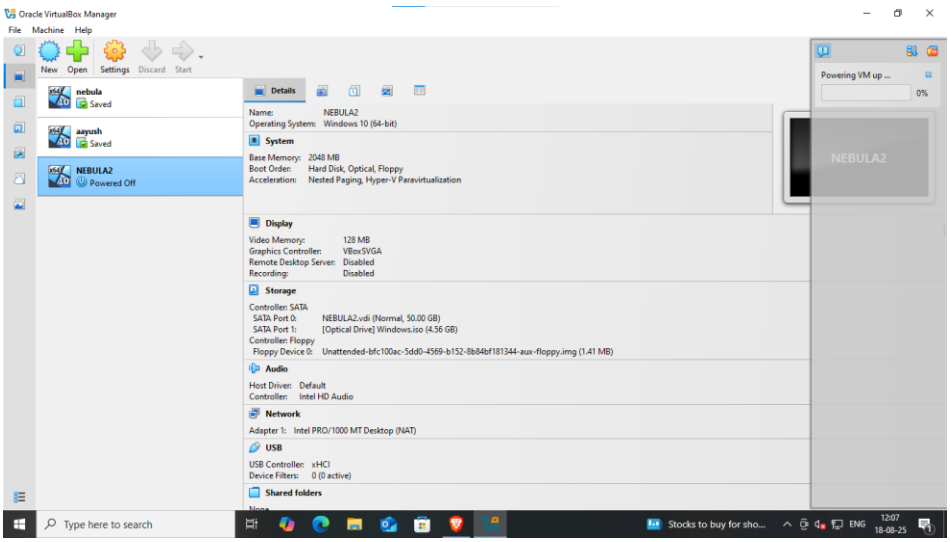
   Click **OPEN**



5. Now click on set up unattempted guest OS installation.And in the place of product key enter "IDONTHAVEPRODUCTKEY" and set password . click on finish.



6. Set the **Memory size (RAM)** for the VM. It is recommended to allocate at least **4096 MB (4 GB)** for Windows 10.
    - Adjust the slider or type in the value, then click **Next**.
7. Set the size of the virtual hard disk. It is recommended to allocate at least **50 GB**.
    - Choose the location or leave default and click **Create**.
8. Start the VM by selecting it and clicking **Start**.



9. The VM will boot from the Windows 10 ISO, and you can proceed with installing Windows 10 inside the virtual machine by following the on-screen instructions.

End of Procedure

Experiment -2

Aim : to install a c compiler in the virtual machine created using virtual box and execute a simple program

Procedure :

STEP 1:
create a virtual machine (refer exp 1)

STEP 2:

Step 2.1: Download Dev-C++

1. Open your VM's web browser.
2. Go to the official Dev-C++ download page:
   https://sourceforge.net/projects/orwelldevcpp/
3. Click **Download** to get the latest installer (`Dev-Cpp.exe`).

Step 2.2: Install Dev-C++

1. Run the downloaded installer inside the VM.
2. Follow the setup wizard:
   - Choose language.
   - Accept license.
   - Choose installation folder (default is fine).
3. Complete the installation.

Step 2.3: Create and run a C/C++ program

1. Launch **Dev-C++** from the Start menu.
2. Click **File > New > Source File**.
3. Type this sample program:

4. Click **Execute > Compile & Run** (or press **F9**).
5. A console window will pop up showing:

Cloud Computing Lab

Experiment-3

Create an Amazon EC2 instance and set up a web-server on the instance and associate an IP address with the instance.

Amazon Web Services (AWS) provides a powerful cloud computing platform with a wide range of services. One of the core services offered by AWS is Amazon Elastic Compute Cloud (EC2), which allows users to launch virtual servers in the cloud.

An Amazon EC2 instance is a virtual server in the cloud that allows users to run applications and workloads on AWS. It's essentially renting a computer in a data center, providing on-demand computing resources that can be scaled up or down as needed. EC2 instances offer various configurations of CPU, memory, storage, and networking, allowing users to choose the best fit for their specific requirements.

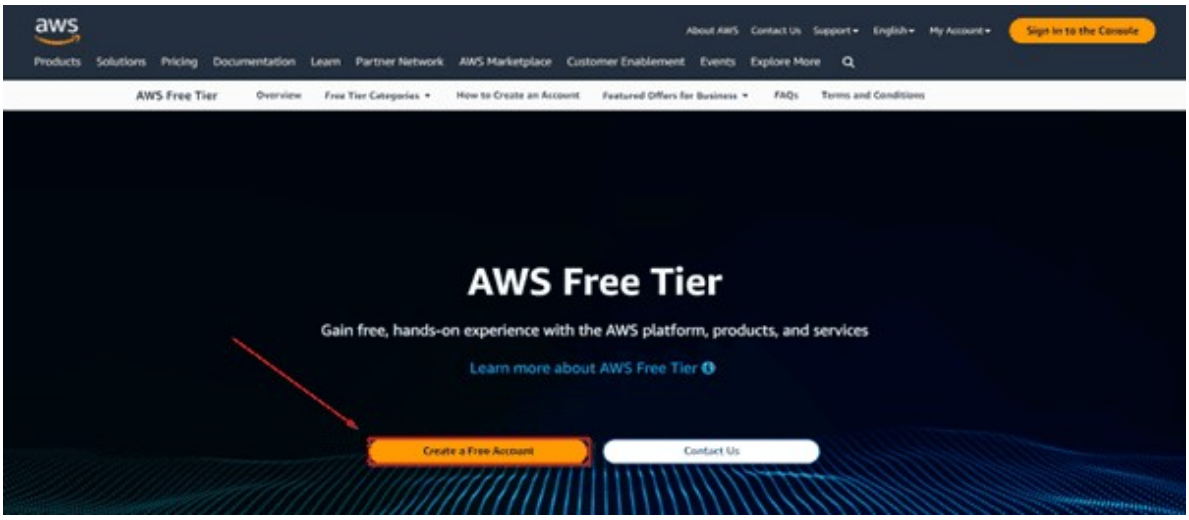
AWS Free Tier Limits

- Monthly **750 hours of Amazon EC2** Cloud computing capability that is scalable.
- **5 GB of basic storage on Amazon S3,** Infrastructure for scalable, robust, and secure object storage.
- Monthly database usage allotted to **Amazon RDS 750 hours** (for relevant database engines) SQL Server, MariaDB, PostgreSQL, and MySQL managed relational database services.
- **5 GB of Amazon EFS storage.** A shared file storage solution that is easy to use and scales for Amazon EC2 instances.
- **30 GB** of General Purpose (SSD) or Magnetic Elastic Block Storage from **Amazon Elastic Store** are long-lasting, dependable, low-latency block-level storage volumes for EC2 instances.

REGISTER FOR AWS FREE-TIER ACCOUNT

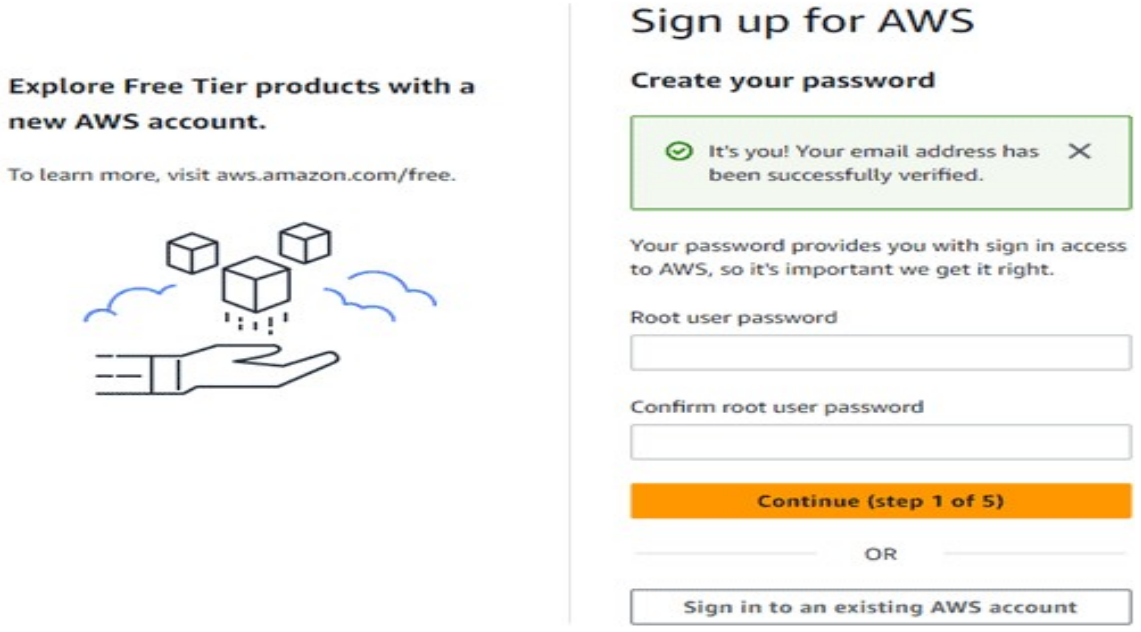
Step 1: First, open your web browser and navigate to the AWS Free Tier Page

Step 2: On the middle click of Create a Free Account



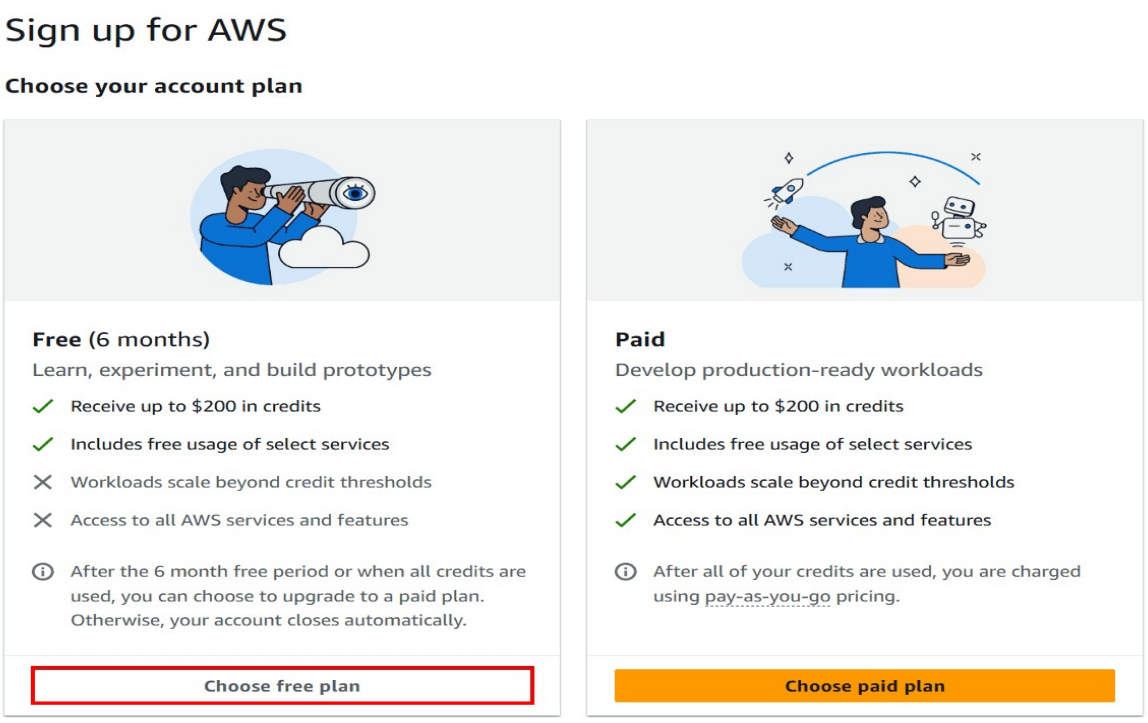
Step 3: Verify your email address.

- **Provide password: Provide the details that you want to use to log in to your AWS account, and click on Continue**



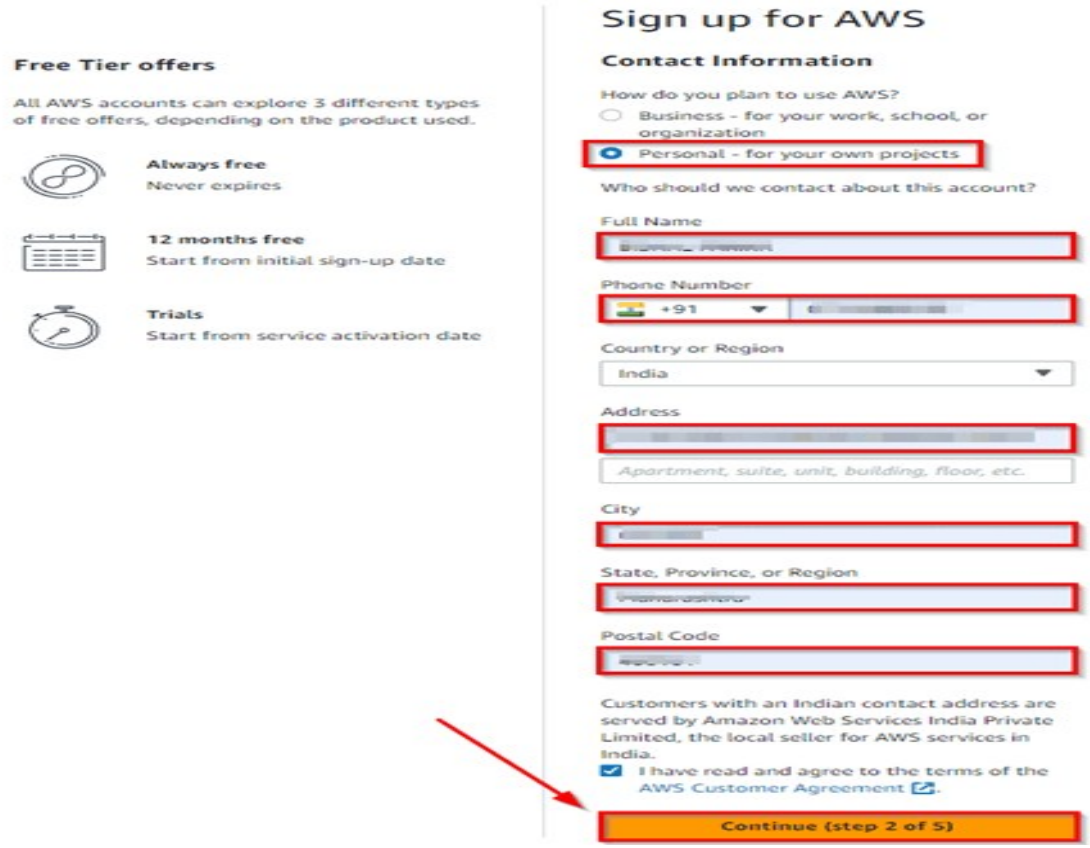
- **Email address: Enter the email ID that hasn't been registered yet with Amazon AWS**
- **Password: Type your Password**
- **Confirm password: Confirm the Password**
- **Captcha: Enter the given security check**

Choose your Option, we will be going with a 6-month Free.



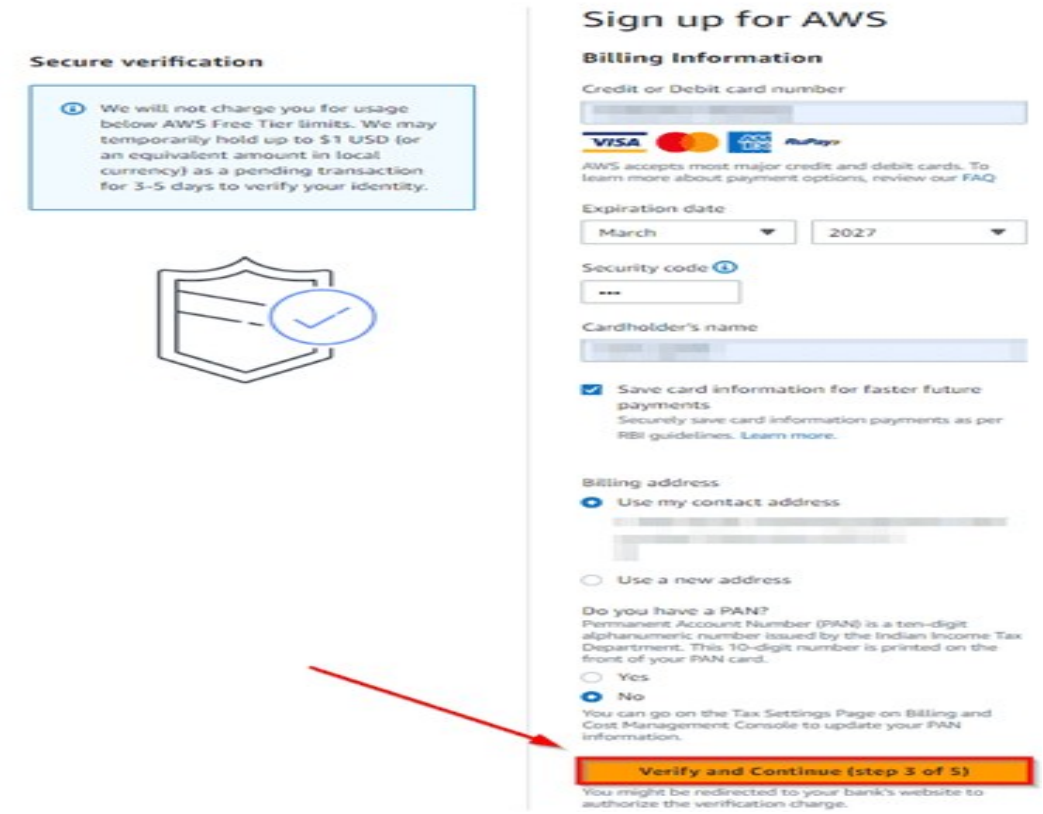
Step 4: Contact Information

Select your AWS type (Professional/ Personal). Fill in the correct information to validate your account if you're going to create a personal use account, then click on "Personal Account" otherwise, use "Company Account", accept the Terms and Conditions, and then click on Create Account and Continue

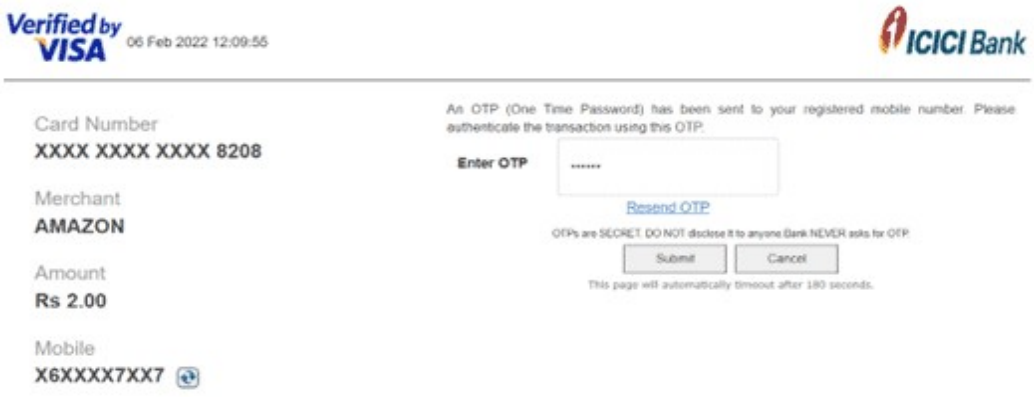


Note: Make sure to provide proper contact details and mobile number to get the Verification code from AWS.

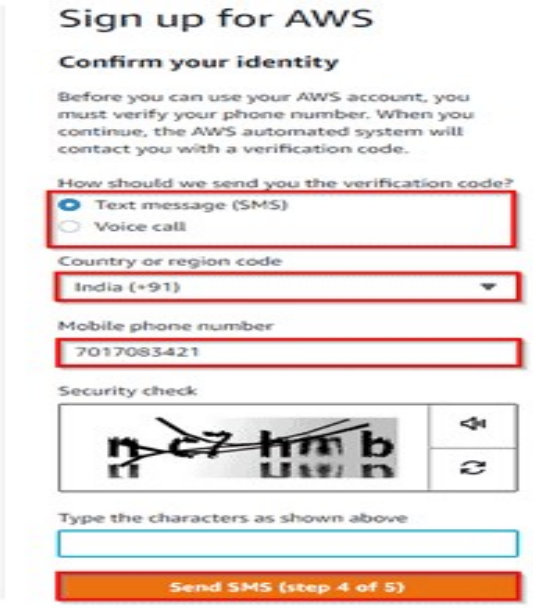
Step 5: Payment and PAN information: In this step, you must fill in your credit card /Debit Card info and billing address and click on Secure Submit.



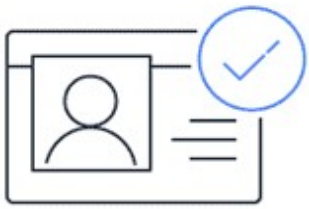
Step 6: In this step, it will take you to the payment gateway to validate your payment information, and for your credit card verification, Amazon will charge the minimum price based onyour Country. Here I have provided India, so Amazon charged 2 INR.



Step 7: Phone verification: Here you will be taken to an identity verification page that will already have your phone number, so you just have to select either “Text message or Voice call” Provide a valid phone number, Solve the captcha, and then click on Send SMS or Call Me Now(depending upon your selection).



Step 8: After clicking on Send SMS or Call me Now, you will immediately receive a call or SMS from Amazon for a verification code. Enter your code, then click on Verify Code.



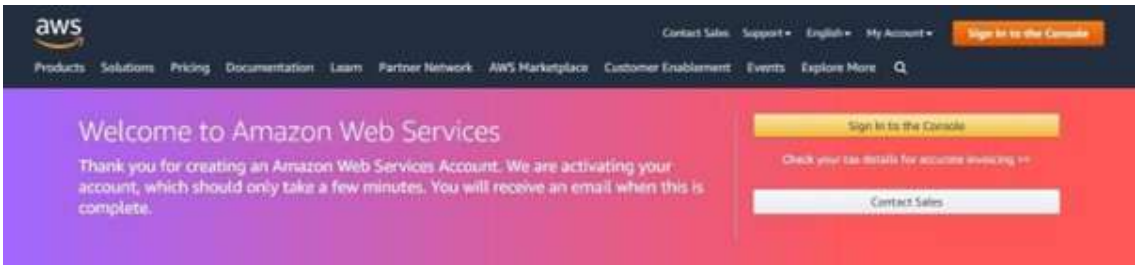
Step 9: Support plan: AWS support offers a selection of plans to meet your business needs.

Select your suitable plan then click continue.

Note: All customers receive free basic support.

Step 10: Registration Confirmation page.

Once you completed all the above steps and processes. You'll get the confirmation page below. Now your account will be processed for activation. It may take somewhere between 30 minutes to 1 hour for you to receive an email confirmation that your Amazon Cloud Services account has been activated.



we have successfully created the AWS Free Tier Account.

Here's a step-by-step guide to **create an Amazon EC2 instance, set up a web server**, and **associate an IP address** with it:

Step 1: Launch an EC2 Instance

1. **Log in** to your AWS Management Console.
2. Go to **EC2** under Services.
3. Click **"Launch Instance"**.
4. **Configure Instance Details**:
   - Name: e.g., MyWebServer
   - AMI: Choose **Amazon Linux 2 AMI**
   - Instance type: e.g., **t2.micro** (free tier eligible)
   - Key pair: Create or choose an existing key pair (used for SSH access)
   - Network: Default VPC (or your custom one)
5. **Configure Storage** (default 8GB is fine)
6. **Security Group Settings**:
   - Add rule: HTTP, Port 80, Source: Anywhere (0.0.0.0/0)
   - Add rule: SSH, Port 22, Source: Your IP
7. Click **Launch Instance**

Step 2: Connect to Your Instance

1. Once the instance is running, click **"Connect"**.
2. Choose **SSH Client** tab.
3. Open terminal (Linux/macOS) or PowerShell (Windows).
4. Use the command provided (something like):

```
ssh -i "your-key.pem" ec2-user@<your-public-ip>
```

Step 3: Install and Start a Web Server

Once connected via SSH:

1. **Update packages**:

```
sudo yum update -y
```

2.Install Apache (httpd):

```
sudo yum install httpd –y
```

3.Start Apache:

```
sudo systemctl start httpd
```

4.Enable Apache on boot:

```
sudo systemctl enable httpd
```

5.Verify by editing the homepage:

```
echo "<h1>Hello from EC2 Web Server</h1>" | sudo tee
/var/www/html/index.html
```

Step 4: Associate a Static IP (Elastic IP)

1. Go to **EC2 > Network & Security > Elastic IPs**.
2. Click **Allocate Elastic IP address**.
3. Once allocated, select the new IP and click **Actions > Associate IP**.
4. Associate it with your instance's network interface.

Step 5: Test Your Web Server

- Open a browser and enter your **Elastic IP**.
- You should see:
  Hello from EC2 Web Server

Let me know if you want to automate this with a script or use a specific OS like Ubuntu

Cloud Computing Lab

Experiment-3

Create an Amazon EC2 instance and set up a web-server on the instance and associate an IP address with the instance.

Amazon Web Services (AWS) provides a powerful cloud computing platform with a wide range of services. One of the core services offered by AWS is Amazon Elastic Compute Cloud (EC2), which allows users to launch virtual servers in the cloud.

An **Amazon EC2(Elastic Compute Cloud)** instance is a virtual server in the cloud that allows users to run applications and workloads on AWS. It's essentially renting a computer in a data center, providing on-demand computing resources that can be scaled up or down as needed. EC2 instances offer various configurations of CPU, memory, storage, and networking, allowing users to choose the best fit for their specific requirements.

The **AWS Management Console** is a centralized platform designed for accessing and managing AWS cloud services. It offers a graphical user interface (GUI) that simplifies the management of various AWS resources, such as compute power (EC2), storage (S3), databases (RDS), and more.

Creating an Amazon EC2 instance involves several steps within **the AWS Management Console**:

1. Log in to your AWS account

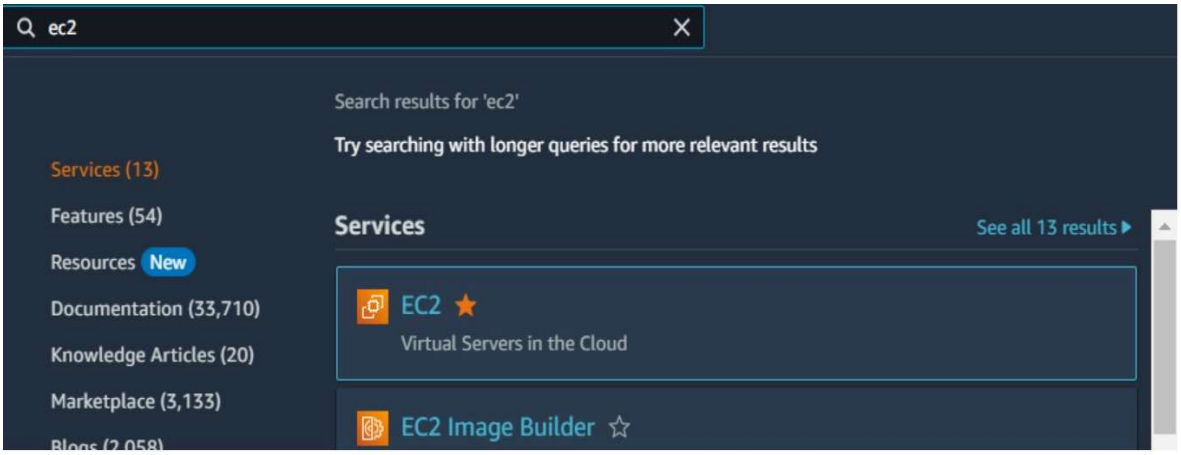
Log in to the AWS Management Console and set up your root account. If you don't already have an account, you will be prompted to create one.

- With the AWS Free Tier, you can get 750 hours/month of select EC2 instances for free.

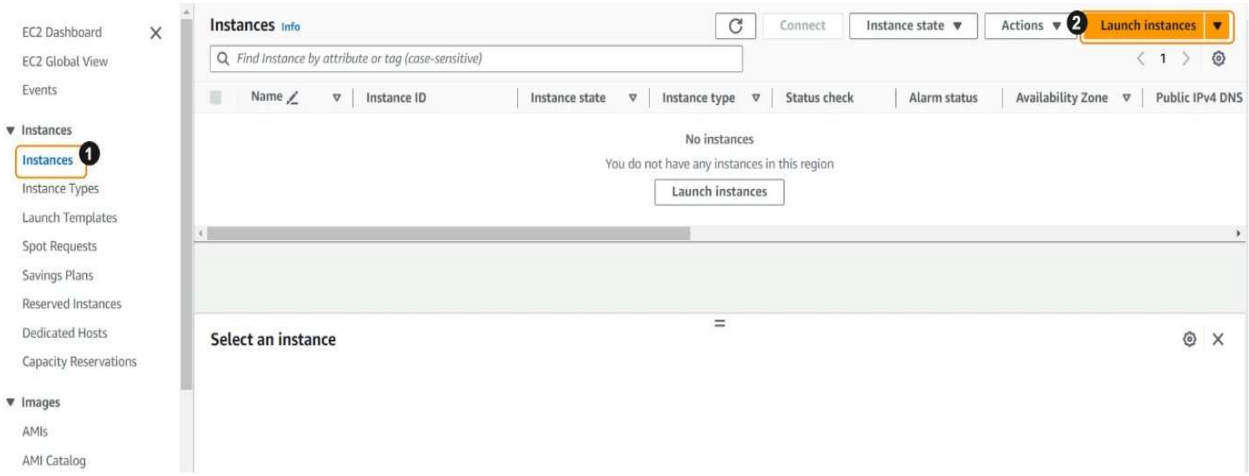
2. Launch your instance

Identify which instance type is best for your workload. For your first instance, we recommend a low-cost, general-purpose instance type—t2.micro—and Amazon Machine Image (AMI)—Amazon Linux 2 AMI—which are both free-tier eligible.

Step-1: Search '*ec2*' in the AWS Management Console search bar. Click **EC2** on the search results.

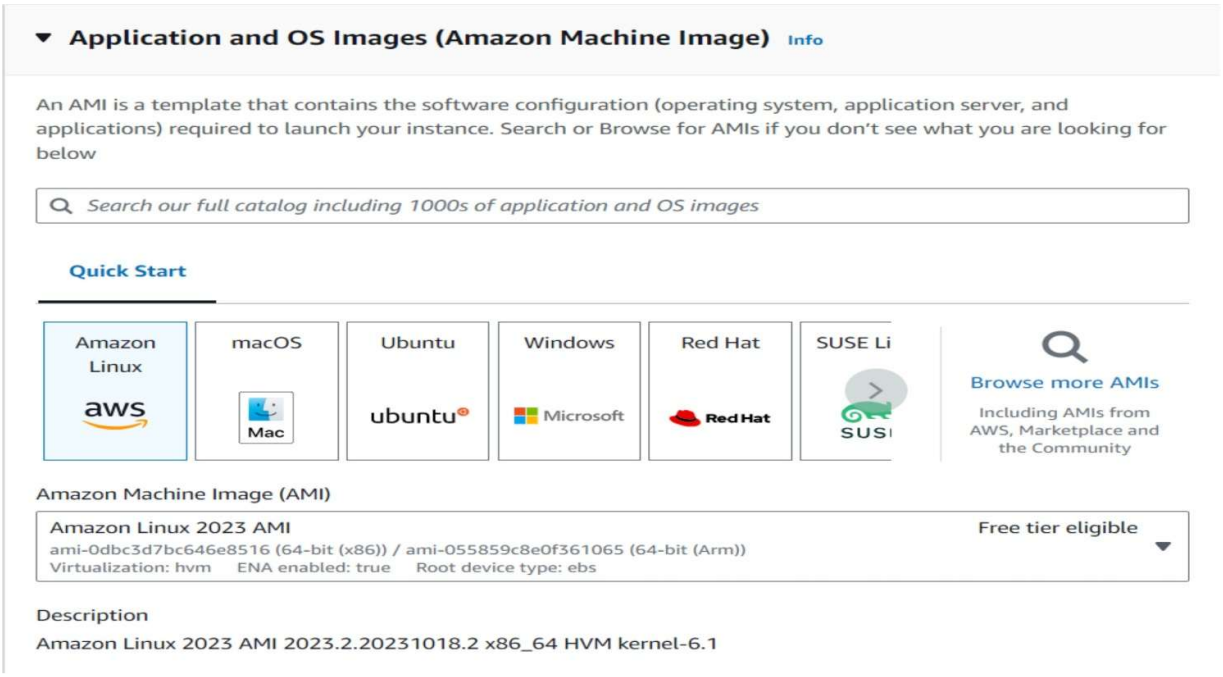


Step-2: In the left window pane, select **Instances**, then click the **Launch instances** option.

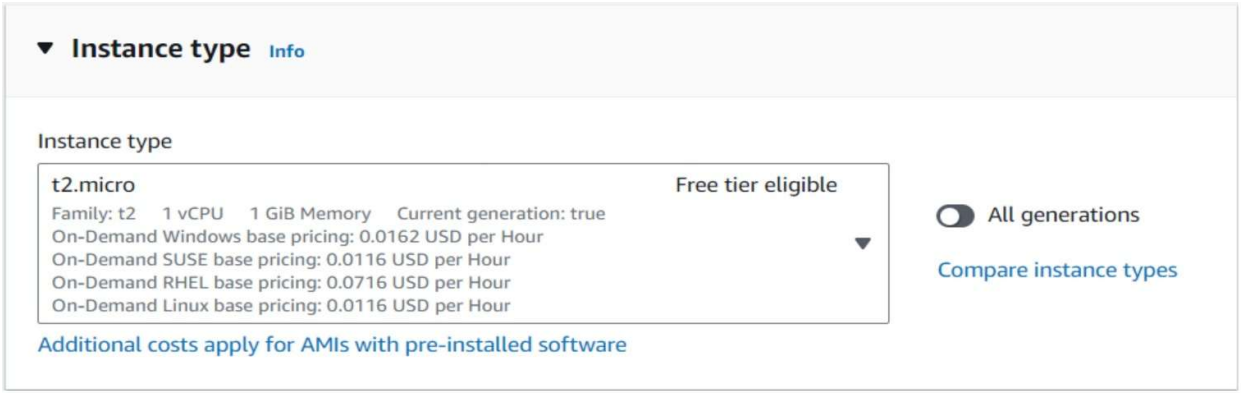


Step-3: Name the instance '***my-web-server***' or any name that you prefer.

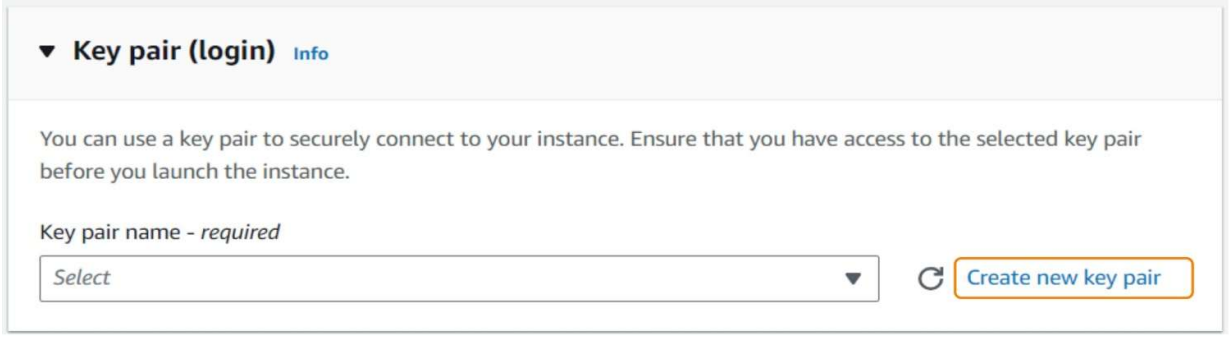
Step-4: Under the **Application and OS Images** section, click the default **Amazon Linux** AMI.



Step-5: Under the **Instance Type** section, select **t2.micro**.



Step-6: Under the **Key Pair** section, click **Create new key pair**.



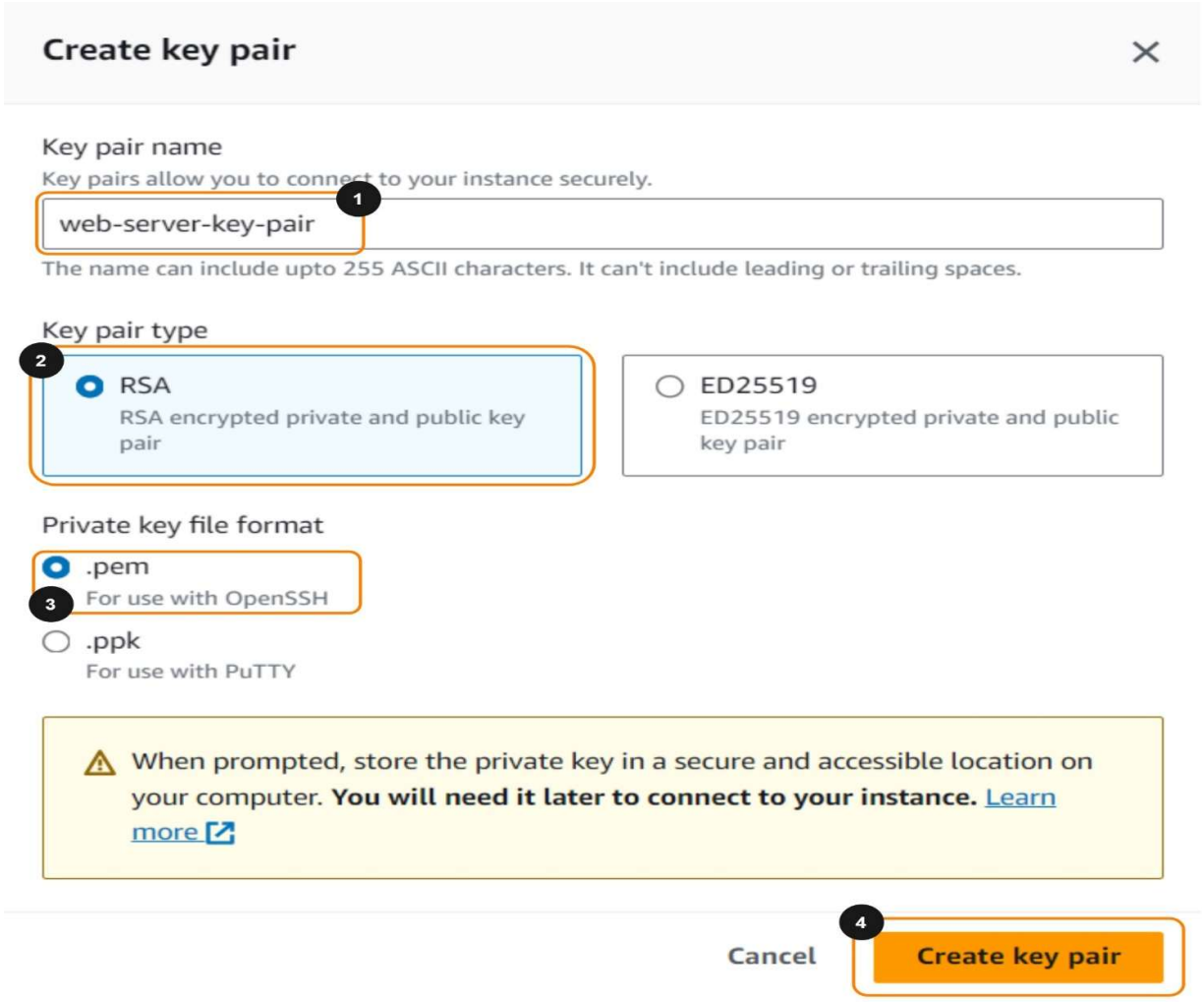
3.Configure your instance

Here are some guidelines when setting up your first instance:

- **Security group**: Create your own firewall rules or select the default VPC security group.
- **Storage**: EC2 offers both magnetic disk and SSD storage. We recommend starting with Amazon EBS gp2 volumes.
- Choose "Launch Instances" to complete the setup.

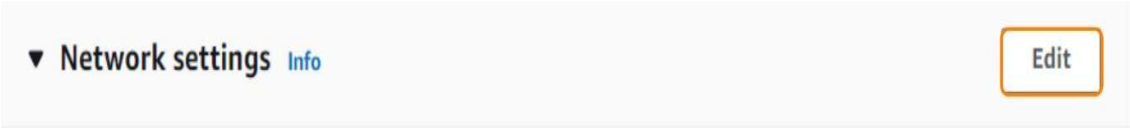
  ** Note: We will use the key pair file (.pem) later.*

Step-1: Enter a key pair name and follow the configurations below. Then, click **Create key pair.**



After creating a key pair, the private key will be downloaded to your computer. Remember to note the location of this file, as you'll need it later to SSH to your EC2 instance.

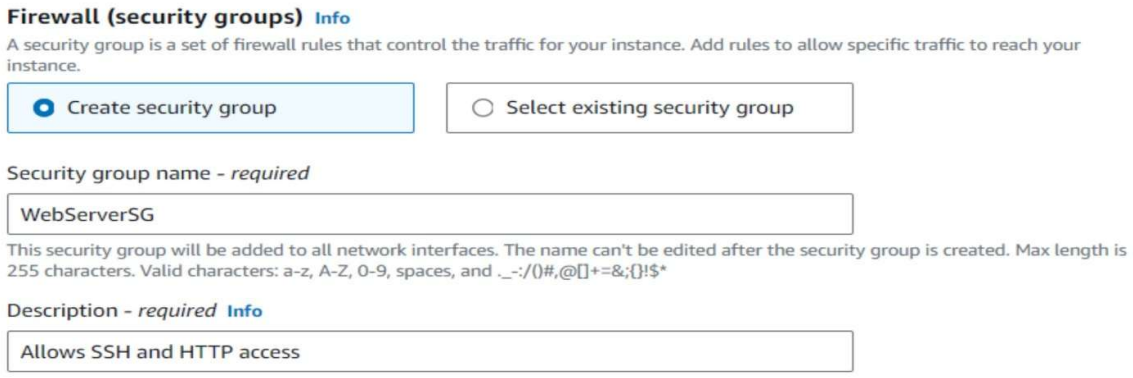
Step-2. Under the **Network settings** section, click **Edit.**



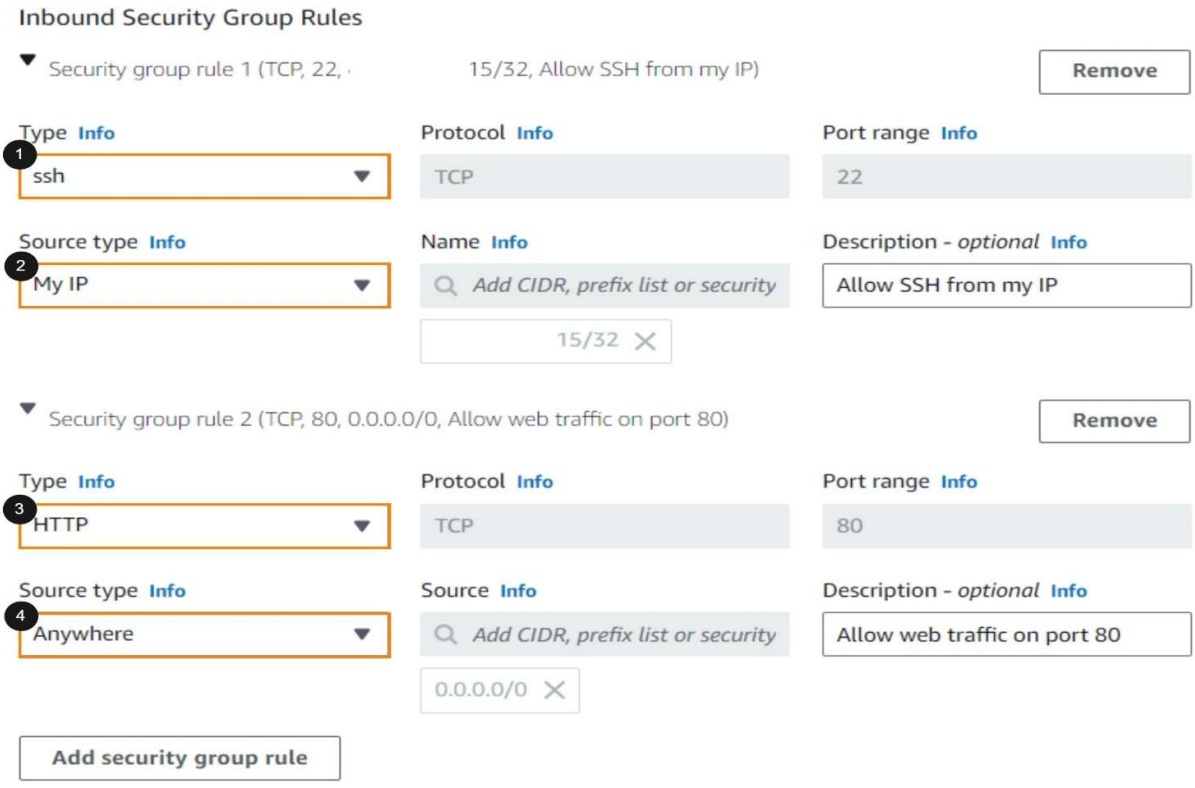
Step-3. Scroll down the **Firewall (Security Groups)** option.

a. Enter '**WebServerSG***'* for the security group name.

b. For Description, enter '***Allows SSH and HTTP access*****'.**



Step-4: Add two inbound security group rules with the following configuration**.**



Inbound rule 1

Type	Source Type
SSH	My IP

Inbound rule 2

Type	Source Type
HTTP	Anywhere (0.0.0.0/0)

Step-5. In the right window pane, at the bottom section, click **Launch instance**.

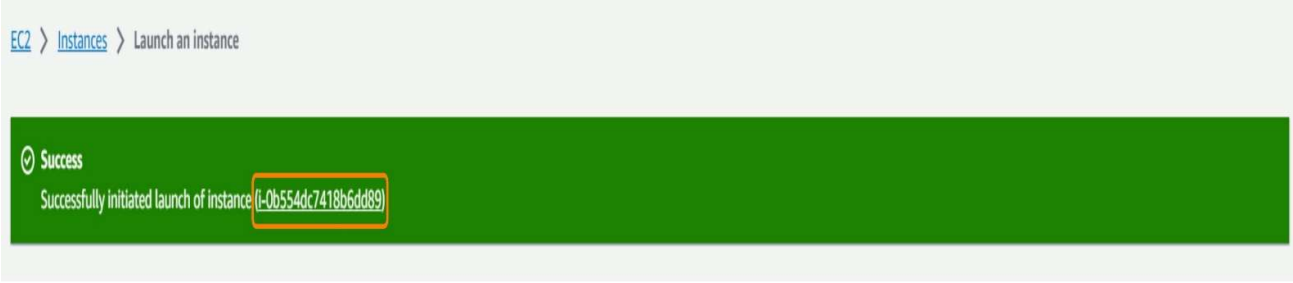
Setting up the web server.

4.Connect to your instance

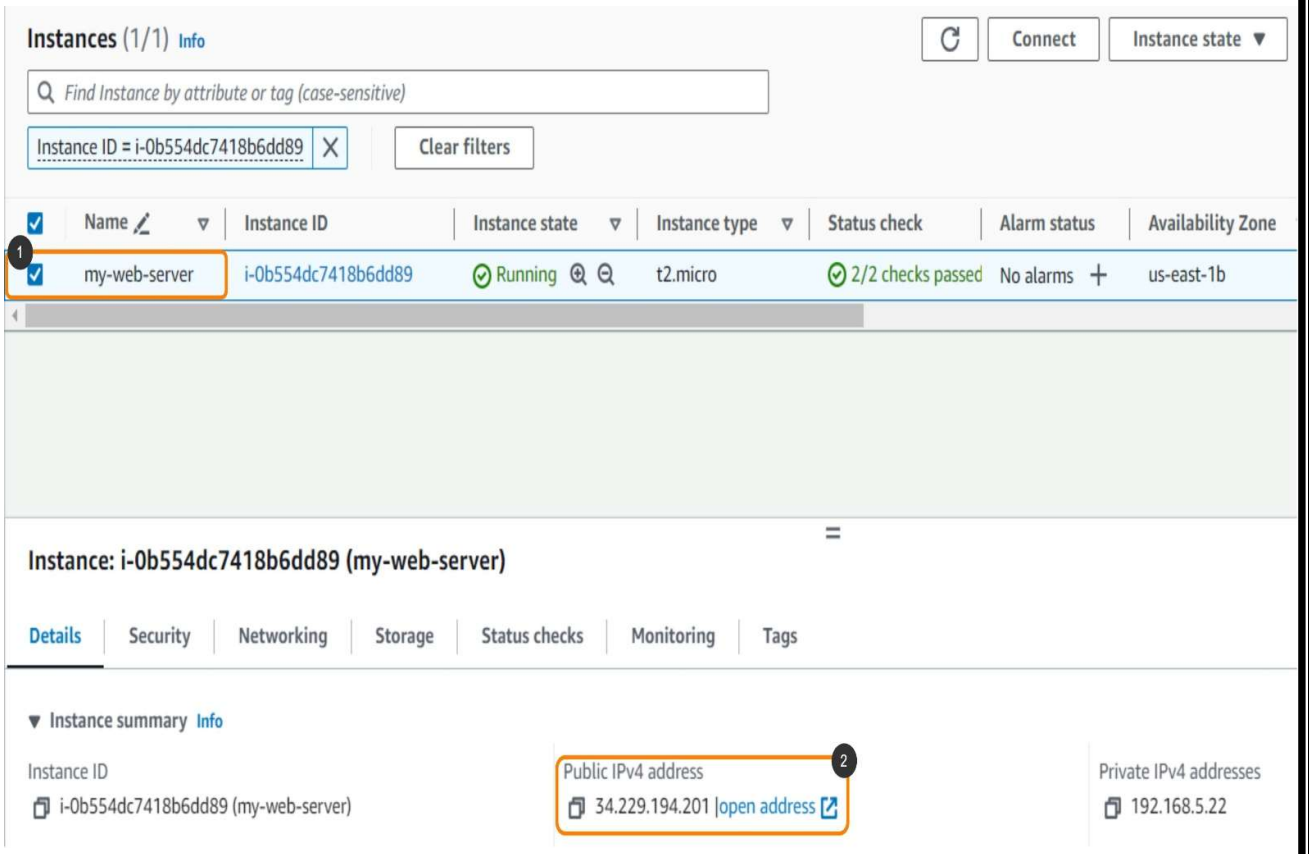
After launching your instance, you can connect to it and use it the way you'd use a computer sitting in front of you. There are several ways to connect to the console depending on the operating system. We recommend using EC2 Instance Connect, an easy-to-use browser-based client.

- Select the EC2 instance that you created and choose "Connect.“
- Select “EC2 Instance Connect.”
- Choose “Connect.” A window opens, and you are connected to your instance.

Step-1: After the instance is created successfully, click the **instance ID.**



Step-2: Tick the checkbox next to your instance name. Then, copy the Public IP address of your instance and paste it somewhere you can easily retrieve it later.



In this lab, we'll be using the SSH utility from OpenSSH. It usually comes built-in with Windows 10 and 11, Mac, and most Linux distributions. If your operating system doesn't have it pre-installed, ensure you install it first before proceeding.

Step-3: Open up a terminal, then run the command below to connect to your instance via SSH.

```
ssh -i /path/to/YOUR-KEY.pem ec2-user@YOUR-EC2-PUBLIC-IP
```

Ensure that you reference the correct path to your private key pair and that you use the correct public IP of your EC2 instance.

Once connected, your shell prompt should change to something similar to ec2-user@ip-192-168-5-22:~$, confirming that you're now connected to your EC2 instance.

In the next steps, you will configure the necessary settings to set up a web server on the EC2 instance.

Step-4: Run the command below to update the system**.**

```
sudo yum update -y
```

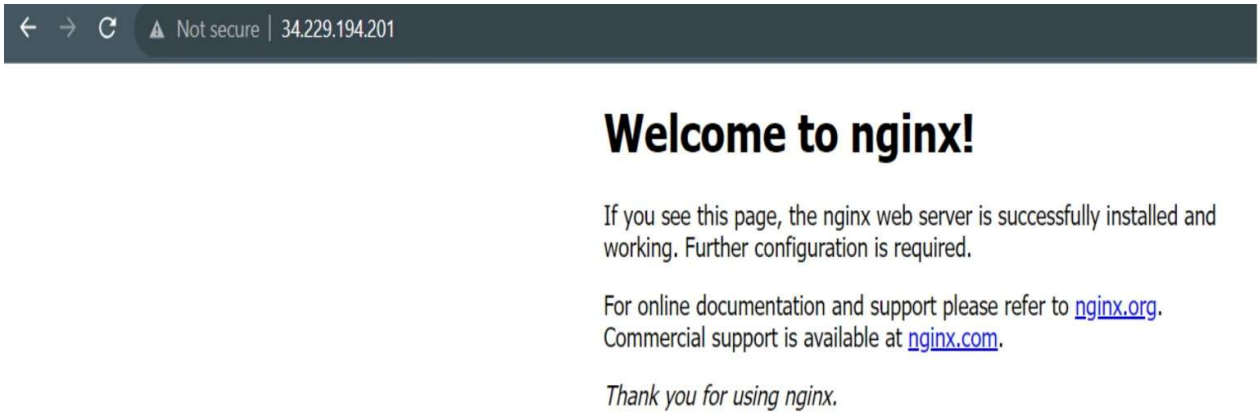
Step-5: Once the update is completed, install Nginx.

```
sudo yum install nginx -y
```

Step-6: Start the Nginx Service.

```
sudo service nginx start
```

Step-7: Enter your EC2 instance's public IP in your browser. The default Nginx welcome page should be displayed.



Now, let's replace the welcome page with a custom one.

Step-8: Go to the /usr/share/nginx/html/

```
cd /usr/share/nginx/html/
```

Step-9: Create a custom HTML page**.**

```
echo '<h1>Welcome to my web page!</h1>' | sudo tee mypage.html > /dev/null
```

Step-10: Let's override the default Nginx configuration by creating a new configuration file in the /etc/nginx/conf.d/

```
sudo vi /etc/nginx/conf.d/server.conf
```

Step-11: Press i to enter Insert mode in Vi and paste the following configuration.

```
server {
    listen 80 default_server;
    server_name _;
    root /usr/share/nginx/html;

    location / {
        index mypage.html;
    }
}
```

Step-12: Press the Escape button and enter :wq! to exit and save your changes.

Step-13: Reload Nginx for the changes to take effect.

```
Sudo nginx -t && sudo service nginx reload
```

Step-7: Reload your browser to see the changes you've made.

Congratulations! You've successfully set up a web server on an Amazon EC2 instance using Nginx. You've also hosted a custom web page, giving you foundational skills in web hosting on the cloud. This is just the beginning. As you continue to explore, you can experiment with different configurations, host more complex web applications, and even integrate databases.

5.Terminate your instance

Amazon EC2 is free to start (learn more), but it is important that you terminate your instances to prevent additional charges. The EC2 instance and the data associated will be deleted.

Select the EC2 instance, choose "Actions," select "Instance State," and then select "Terminate."

.EX.No:4 Installation of a GoogleApp Engine

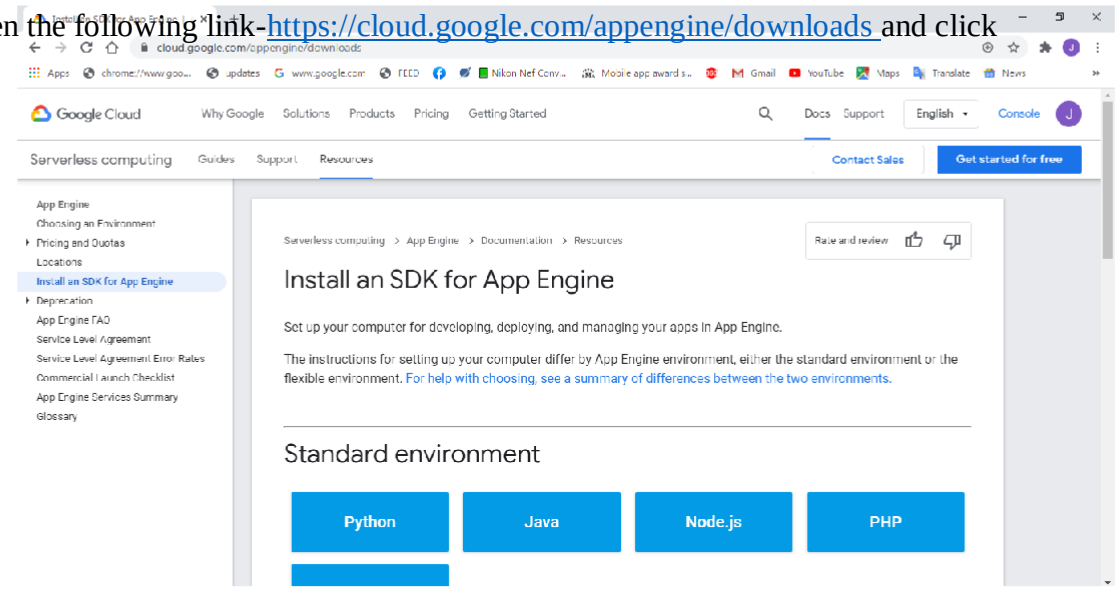
Date:

Aim:

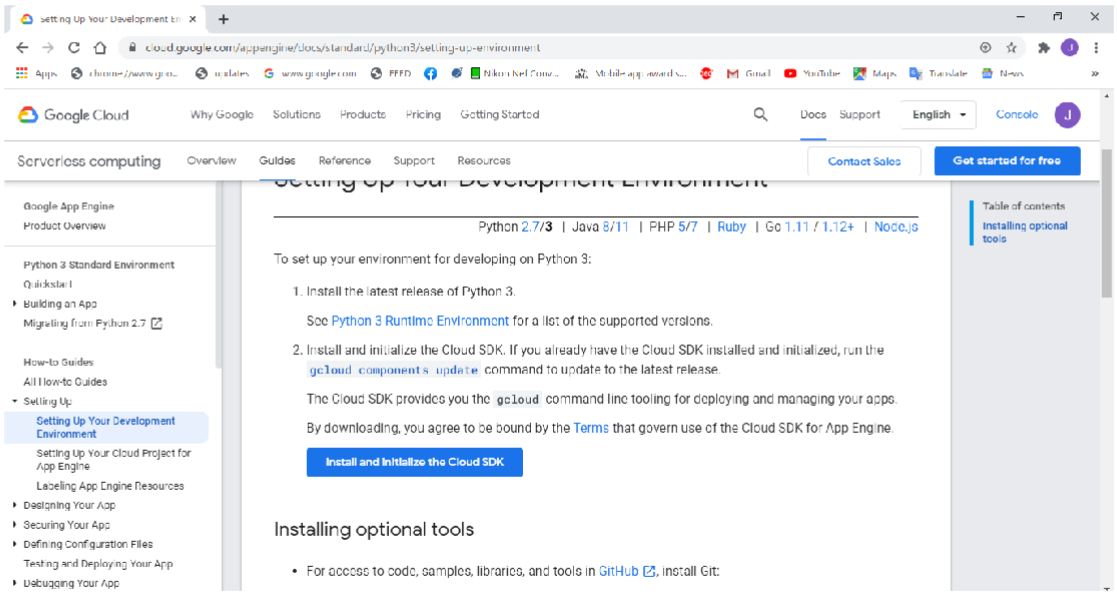
To install the Google app engine in the system.

Procedure:

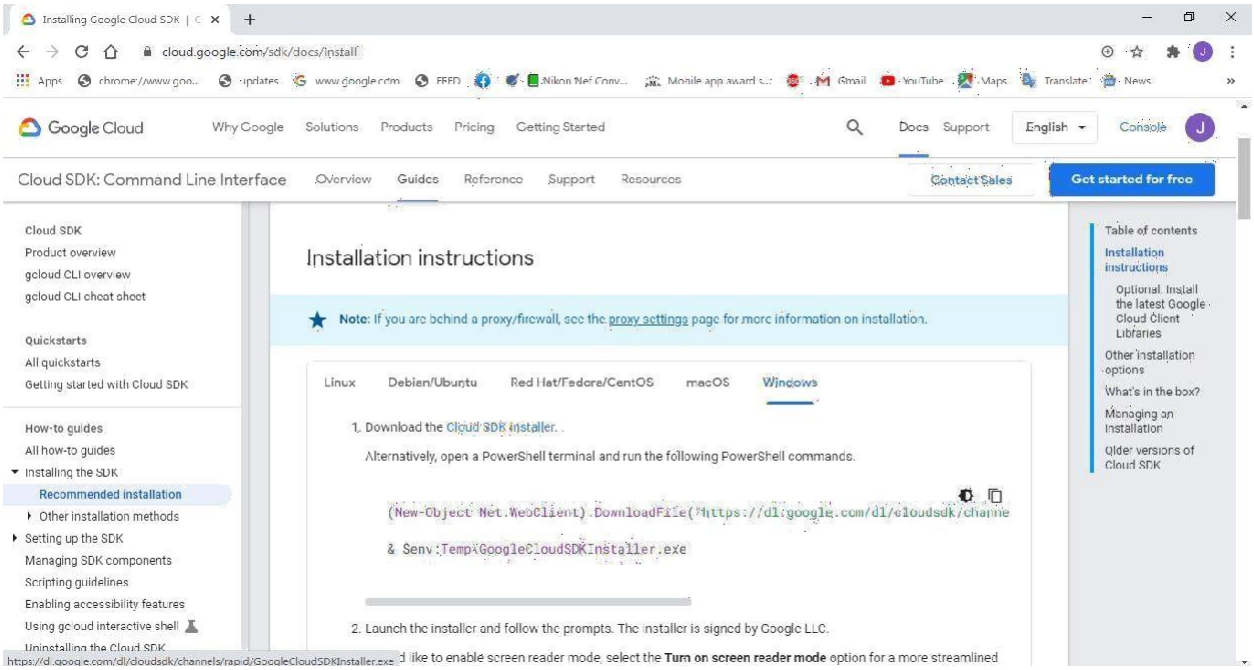
Step1: Open the following link-https://cloud.google.com/appengine/downloads and click python.



Step2: Select setting up your environment development and click on install and initialize the cloud sdk.

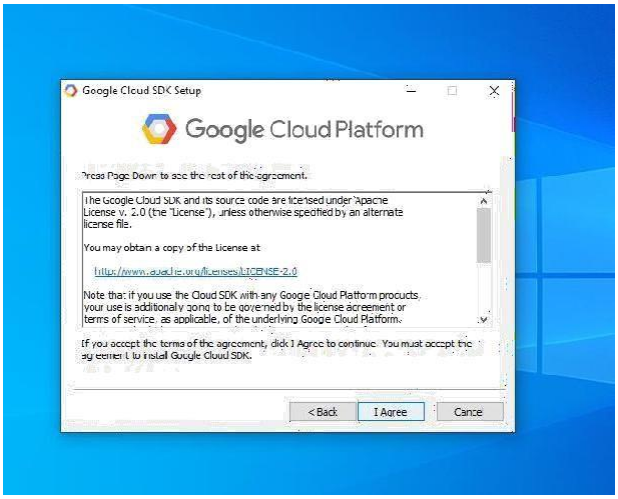


Step3:Download the sdk installer and install it.

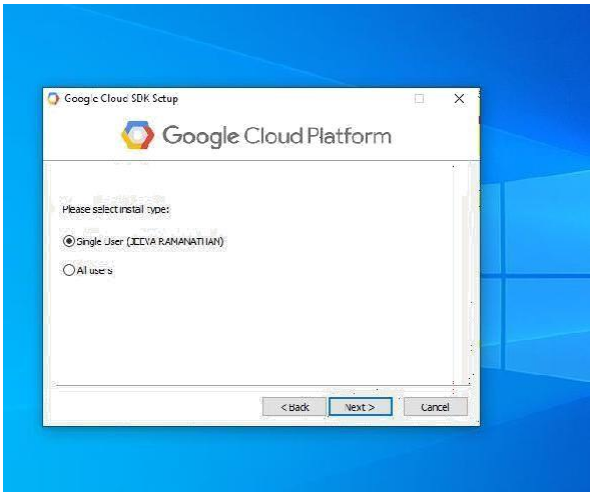


Step4: Click Next.

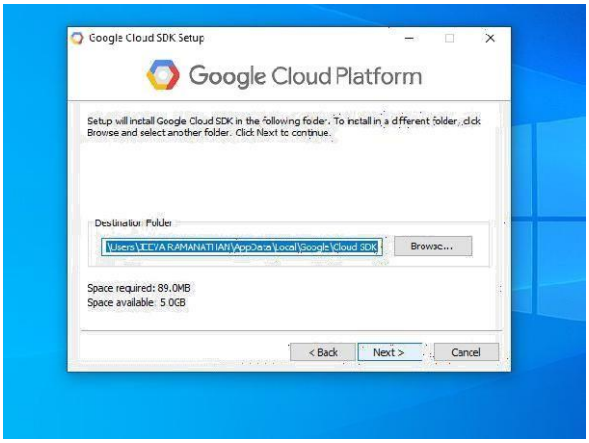
Step5:Click I Agree.



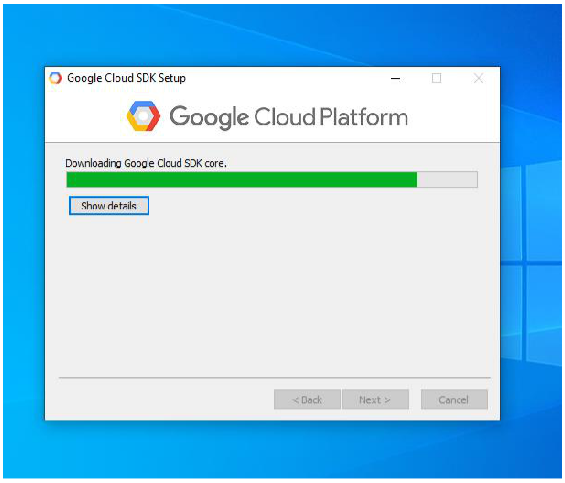
Step6: Select single user and click Next.

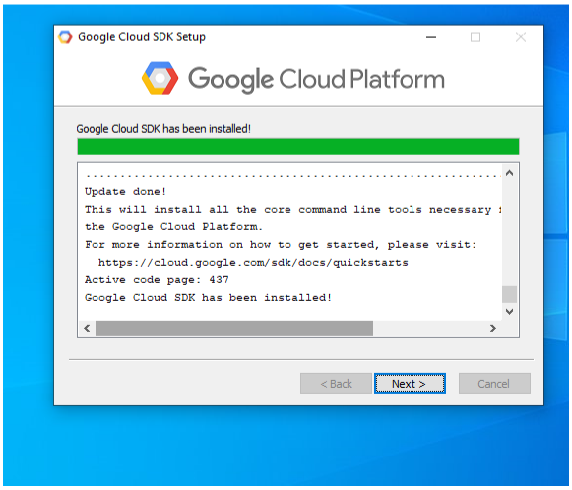


Step7: Select the destination location and click Next.

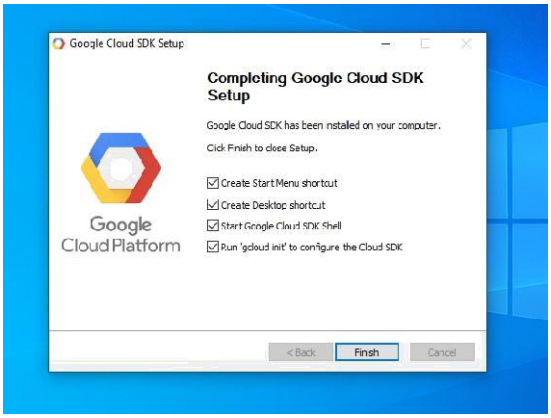


Step8: Downloading all the requirements and installing

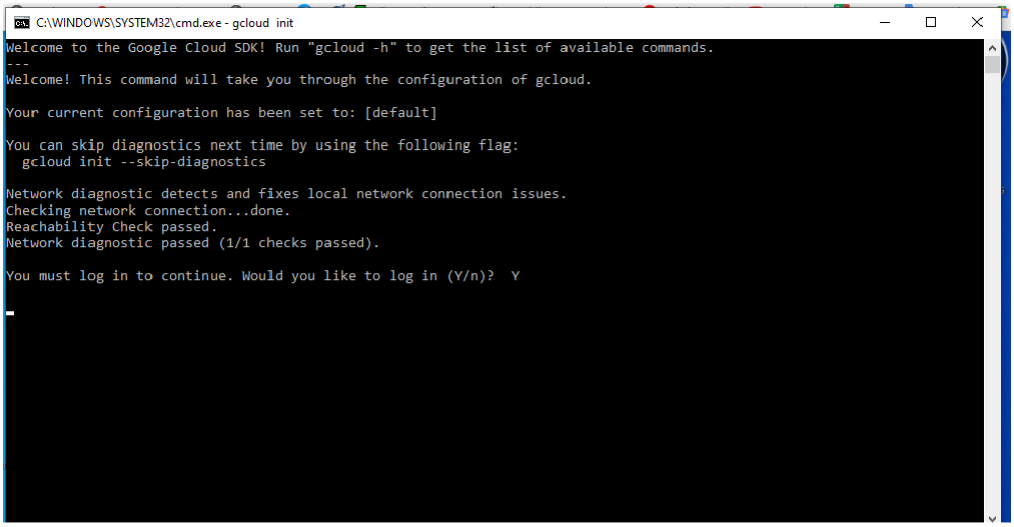


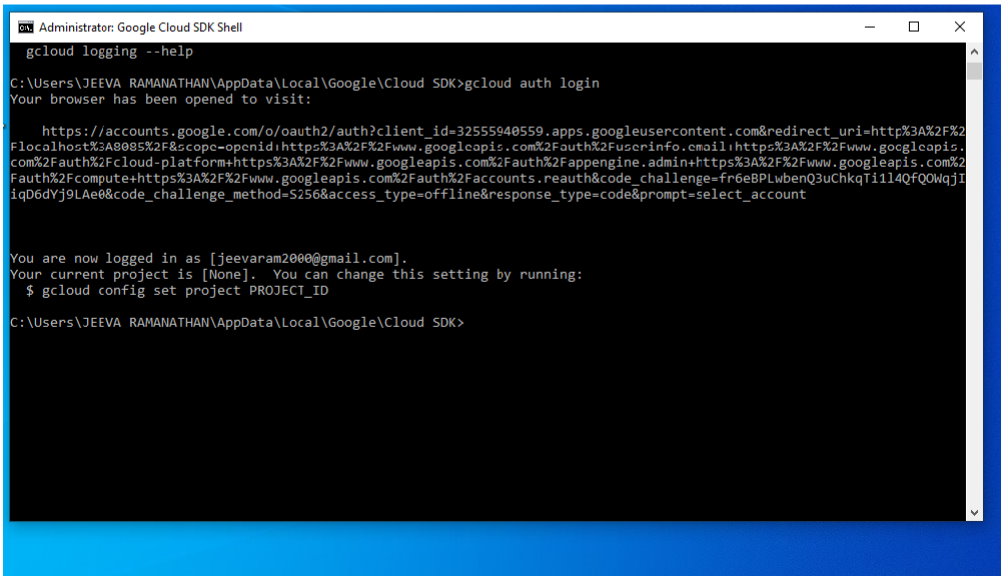


Step9:Click Finish.



Step10:Once successfully installed cmd line in login with your Google account.





Result:

Thus googleapp engine is installed successfully in the system.

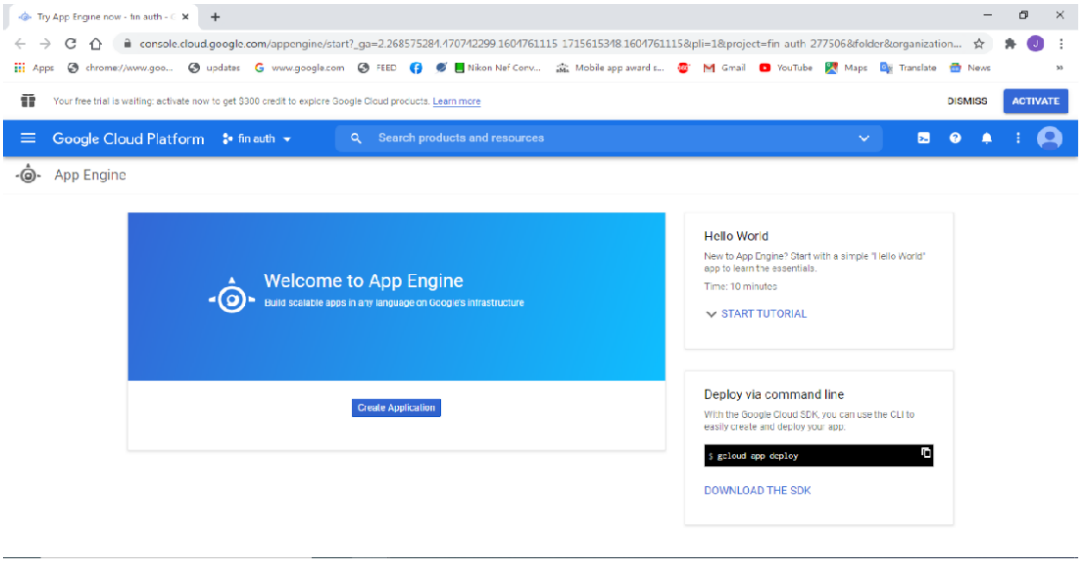
EX.No:4 Launch a web application using GAE launcher

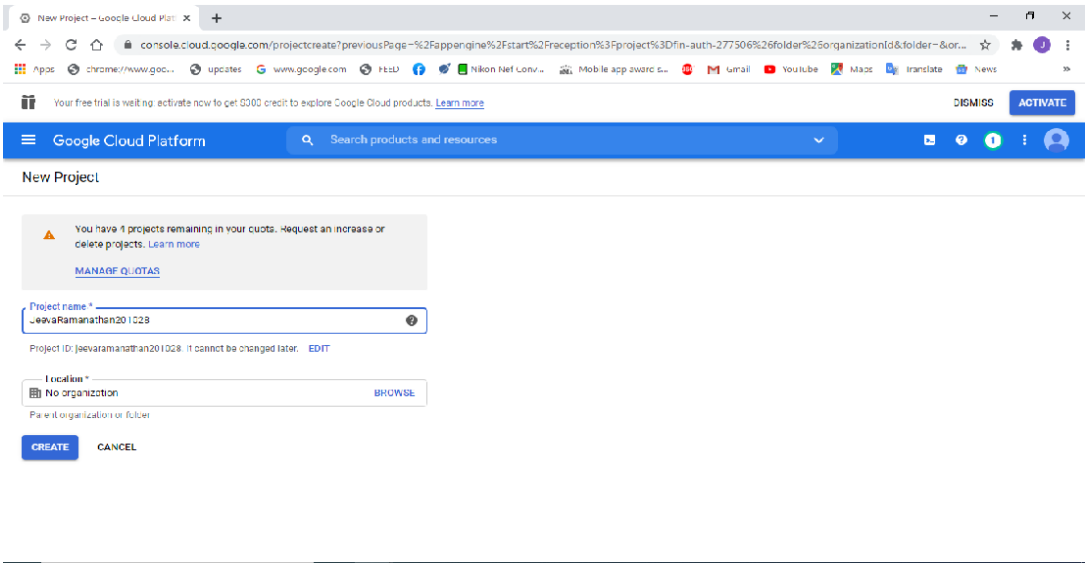
Aim:

To launch the web applications by using the GAE launcher.

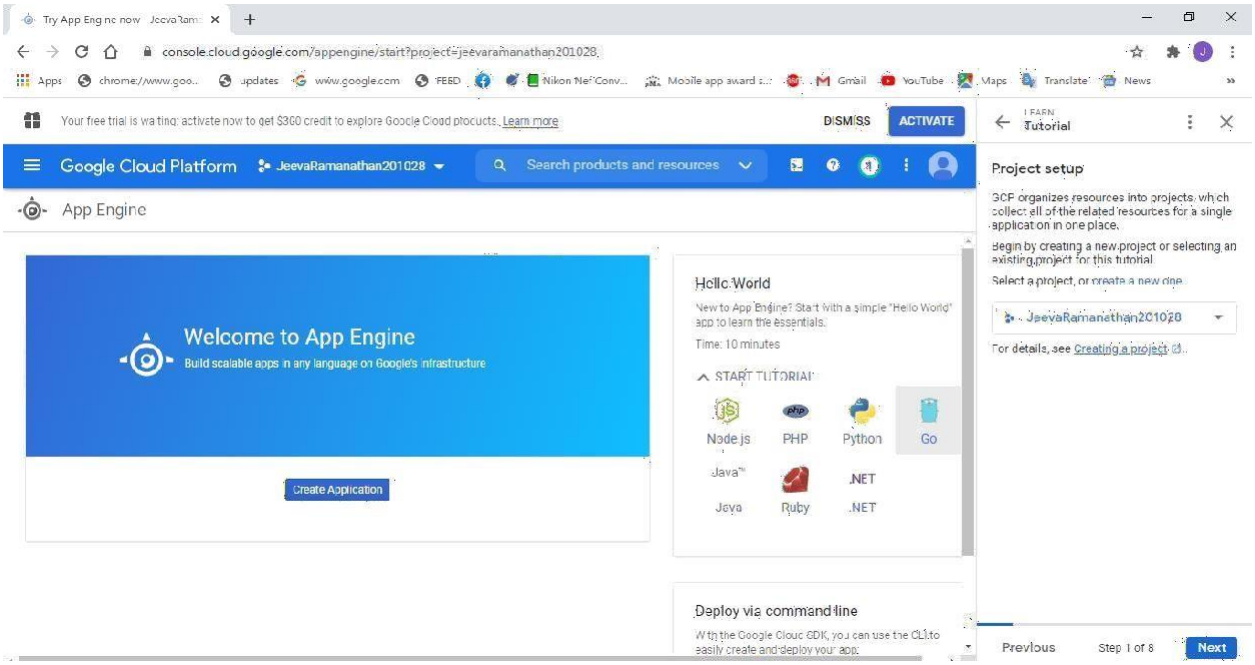
Procedure:

Step 1:Go to the following website https://console.cloud.google.com/start/appengine?_ga=2.268575284.470742299.1604761115-1715615348.1604761115&pli=1 and create a new project.



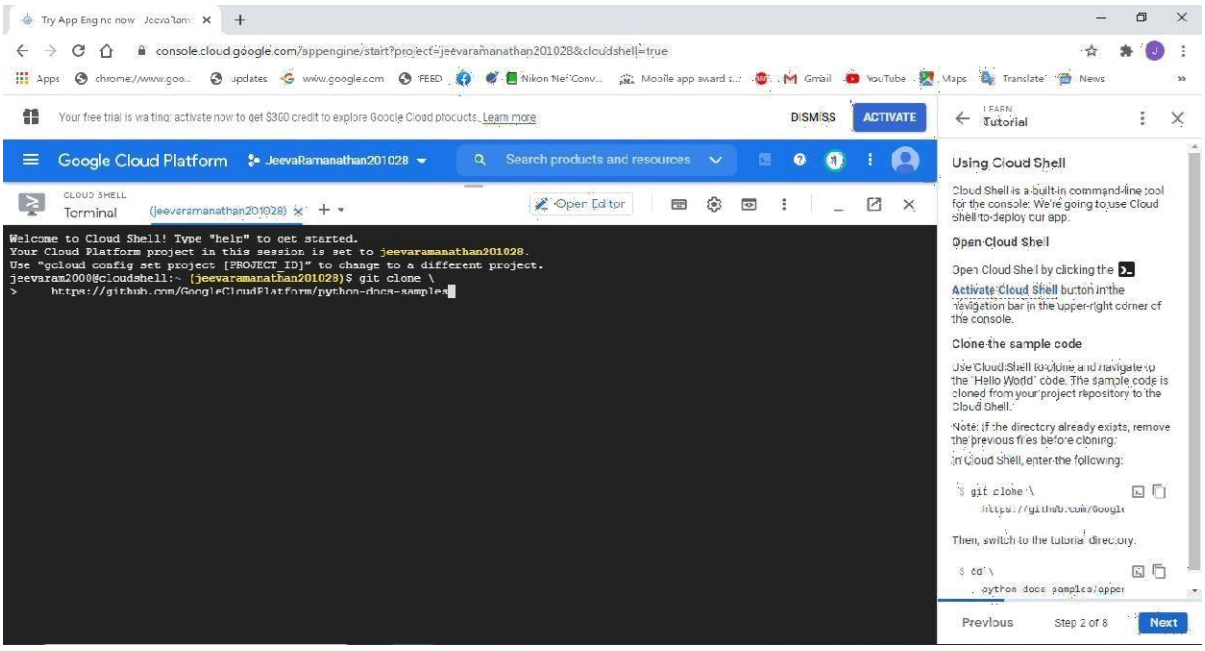


Step2:Select python and click next.

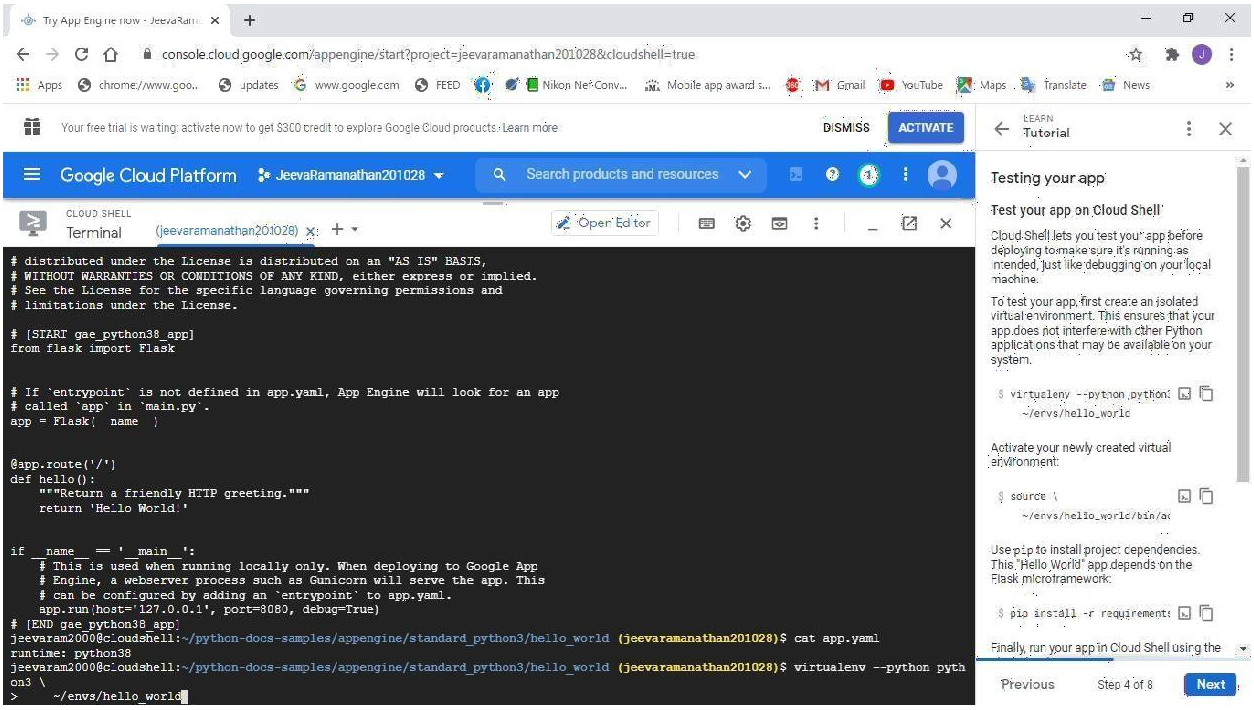


Step3: Open the cloud shell and follow the steps in the tutorial.

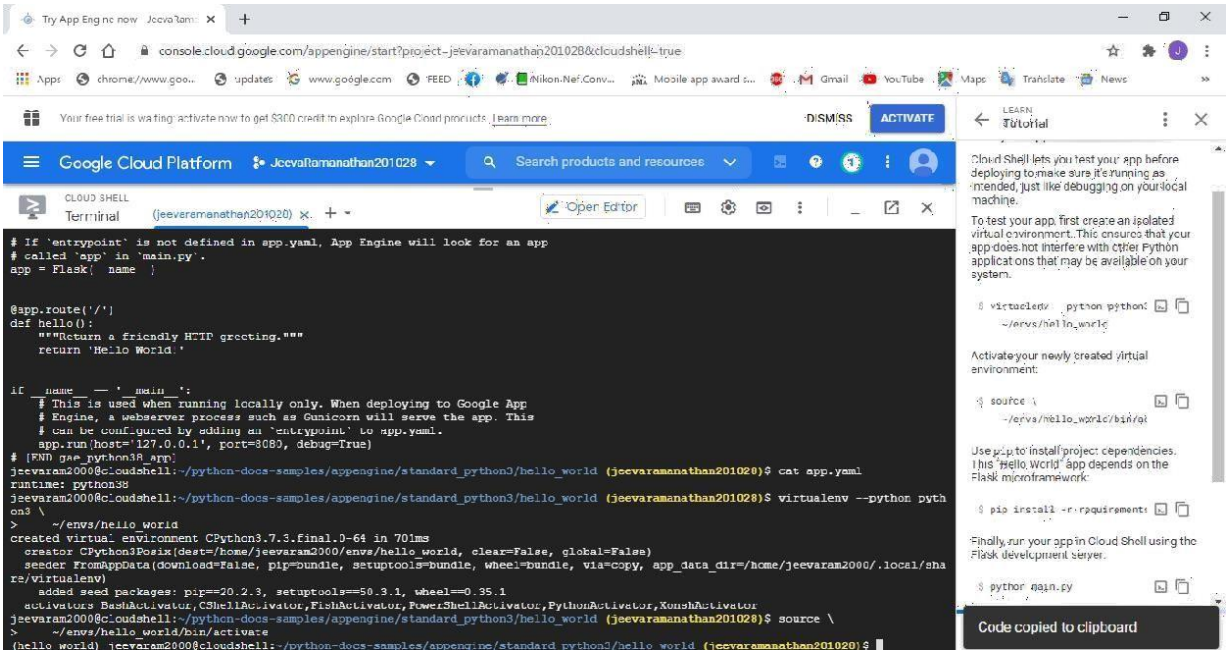
Clone the repository by using the given command



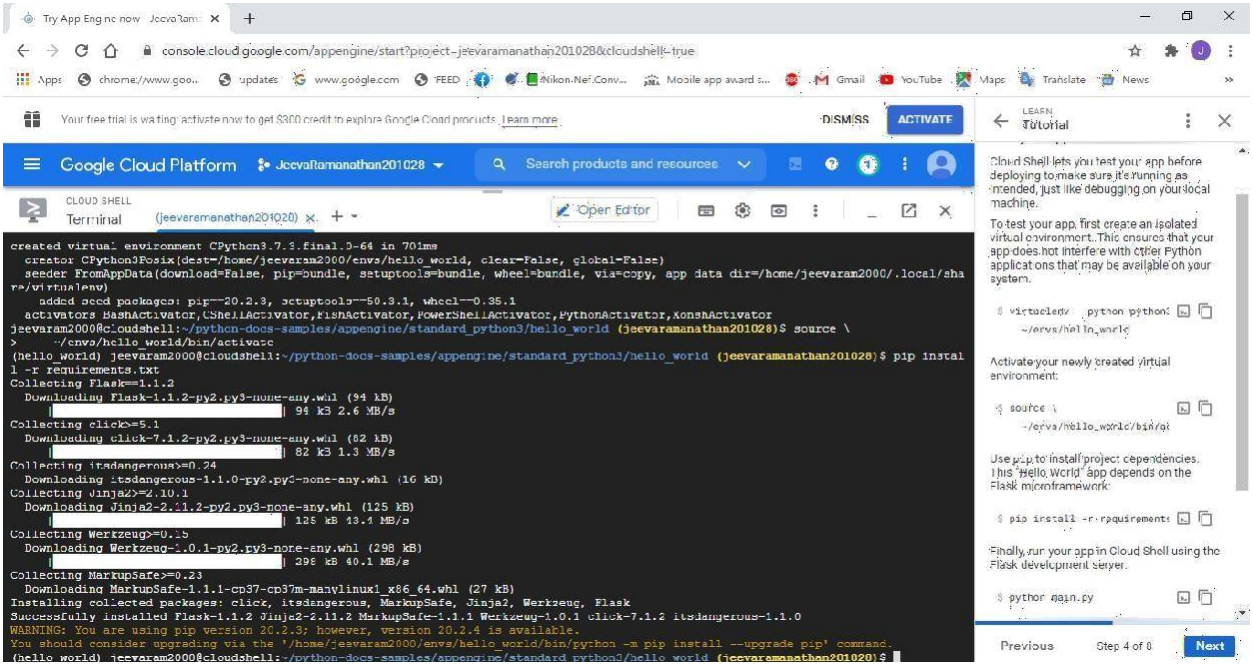
Step4: Create the virtual environment

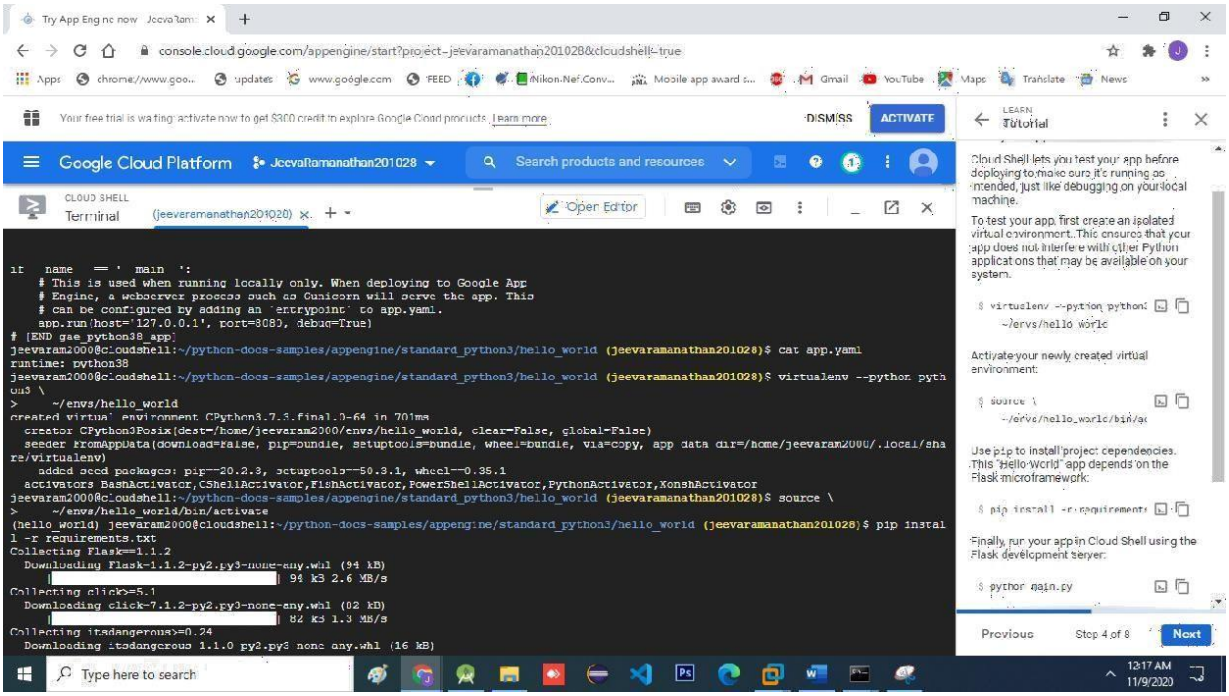


Step5: Activate your virtual environment.

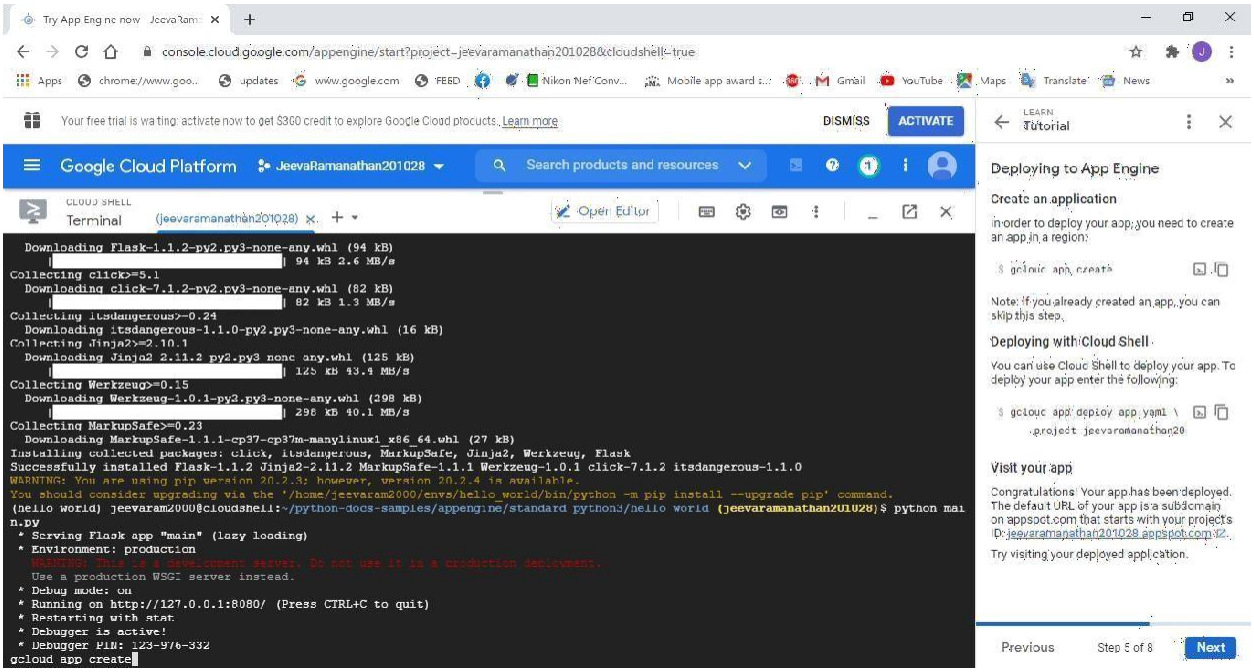


Step6: Installing requirements and run the app.

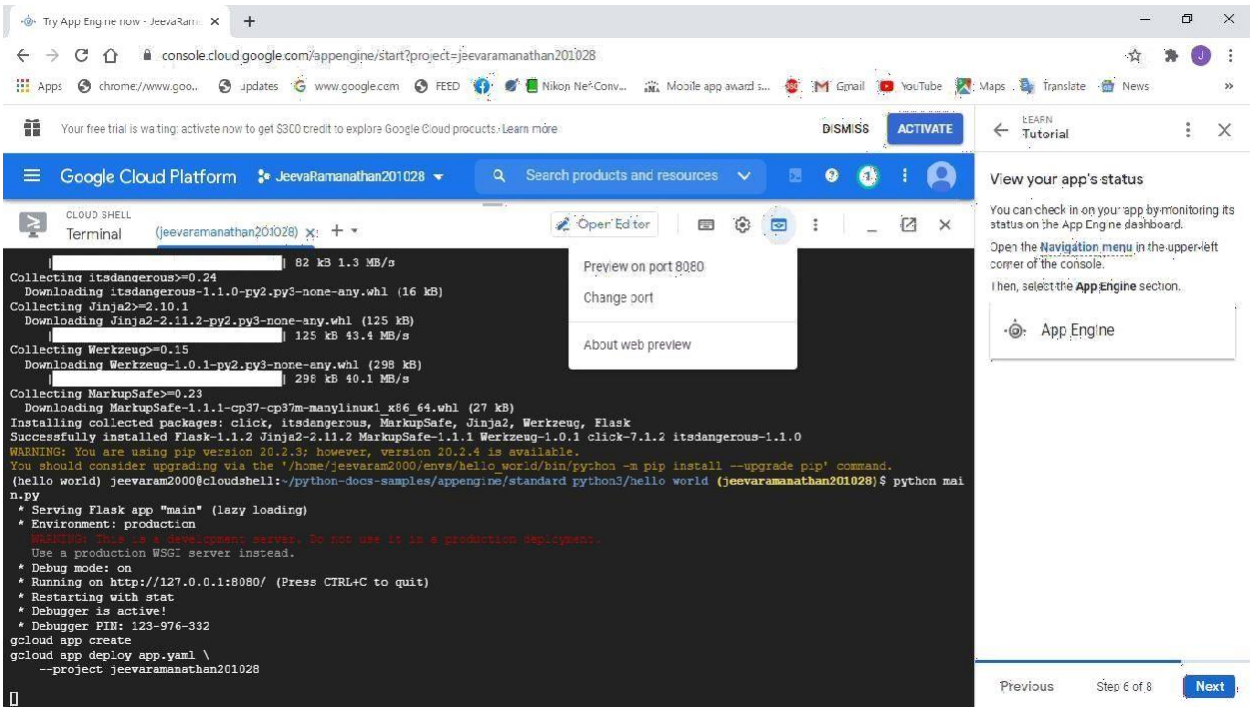




Step7: Create an application and deploy it in cloud shell.



Step8: Click preview on port 8080 to see your deployed application



Step9: Finally the application is deployed and the output is seen.

Result:

Thus a web application is launched by using the GAE launcher and the output is obtained successfully.

Week 5

Title

Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Requirements

1. Java JDK and JRE
2. CloudSim archives (CloudSim4)
3. Eclipse IDE

Theory

A) CloudSim

1. CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services.
2. It is developed by the CLOUDS Lab organization and is written entirely in Java.
3. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.
4. If you were to deploy an application or a website on the cloud and wanted to test the services and load that your product can handle and also tune its performance to overcome bottlenecks before risking deployment, then such evaluations could be performed by simply coding a simulation of that environment with the help of various flexible and scalable classes provided by the CloudSim package, free of cost.

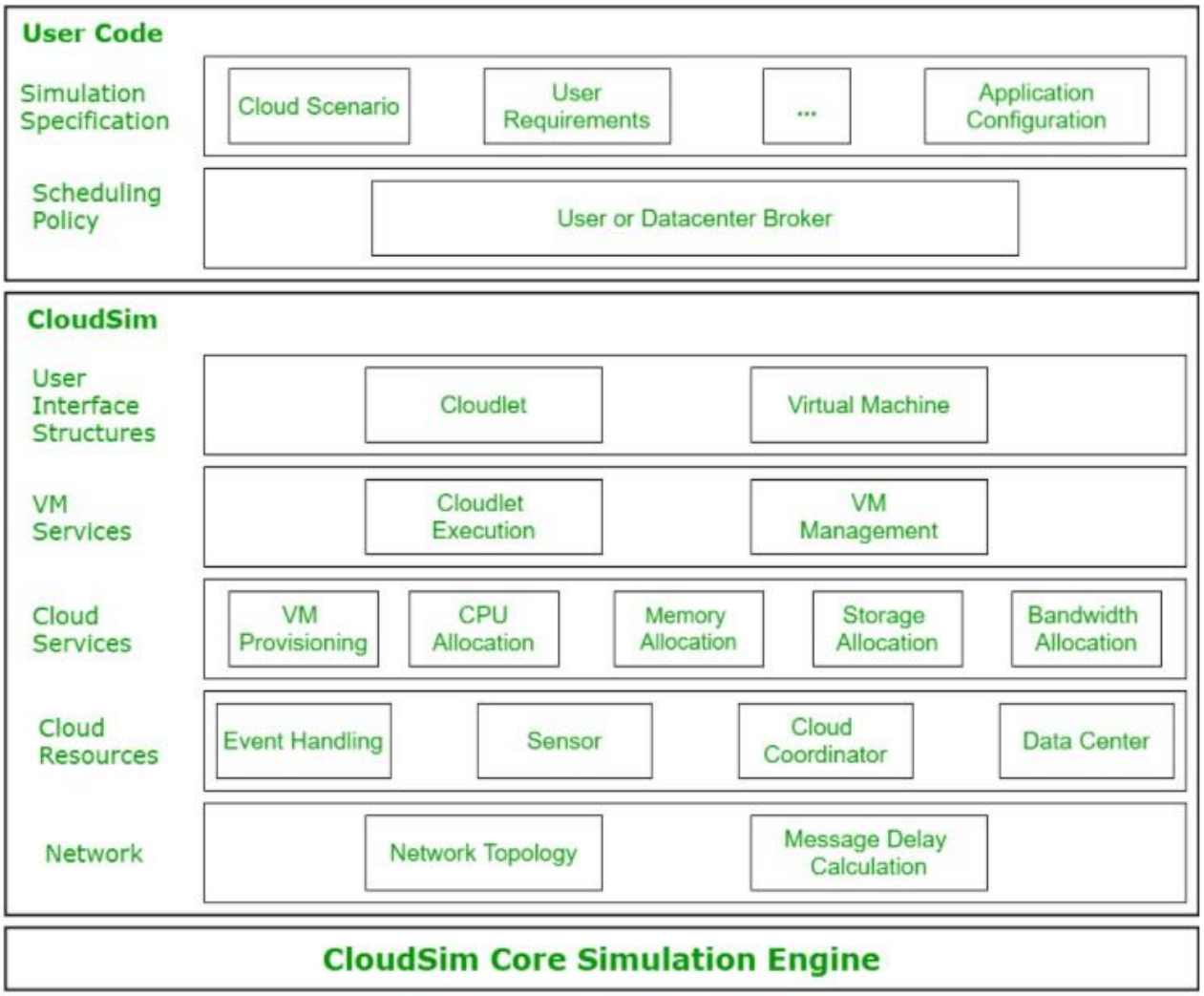
B) Benefits of CloudSim

1. No capital investment involved

2. Easy to use and Scalable
3. Risks can be evaluated at an earlier stage
4. No need for try-and-error approaches

C) Architecture

1. CloudSim has a layered architecture which separates the User Code and the simulation environment.
2. It can be depicted as follows



CloudSim Layered Architecture

D) CloudSim Components

- **Datacenter**: used for modelling the foundational hardware equipment of any cloud environment, that is the Datacenter. This class provides methods to specify the functional requirements of the Datacenter as well as methods to set the allocation policies of the VMs etc.
- **Host**: this class executes actions related to management of virtual machines. It also defines policies for provisioning memory and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
- **VM**: this class represents a virtual machine by providing data members defining a VM's bandwidth, RAM, mips (million instructions per second), size while also providing setter and getter methods for these parameters.
- **Cloudlet**: a cloudlet class represents any task that is run on a VM, like a processing task, or a memory access task, or a file updating task etc. It stores parameters defining the characteristics of a task such as its length, size, mi (million instructions) and provides methods similarly to VM class while also providing methods that define a task's execution time, status, cost and history.
- **DatacenterBroker**: is an entity acting on behalf of the user/customer. It is responsible for functioning of VMs, including VM creation, management, destruction and submission of cloudlets to the VM.
- **CloudSim**: this is the class responsible for initializing and starting the simulation environment after all the necessary cloud entities have been defined and later stopping after all the entities have been destroyed.

E) SJF algorithm

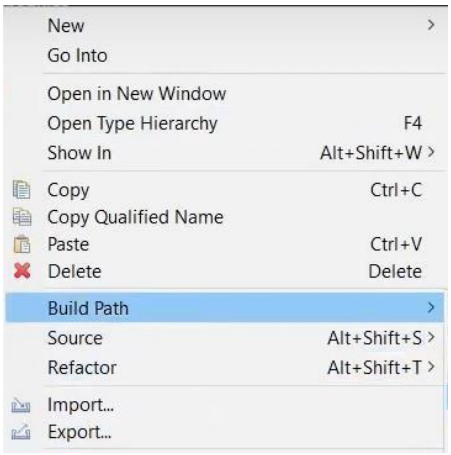
1. SJF stands for Shortest Job First
2. Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.
3. It is a Greedy Algorithm.
4. It may cause starvation if shorter processes keep coming. This problem can be solved using the concept of ageing.
5. It is practically infeasible as Operating System may not know burst time and therefore may not sort them. While it is not possible to predict

execution time, several methods can be used to estimate the execution time for a job, such as a weighted average of previous execution times. SJF can be used in specialized environments where accurate estimates of running time are available.

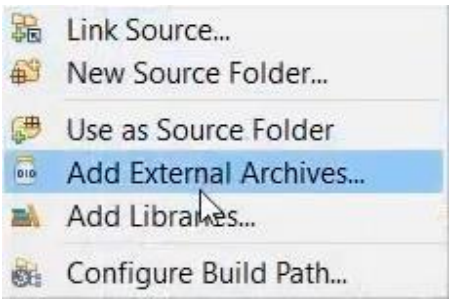
Steps

A) Installation of CloudSim and creation of simulation environment

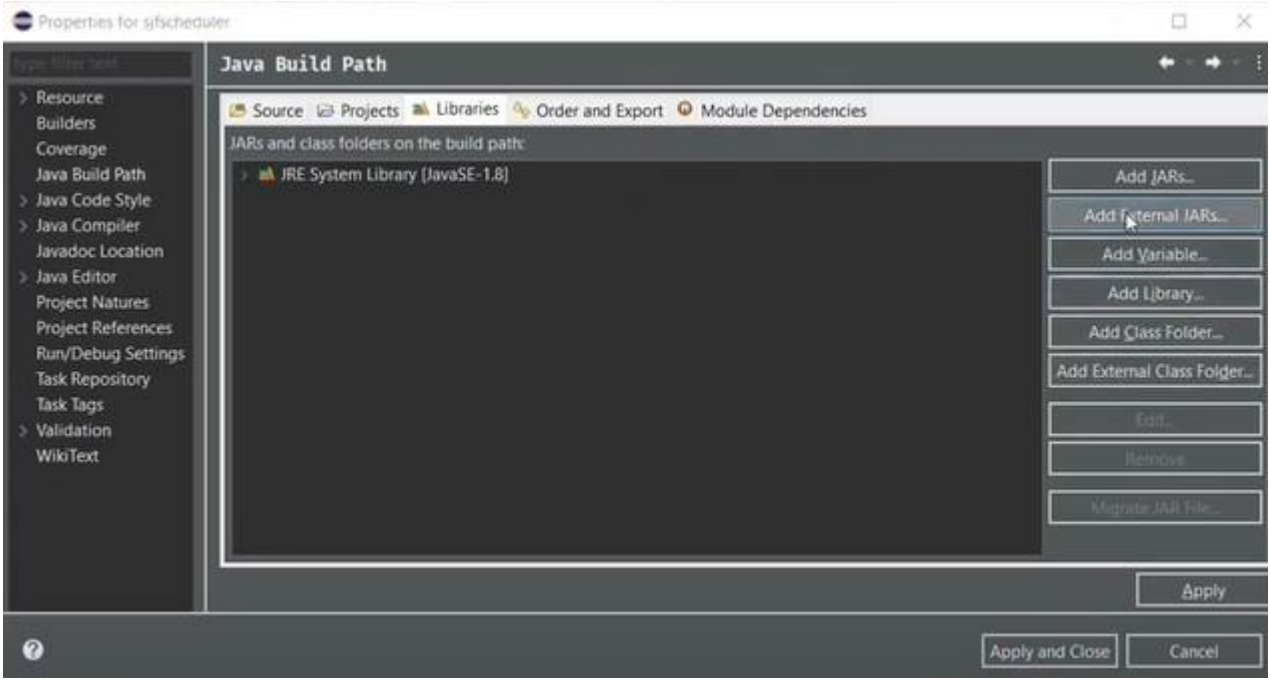
1. Visit https://github.com/Cloudslab/cloudsim/releases to download the CloudSim archives for CloudSim 4.
2. Extract the archive.
3. The jars folder of the extracted archive should contain the following files:
    a) cloudsim-4.0.jar
    b) cloudsim-examples.jar
4. Create a new Java Project using the Eclipse IDE.
5. Right click on the project root and select the Build Path option from the dropdown.



6. Select the Configure Build Path section from the extended dropdown



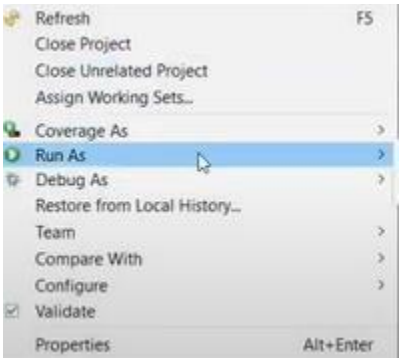
7. Select the Libraries section and click on Add External JARs field on the pop up



8. Navigate to the jars directory of the CloudSim archive and include the 2 jars in the project.
9. Create a new package in the src directory of the project.
10. Copy the code files for the constants, Data Center Creator, Data Center Broker, Matrix Generator and the SJF scheduler files from the https://github.com/suyash-more/Cloud-Computing-Projects/tree/master/Scheduling-Algorithm-in-CloudSim/src link.
11. Make sure that the package name provided in each file is the same as the previously created package.

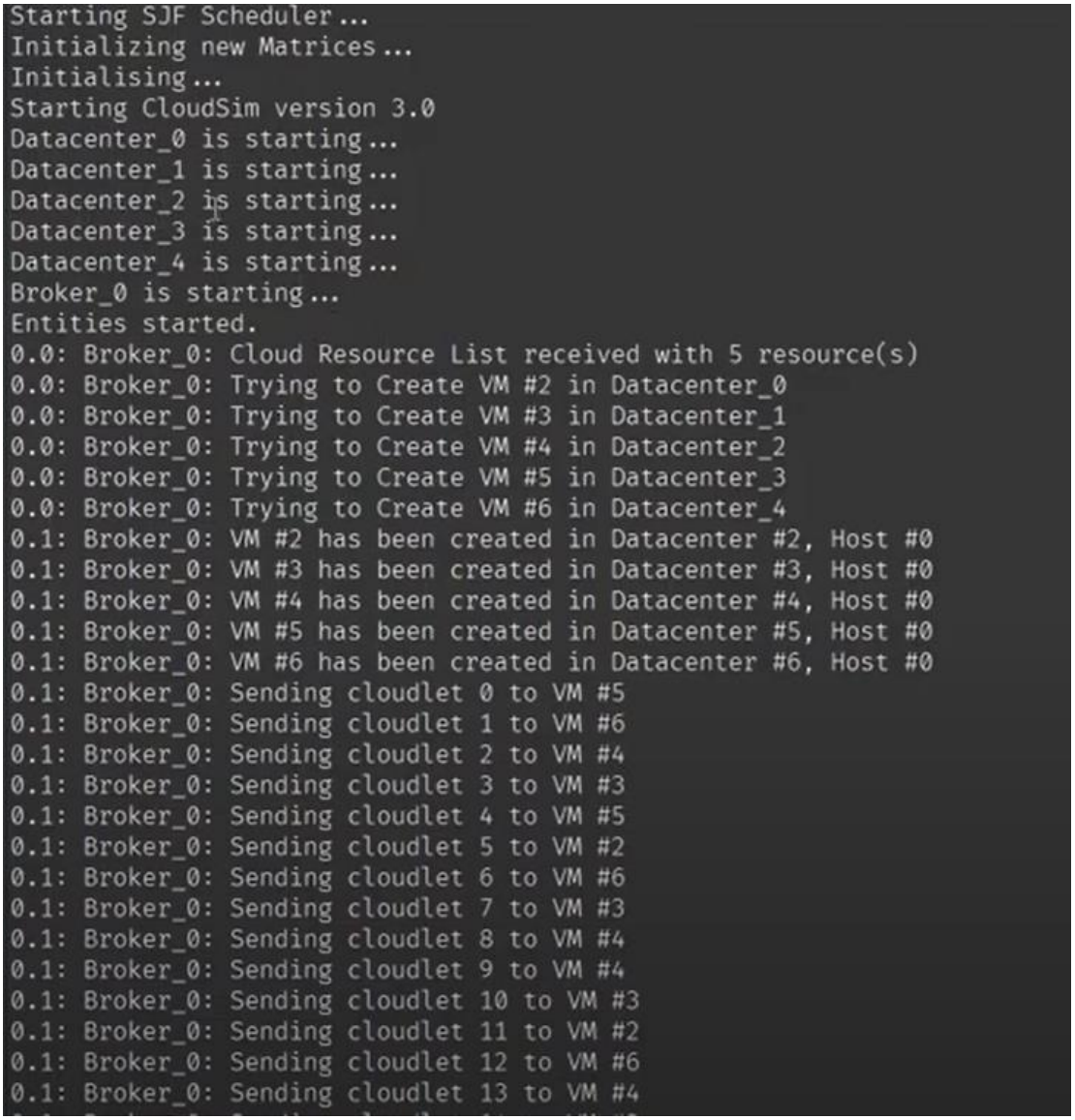
B) Execution of code

1. Right click on the project



2. Run the project as a Java Application (option in extended dropdown)

3. The result is displayed on the console


Starting SJF Scheduler...
Initializing new Matrices...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Datacenter_2 is starting...
Datacenter_3 is starting...
Datacenter_4 is starting...
Broker_0 is starting...
Entities started.
0.0: Broker_0: Cloud Resource List received with 5 resource(s)
0.0: Broker_0: Trying to Create VM #2 in Datacenter_0
0.0: Broker_0: Trying to Create VM #3 in Datacenter_1
0.0: Broker_0: Trying to Create VM #4 in Datacenter_2
0.0: Broker_0: Trying to Create VM #5 in Datacenter_3
0.0: Broker_0: Trying to Create VM #6 in Datacenter_4
0.1: Broker_0: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker_0: VM #3 has been created in Datacenter #3, Host #0
0.1: Broker_0: VM #4 has been created in Datacenter #4, Host #0
0.1: Broker_0: VM #5 has been created in Datacenter #5, Host #0
0.1: Broker_0: VM #6 has been created in Datacenter #6, Host #0
0.1: Broker_0: Sending cloudlet 0 to VM #5
0.1: Broker_0: Sending cloudlet 1 to VM #6
0.1: Broker_0: Sending cloudlet 2 to VM #4
0.1: Broker_0: Sending cloudlet 3 to VM #3
0.1: Broker_0: Sending cloudlet 4 to VM #5
0.1: Broker_0: Sending cloudlet 5 to VM #2
0.1: Broker_0: Sending cloudlet 6 to VM #6
0.1: Broker_0: Sending cloudlet 7 to VM #3
0.1: Broker_0: Sending cloudlet 8 to VM #4
0.1: Broker_0: Sending cloudlet 9 to VM #4
0.1: Broker_0: Sending cloudlet 10 to VM #3
0.1: Broker_0: Sending cloudlet 11 to VM #2
0.1: Broker_0: Sending cloudlet 12 to VM #6
0.1: Broker_0: Sending cloudlet 13 to VM #4

```
1292.724: Broker_0: Cloudlet 0 received
1907.232: Broker_0: Cloudlet 5 received
2260.772: Broker_0: Cloudlet 1 received
2784.16: Broker_0: Cloudlet 3 received
2903.4: Broker_0: Cloudlet 2 received
3065.932: Broker_0: Cloudlet 11 received
4113.036: Broker_0: Cloudlet 4 received
4485.576: Broker_0: Cloudlet 18 received
4837.776: Broker_0: Cloudlet 20 received
4956.164: Broker_0: Cloudlet 6 received
5643.272: Broker_0: Cloudlet 8 received
5807.608: Broker_0: Cloudlet 21 received
6354.656: Broker_0: Cloudlet 7 received
6905.915999999999: Broker_0: Cloudlet 23 received
7719.535999999999: Broker_0: Cloudlet 24 received
8614.368: Broker_0: Cloudlet 12 received
8752.444: Broker_0: Cloudlet 10 received
8986.24: Broker_0: Cloudlet 9 received
10703.216: Broker_0: Cloudlet 15 received
10857.967999999999: Broker_0: Cloudlet 14 received
11948.996: Broker_0: Cloudlet 25 received
13310.556: Broker_0: Cloudlet 13 received
13635.776: Broker_0: Cloudlet 16 received
15582.328: Broker_0: Cloudlet 17 received
16230.772: Broker_0: Cloudlet 19 received
17007.956: Broker_0: Cloudlet 27 received
19003.152000000002: Broker_0: Cloudlet 22 received
19533.66: Broker_0: Cloudlet 28 received
22878.644: Broker_0: Cloudlet 26 received
25918.7: Broker_0: Cloudlet 29 received
25918.7: Broker_0: All Cloudlets executed. Finishing...
```

```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time    Waiting Time
    00         SUCCESS         05           05        1292.62       00.1           1292.72            00
    05         SUCCESS         02           02        1907.13       00.1           1907.23            00
    01         SUCCESS         06           06        2260.67       00.1           2260.77            00
    03         SUCCESS         03           03        2784.06       00.1           2784.16            00
    02         SUCCESS         04           04        2903.3        00.1           2903.4             00
    11         SUCCESS         02           02        1158.7       1907.23          3065.93           1907.13
    04         SUCCESS         05           05        2820.31       1292.72          4113.04           1292.62
    18         SUCCESS         02           02        1419.64       3065.93          4485.58           3065.83
    20         SUCCESS         02           02        352.2        4485.58          4837.78           4485.48
    06         SUCCESS         06           06        2695.39       2260.77          4956.16           2260.67
    08         SUCCESS         04           04        2739.87       2903.4           5643.27           2903.3
    21         SUCCESS         05           05        1694.57       4113.04          5807.61           4112.94
    07         SUCCESS         03           03        3570.5       2784.16          6354.66           2784.06
    23         SUCCESS         02           02        2068.14       4837.78          6905.92           4837.68
    24         SUCCESS         02           02        813.62       6905.92          7719.54           6905.82
    12         SUCCESS         06           06        3658.2       4956.16          8614.37           4956.06
    10         SUCCESS         03           03        2397.79       6354.66          8752.44           6354.56
    09         SUCCESS         04           04        3342.97       5643.27          8986.24           5643.17
    15         SUCCESS         06           06        2088.85       8614.37          10703.22           8614.27
    14         SUCCESS         03           03        2105.52       8752.44          10857.97           8752.34
    25         SUCCESS         03           03        1091.03       10857.97           11949           10857.87
    13         SUCCESS         04           04        4324.32       8986.24          13310.56           8986.14
    16         SUCCESS         06           06        2932.56       10703.22          13635.78           10703.12
    17         SUCCESS         06           06        1946.55       13635.78          15582.33           13635.68
    19         SUCCESS         04           04        2920.22       13310.56          16230.77           13310.46
    27         SUCCESS         06           06        1425.63       15582.33          17007.96           15582.23
    22         SUCCESS         04           04        2772.38       16230.77          19003.15           16230.67
    28         SUCCESS         06           06        2525.7       17007.96          19533.66           17007.86
    26         SUCCESS         04           04        3875.49       19003.15          22878.64           19003.05
    29         SUCCESS         04           04        3040.06       22878.64          25918.7            22878.54
Makespan using SJF: 4396.012266367984
```

Cloud Computing Lab

Experiment-6

Find a procedure to transfer the files from one virtual machine to another virtual machine.

Objective:

1. To learn about Virtual machines(VM).
2. To learn the Basics of the File transfer mechanism.
3. To transfer the files from one VM to another VM.

Requirements:

1. Two Virtual machines installed. (For example, we have Ubuntu 21 and Kali Linux for this assignment)

Theory:

What is a virtual machine?

A virtual machine is an application providing a platform-independent programming runtime that allows applications to execute in the same manner on different platforms. The virtual machine acts as a bridge to the real environment, hiding the details of the operating system. Do not confuse this term with system virtual machines, such as VMware, Virtual server, Xen which enables one to run multiple OS on a single piece of hardware.

Types of virtualization techniques:

1. Guest Operating system virtualization
2. Shared Kernel Virtualization
3. Kernel Level Virtualization
4. Hypervisor Virtualization

Types of virtual machines:

We can classify virtual machines into two types:

System Virtual Machine:

These types of virtual machines give us complete system platform and gives the execution of the complete virtual operating system. Just like virtual box, system virtual machine is providing an environment for an OS to be installed completely. We can see in below image that our hardware of Real Machine is being distributed between two simulated operating systems by Virtual machine monitor. And then some programs, processes are going on in that distributed hardware of simulated machines separately.

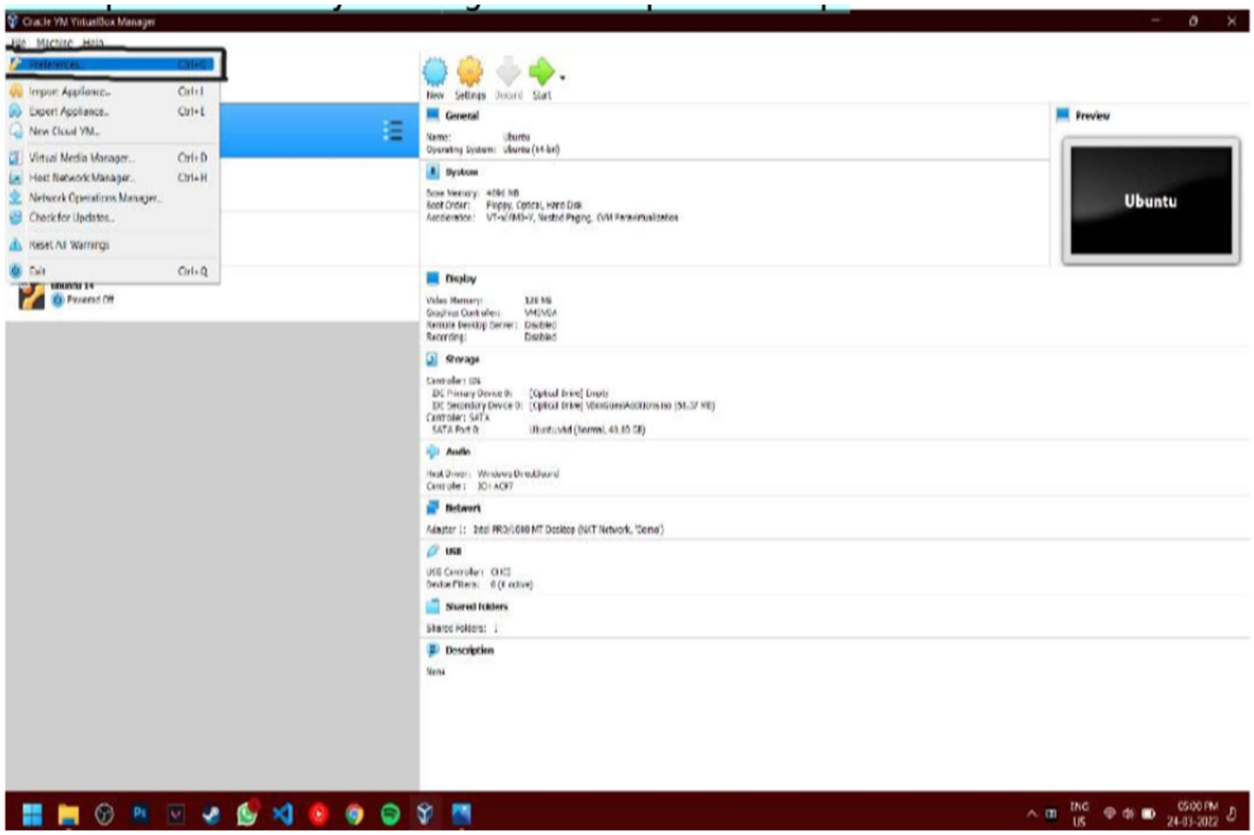
Process Virtual Machine:

While process virtual machines, unlike system virtual machines, do not provide us with the facility to install the virtual operating system completely. Rather it creates a virtual environment of that OS while using some app or program and this environment will be destroyed as soon as we exit from that app. Like in the below image, there are some apps running on the main OS as well some virtual machines are created to run other apps. This shows that as those programs required different OS, process virtual machine provided them with that for the time being those programs are running.

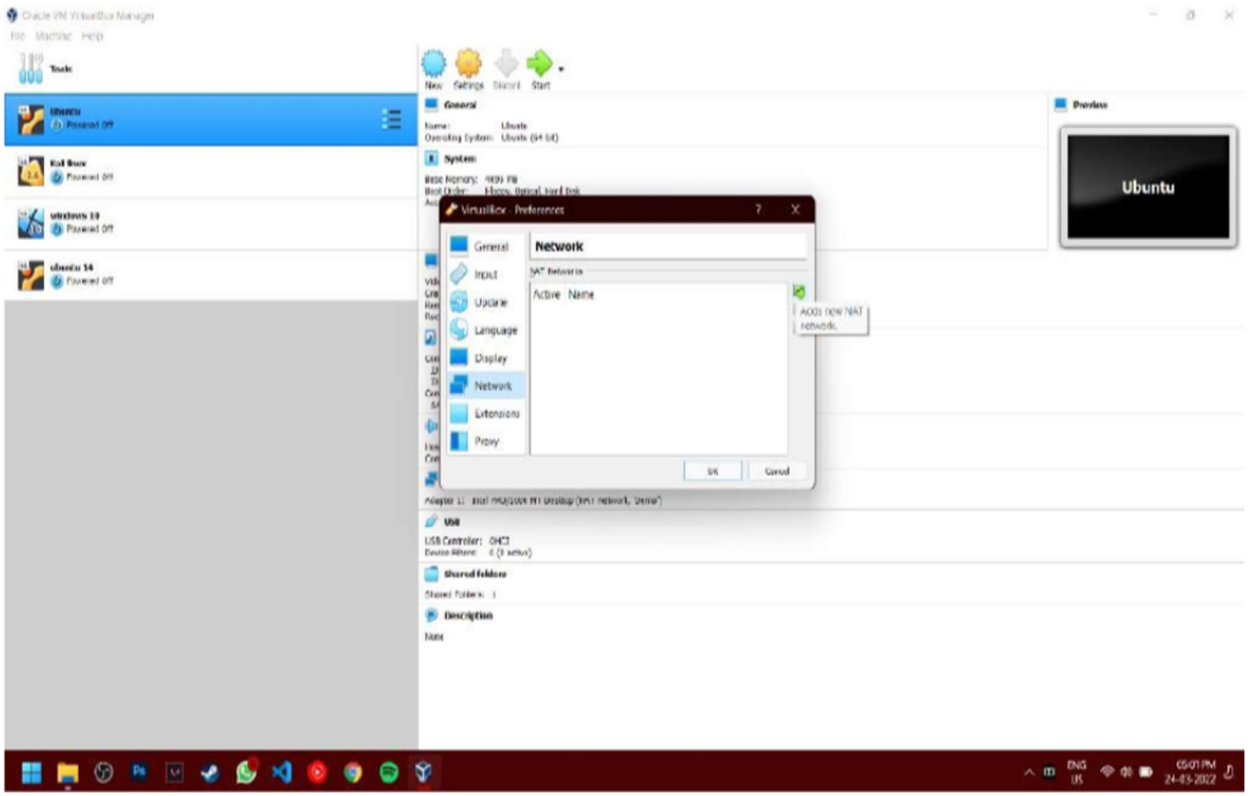
Steps:

Step 1. Create a Nat network in which 2 virtual machines can communicate.

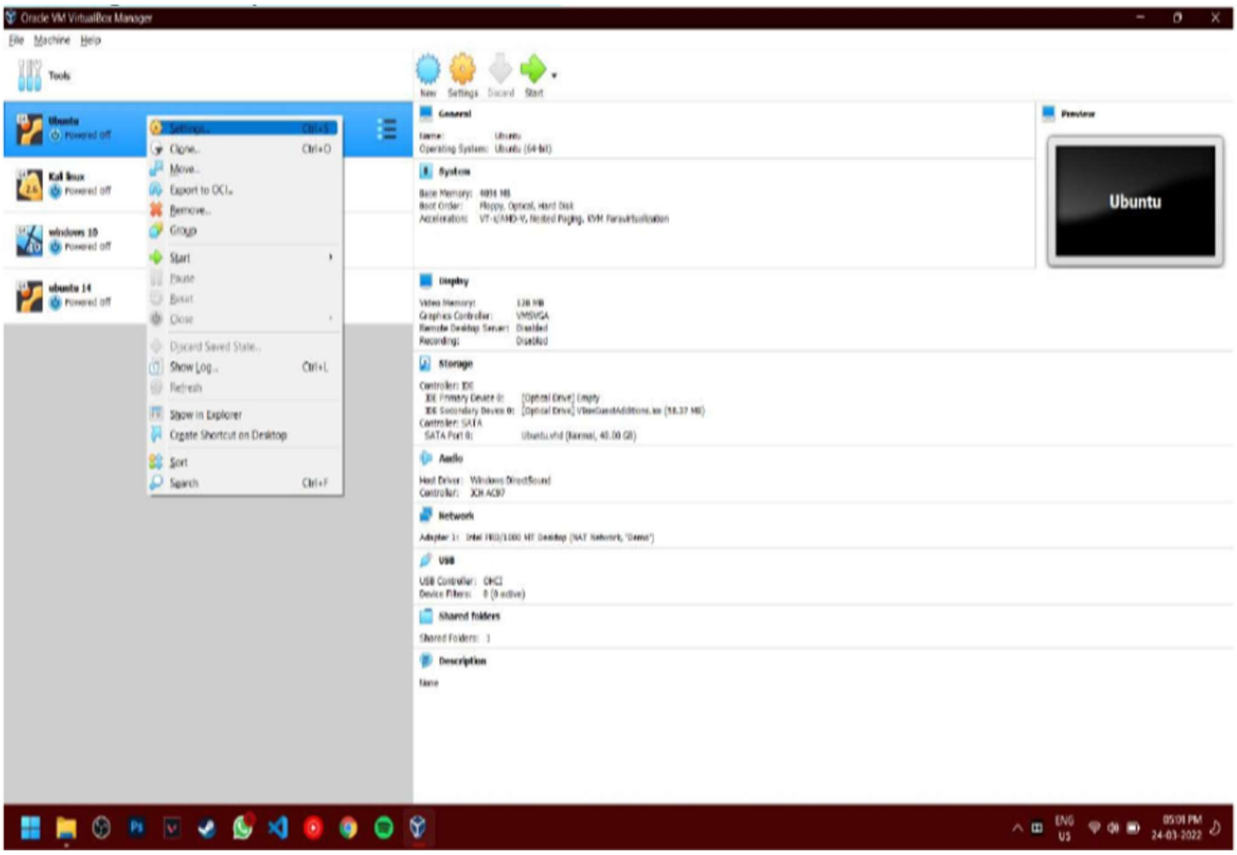
Go to preferences by clicking the File option in Top.



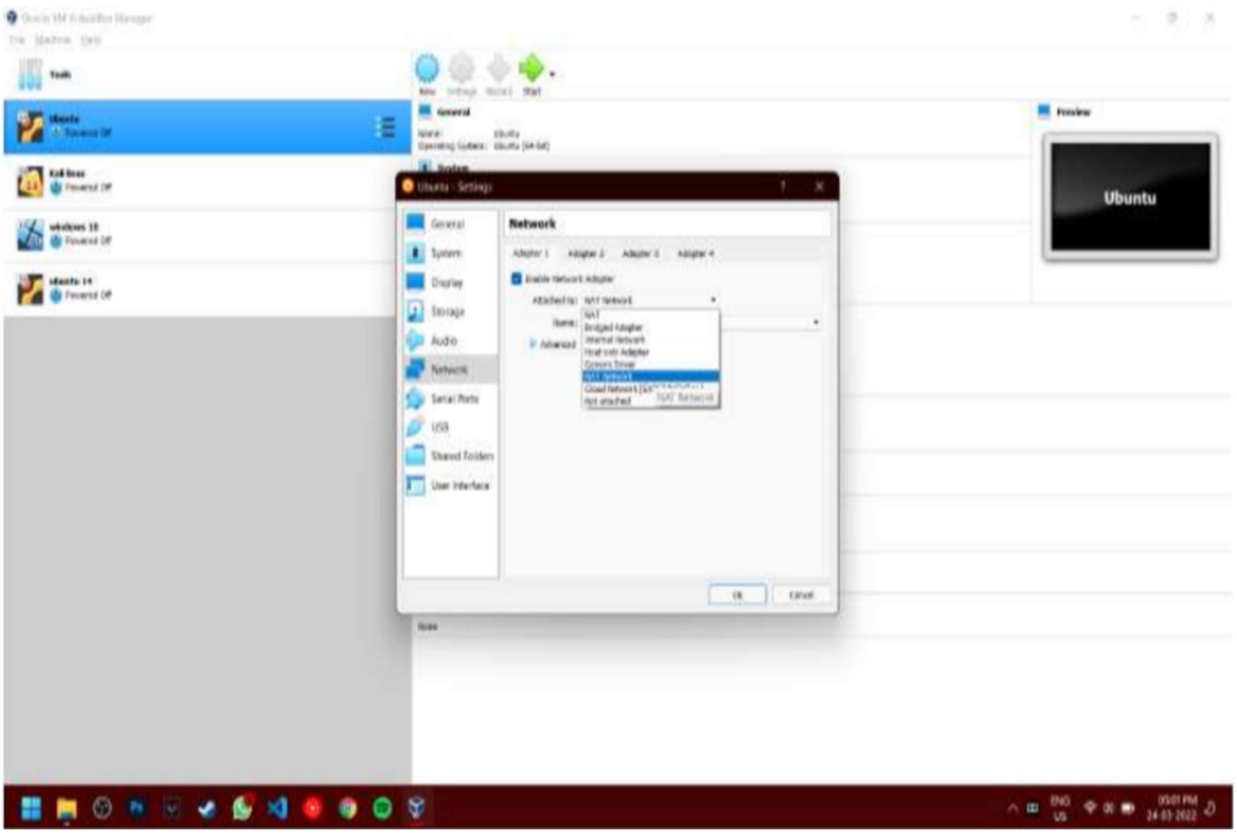
Now select the network option and create a new NAT network

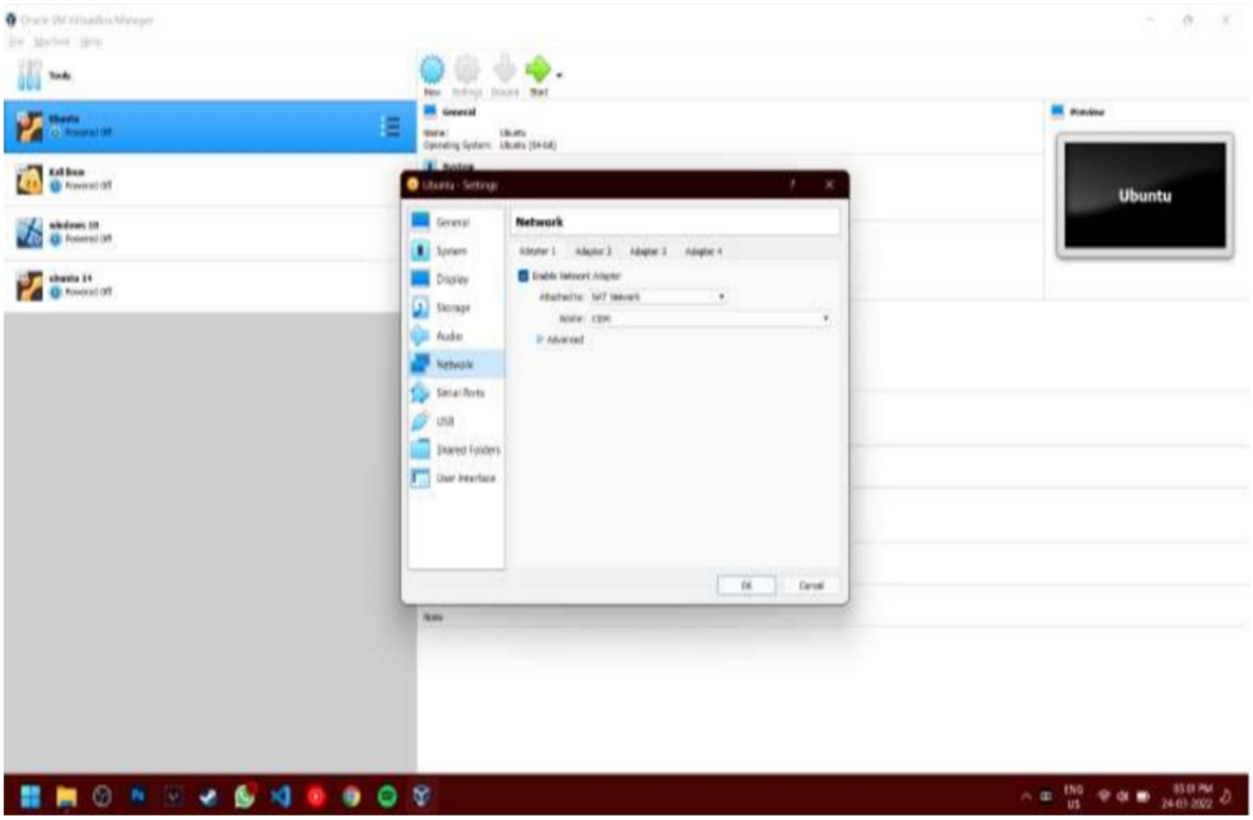


After creating a NAT network, now go to virtual machine setting by right clicking on the preferred machine.



Now go to Network and change the attached option to “NAT network” and select the network we created earlier.

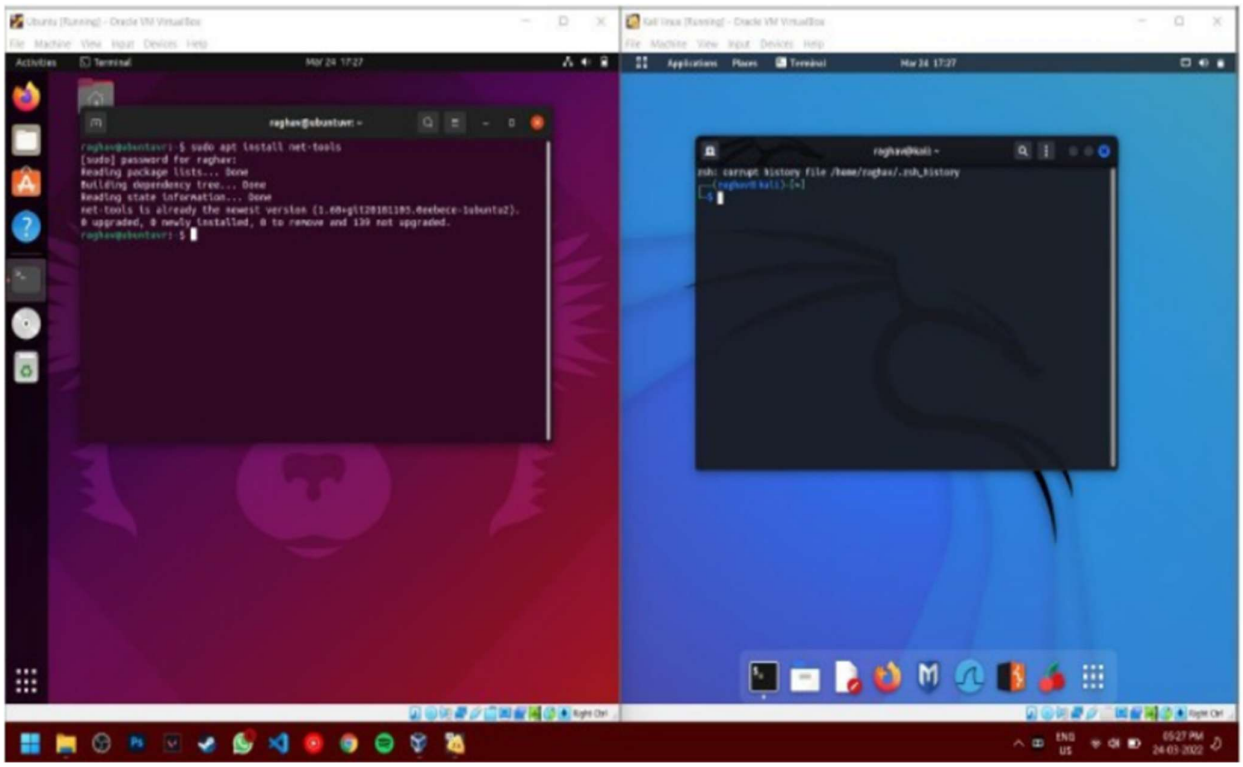




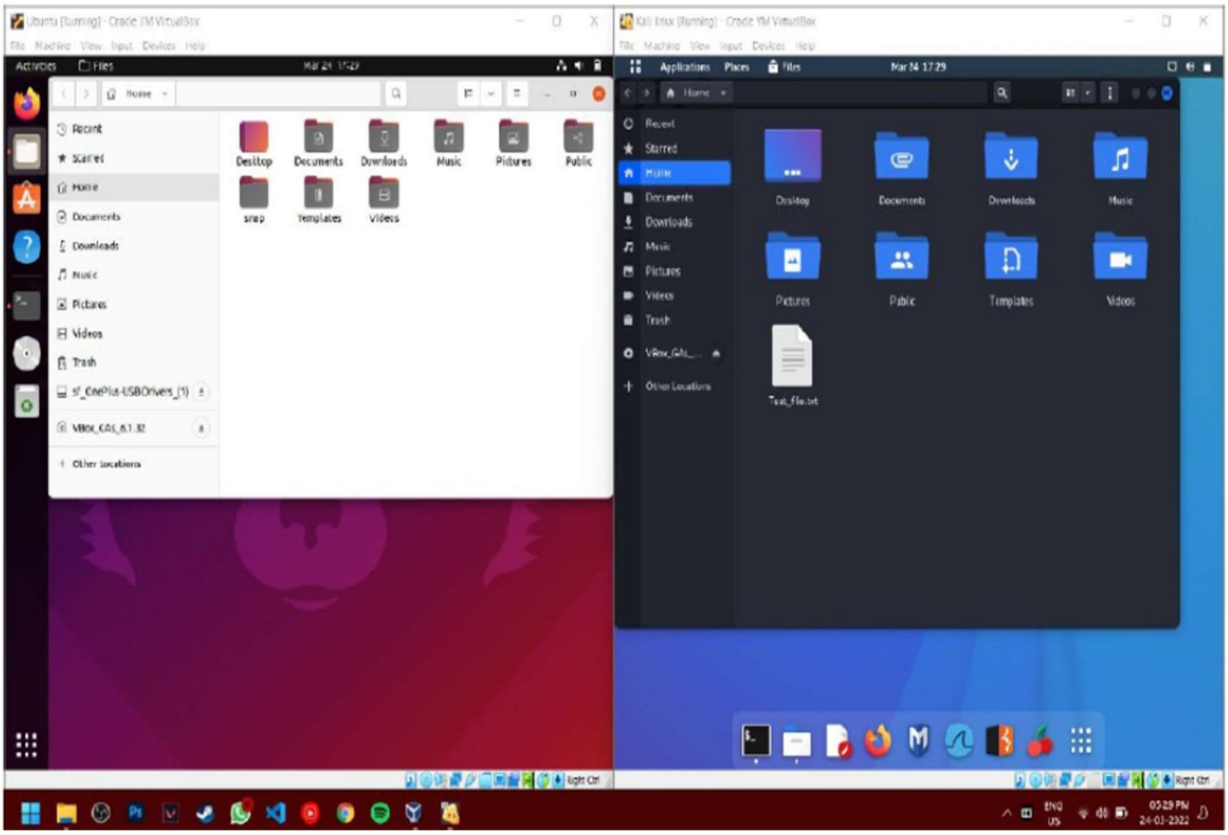
Now repeat the same process for another machine

Step 2. Launch both virtual machines

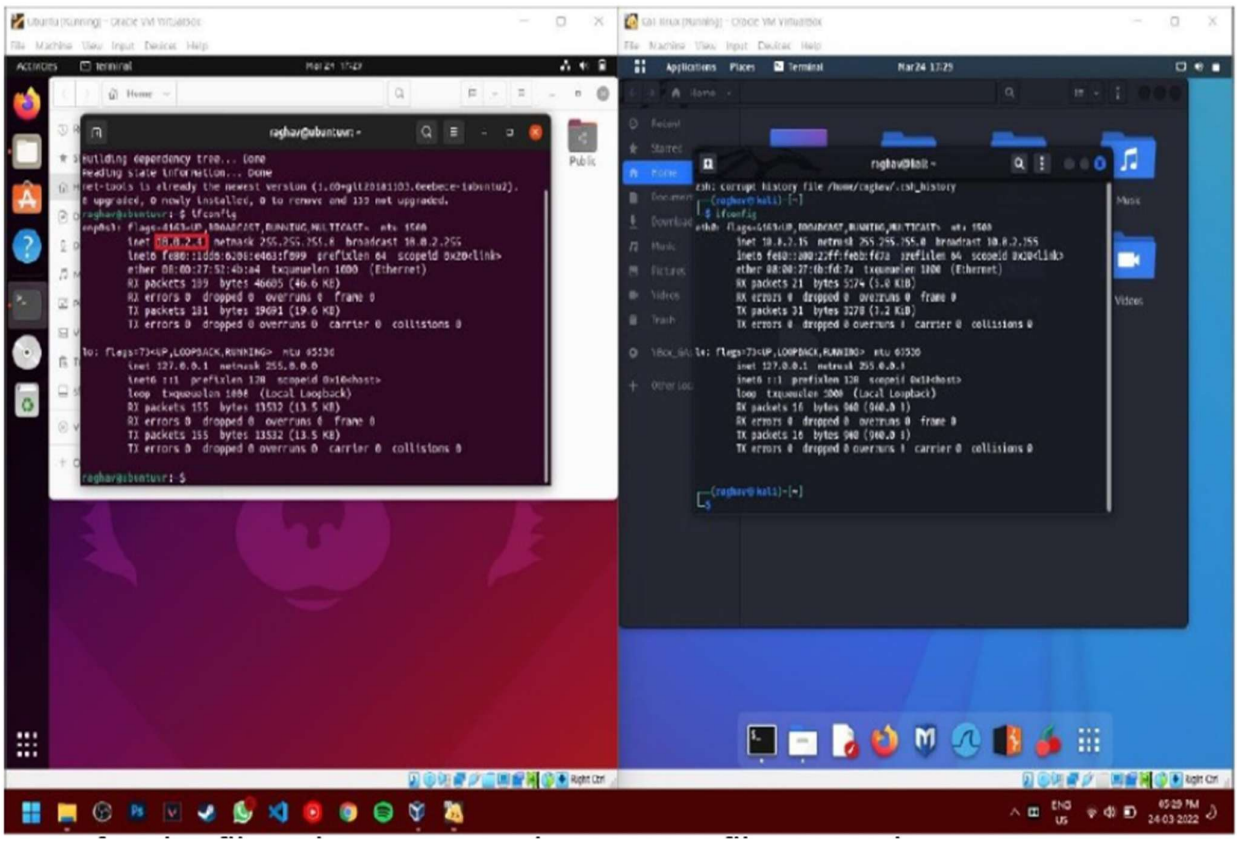
Now install “Net-tools” on both machine which will help to identify i/p address of the machine. Command: sudo apt install net-tools



Now create a file in Home folder using any text editor. Here we have used “Test_file.txt” and will transfer from Kali Linux (Right Machine) to Ubuntu (left Machine)

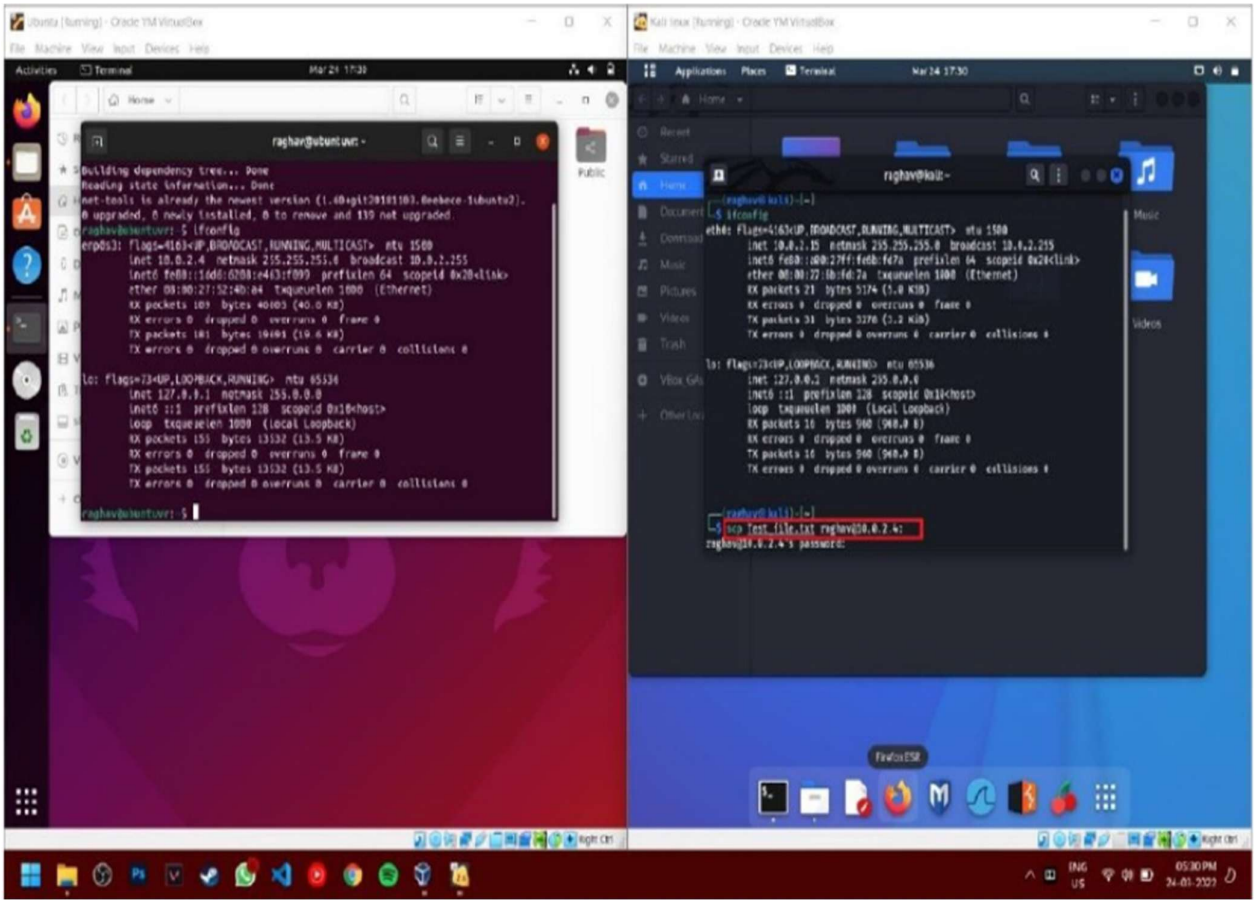


Now we can check i/p address of Ubuntu where we want to transfer the file using " ifconfig " command. Here Ubuntu has i/p address 10.0.2.4.



Transfer the file using command- scp Test_file.txt raghav@10.0.2.4: Where Test_file.txt is our file raghav is the username of the Ubuntu 10.0.2.4 is ip address of Ubuntu (left machine)

Optional: if scp is not installed then install by using the command: sudo apt install openssh-server



Now enter the password for Ubuntu(left machine) admin, after enter the password, the file will be sent from Kali Linux(Right machine) to Ubuntu(Left machine)

Cloud Computing Lab

Experiment-7

Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

TryStack was once an OpenStack-powered online demo platform for testing OpenStack's features, such as launching virtual machines (VMs). However, TryStack has been discontinued. Today, the best way to try OpenStack without a full installation is to use either **devstack** (for local installation on a single machine) or public cloud providers that offer OpenStack-based services.

OpenStack is an open-source software cloud computing platform. OpenStack is primarily

used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS).

TryStack is the easiest and free way to do it.

Minimum requirements for OpenStack is listed below:

4 GB Of Ram.

4 CPU Units.

30 GB Disk Space.

Step 1: Prepare the environment for installing OpenStack:

Run the following commands and you will be done with it to bring all your packages to

the latest version and install git so that we can clone OpenStack to our Linux machine.

Commands :

```
sudo apt-get update
sudo apt-get upgrade
sudo  apt-get  dist-
upgrade  sudo  apt-get
install git -y sudo reboot
```

Step 2: Download and Install OpenStack!

Note: If you already have a "stack" user on your virtual machine or laptop (with sudo privileges), then you do not need to create an additional user.

After your virtual machine is done with a reboot, you are now ready to install OpenStack.

Normally OpenStack runs under non-root user with sudo privileges. We can easily create one to start with using:

Create the user named as "stack"

```
Sudo useradd -s /bin/bash -d /opt/stack -m stack
```

Now let us give this user sudo privileges using:

```
echo"stack ALL=(ALL) NOPASSWD: ALL"|sudotee/etc/sudoers.d/stack
```

We now have to log in as user “stack” to proceed with our installation as

```
sudosu–stack
```

Start with the installation of openstack by downloading the required material.

git clone https://git.openstack.org/openstack-dev/devstack

```
# cd to the cloned directory
cd devstack
```

Normally during installing it will ask you to set various passwords, you can automate this process by creating a file in your current directory named “local.conf”. Save and exit the following file, this will automate the installation process.

```
# create the file
$nanolocal.conf
```

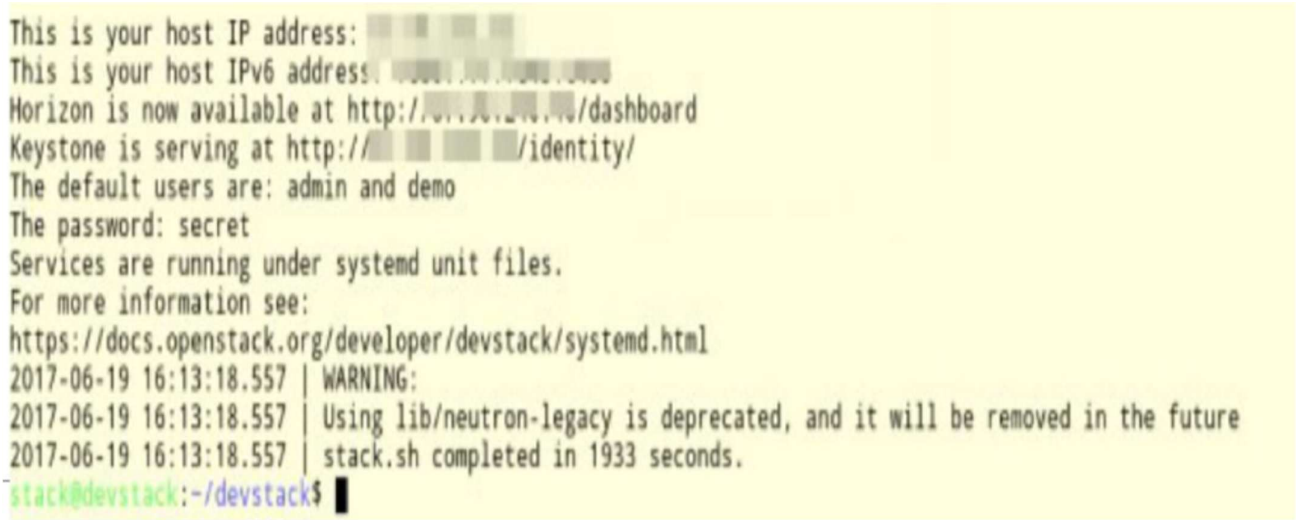
Now paste following contents in the file [[local|localrc]]

```
ADMIN_PASSWORD=secret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

We are now ready to run the installation script. Installation script can be launched using the command:

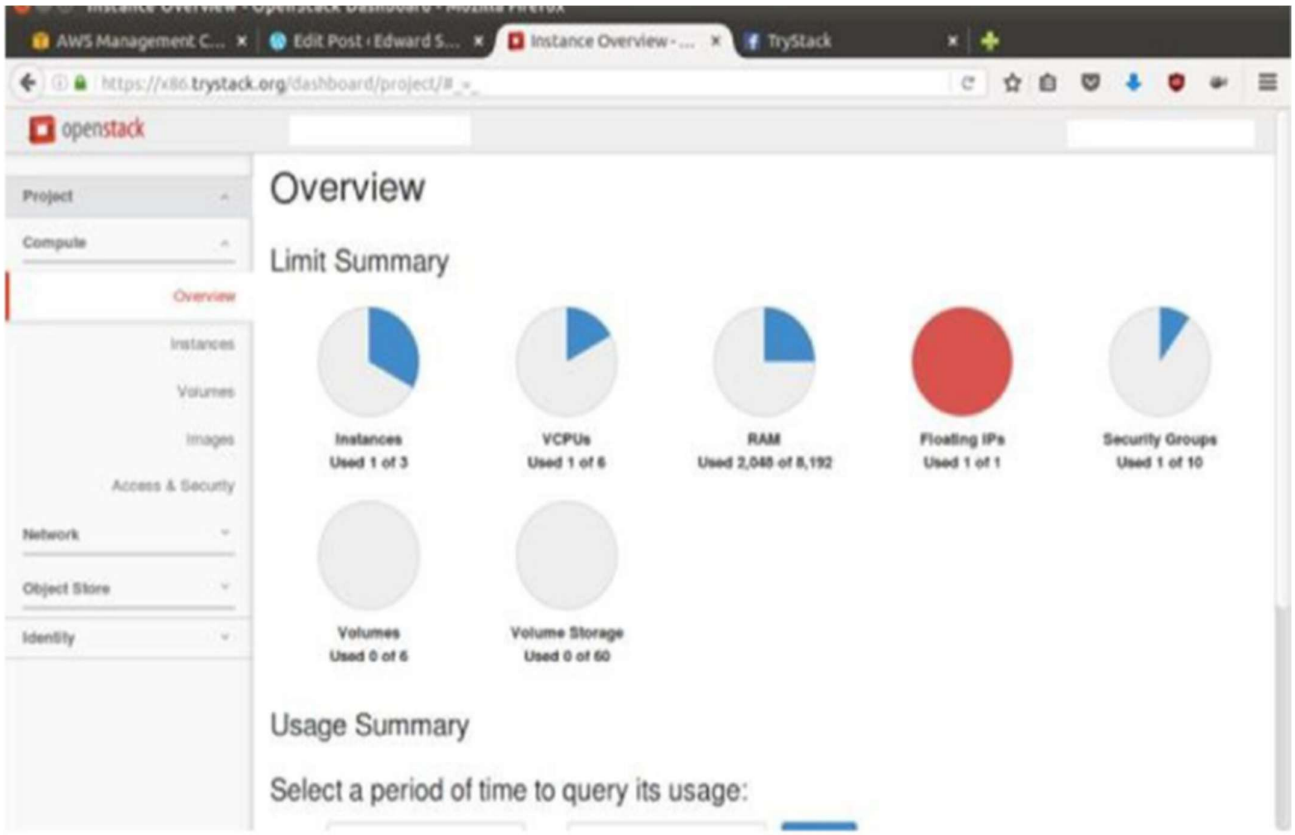
```
./stack.sh
```

Once installation is complete your screen should be :



OpenStack dashboard can now be accessed at :

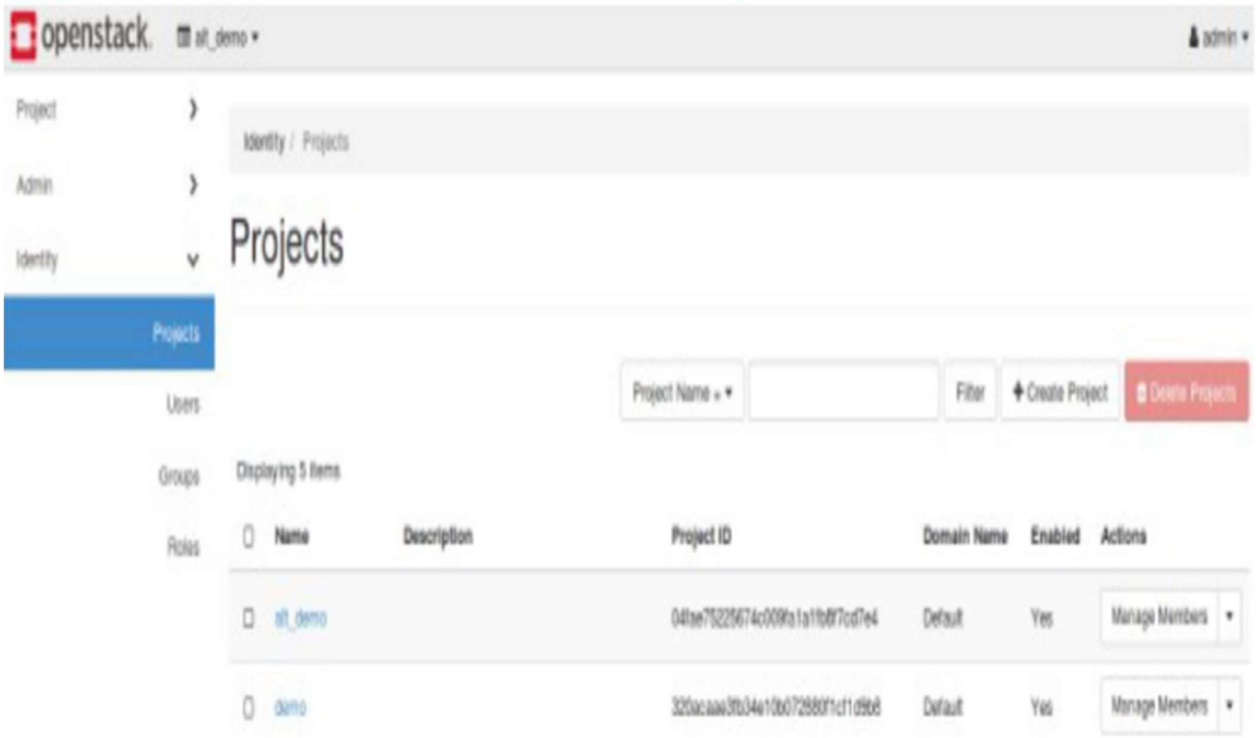
```
http://<IPAddress>/dashboard/
```



Step 3: Create Network!

Note: Please remember that without having a network, you can not launch an instance/virtual machine.

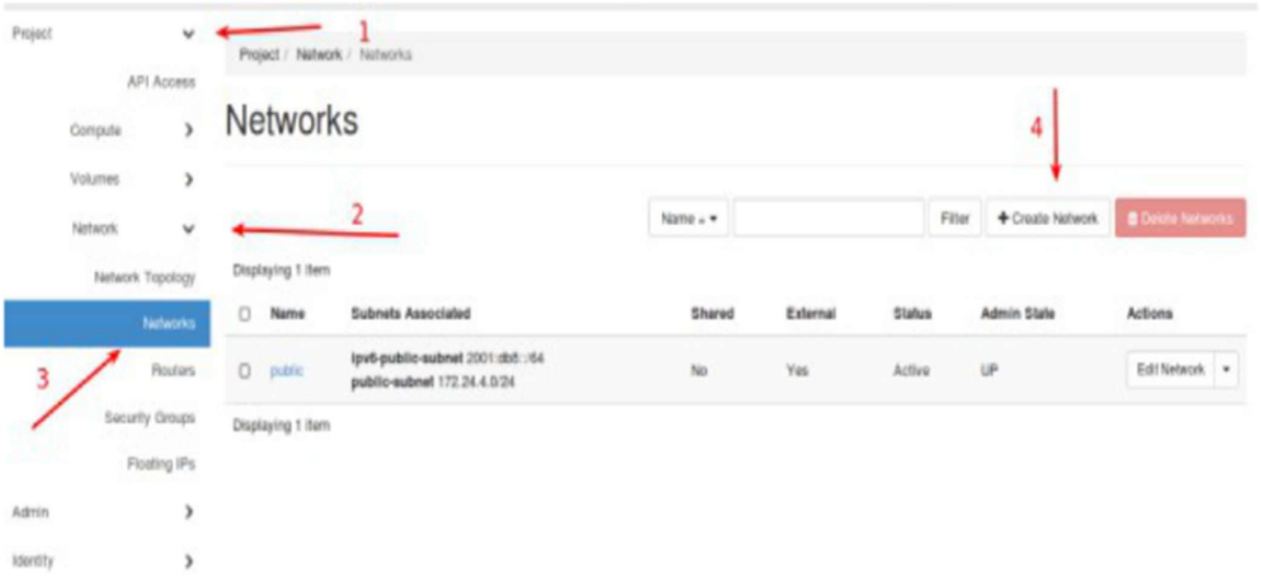
After you are logged into the OpenStack Dashboard it will look something like this:

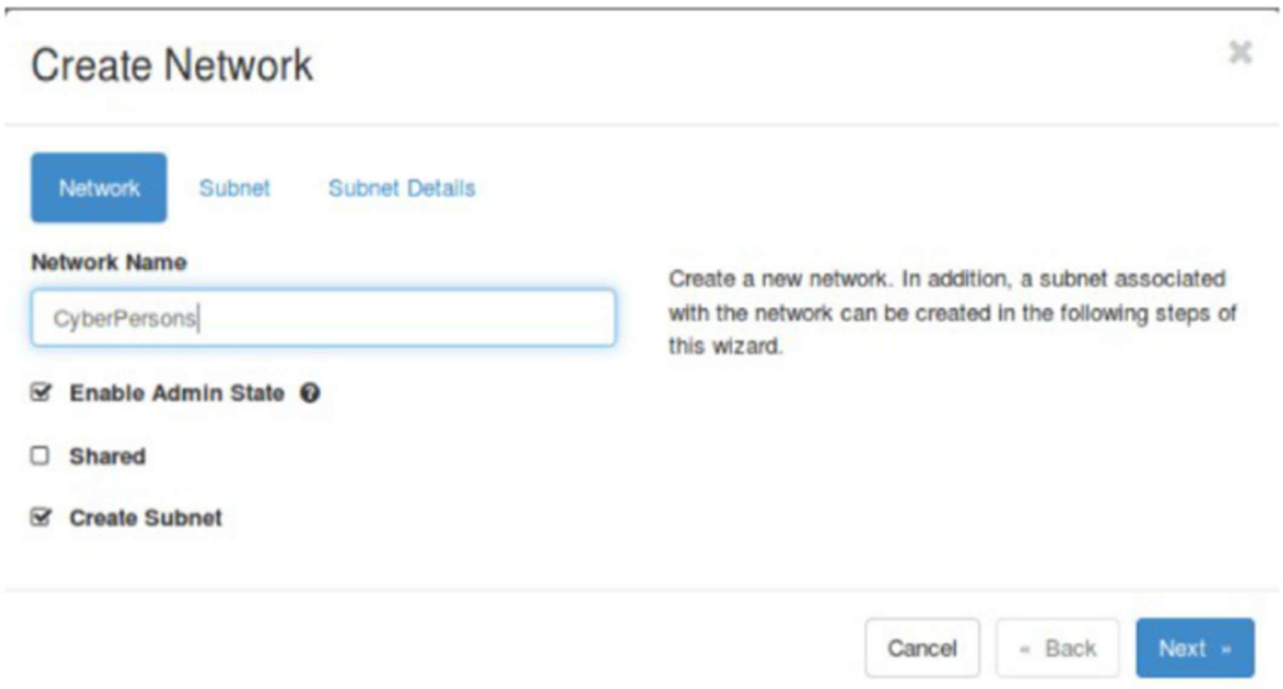


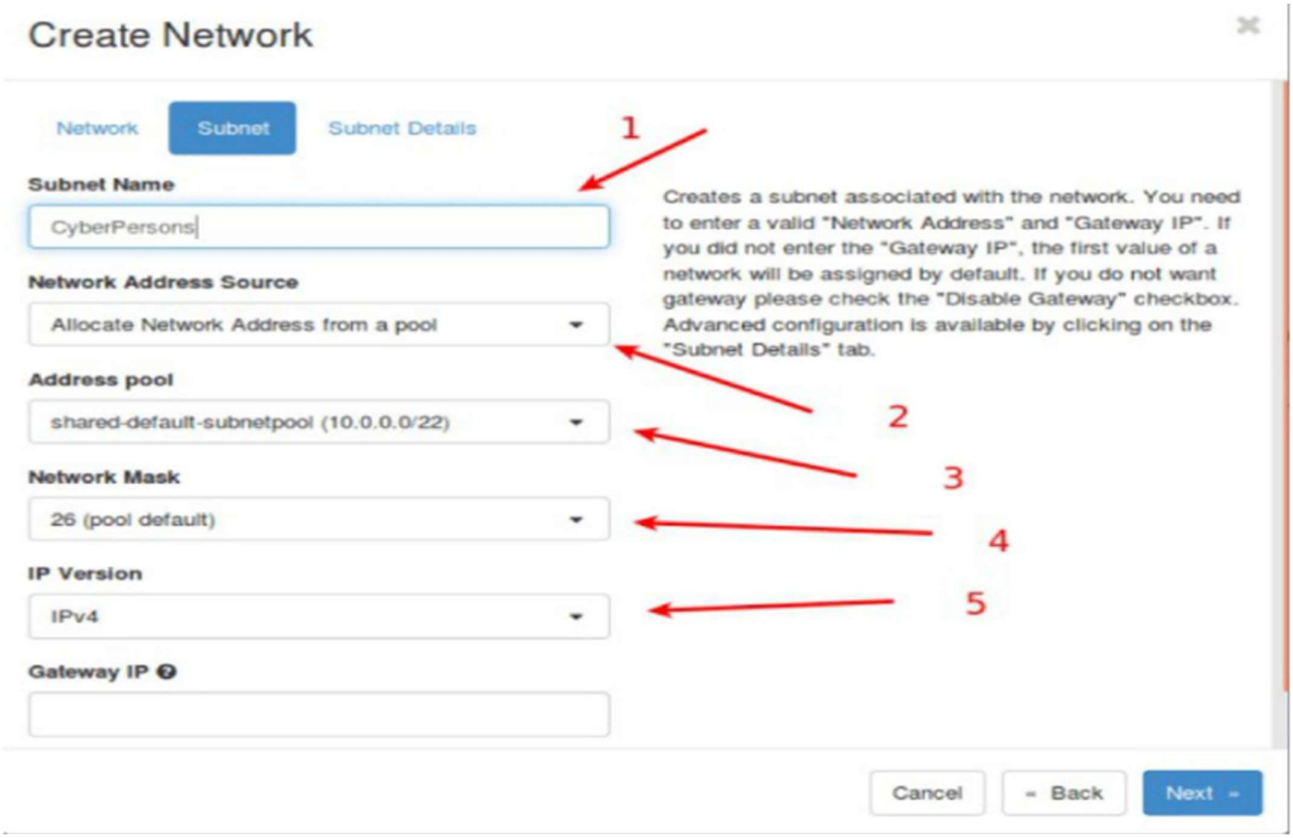
We need to create a network that virtual machine can use. For now, it will just be a dummy network:

Steps:

1. Click the Project Drop Down.
2. Click the Network Drop Down.
3. From network Drop Down select Networks, and this window will open that you see on the right side.
4. Finally, click Create Network

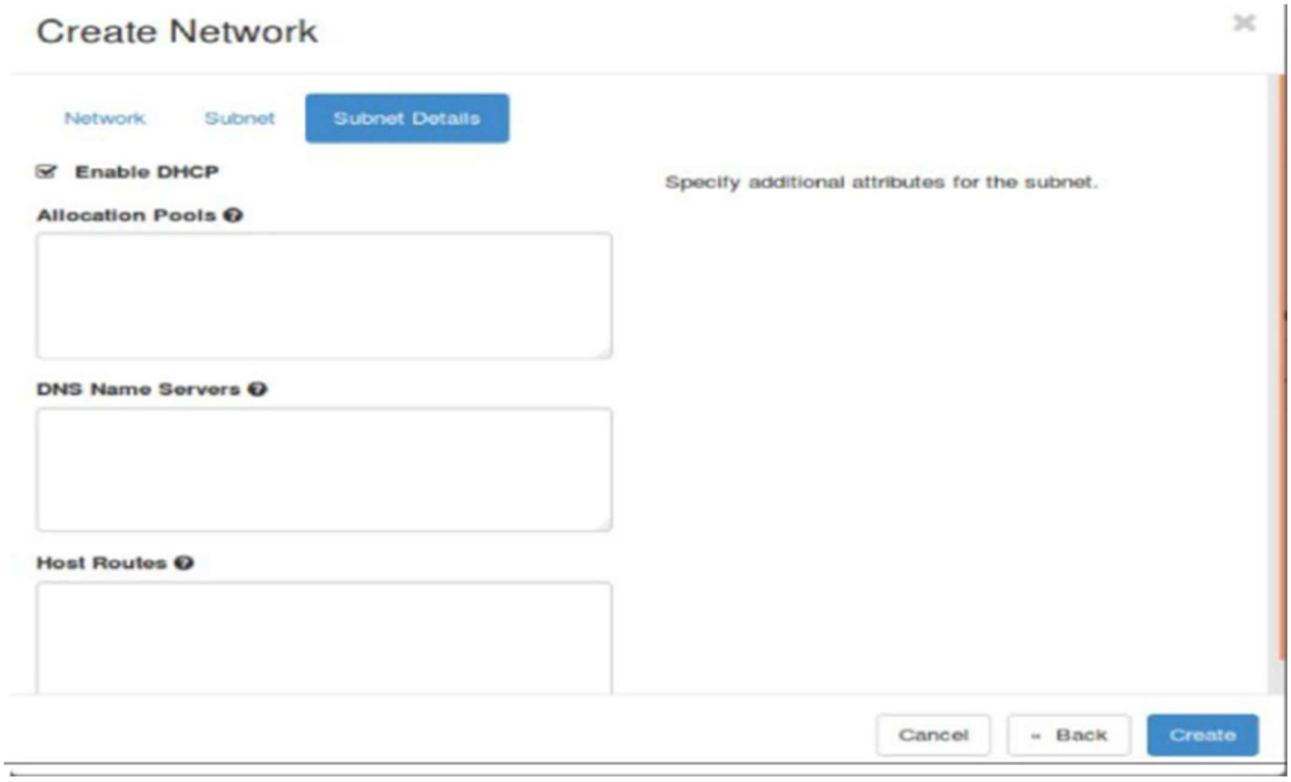






Once all these things are done, click “Next”.

Now everything is optional in this below window if you are interested in filling something up, you can. Otherwise, leave everything as it is and click “Create”. You now have a network that you can use to launch a virtual machine.

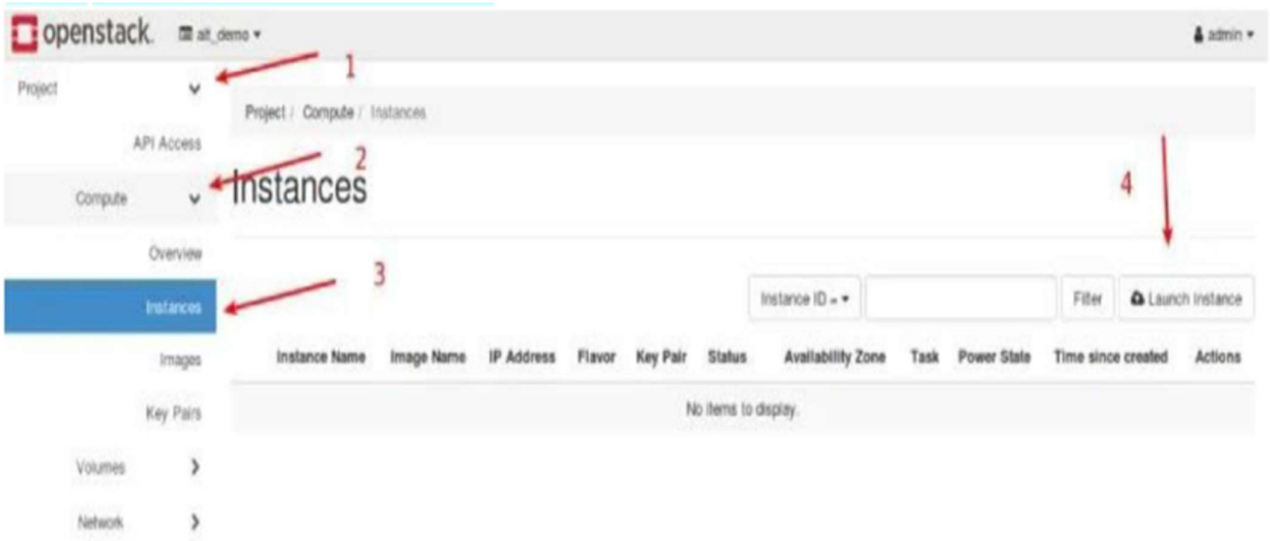


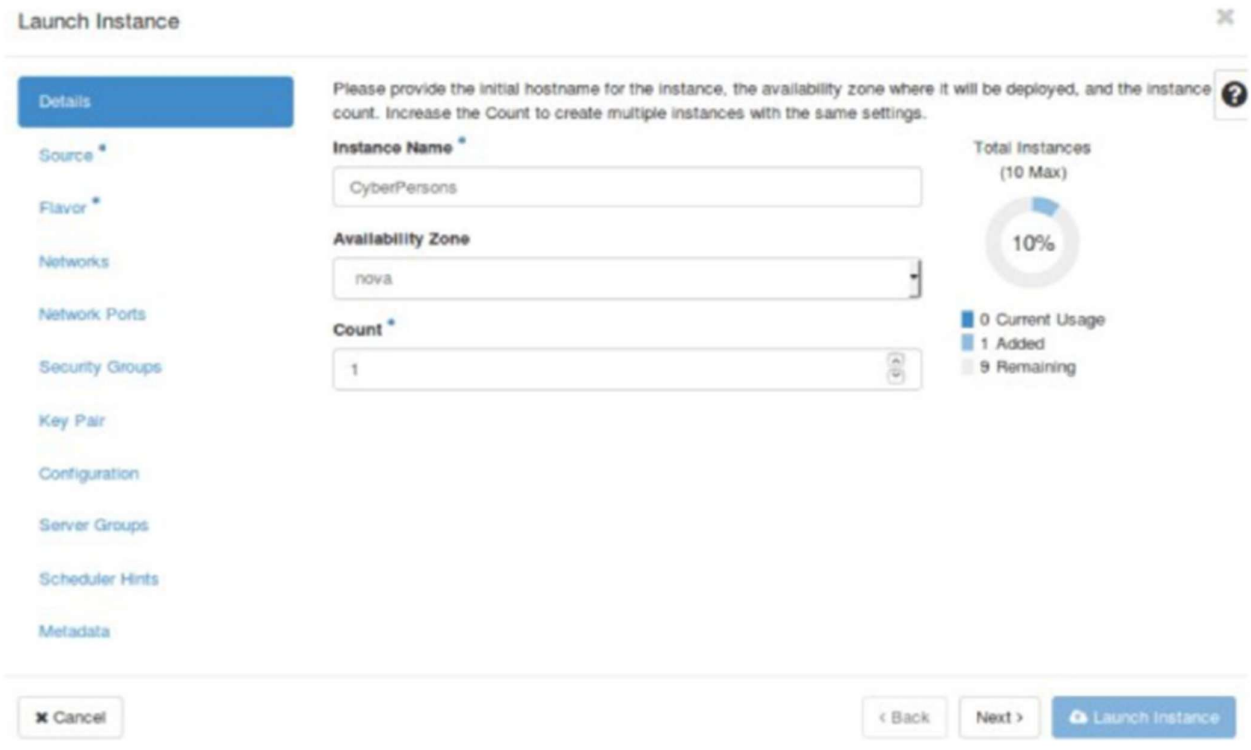
Step 4: Create Virtual Machine/Instance

After the network is created, we are now ready to create our very first virtual machine.

Steps:

1. Click on “Project” drop down.
2. Inside project click “Compute” drop down.
3. Under compute you have four options, since we are interested in creating an instance, you have to click on “Instance”.
4. Finally, click “Lunch Instance”.





Now, there are 11 tabs to create an instance, we will go through each tab one by one.

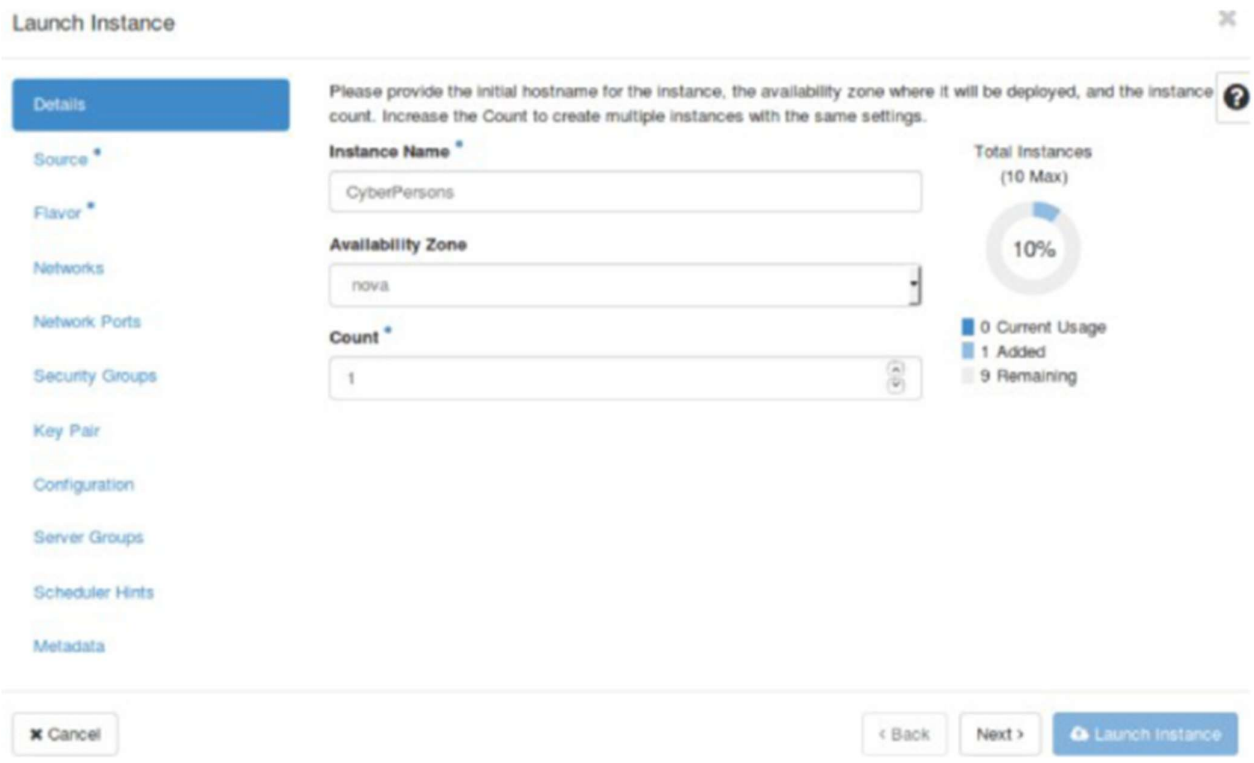
Details Tab :

This is a general information tab for creating an instance,

You will have to assign a name to your virtual machine on this tab.

Select zone to launch a virtual machine, and

Tell how many copies of virtual machine you want.



Source Tab

Normally when we create a virtual machine on Proxmox or VMWare we need to insert CD-ROM

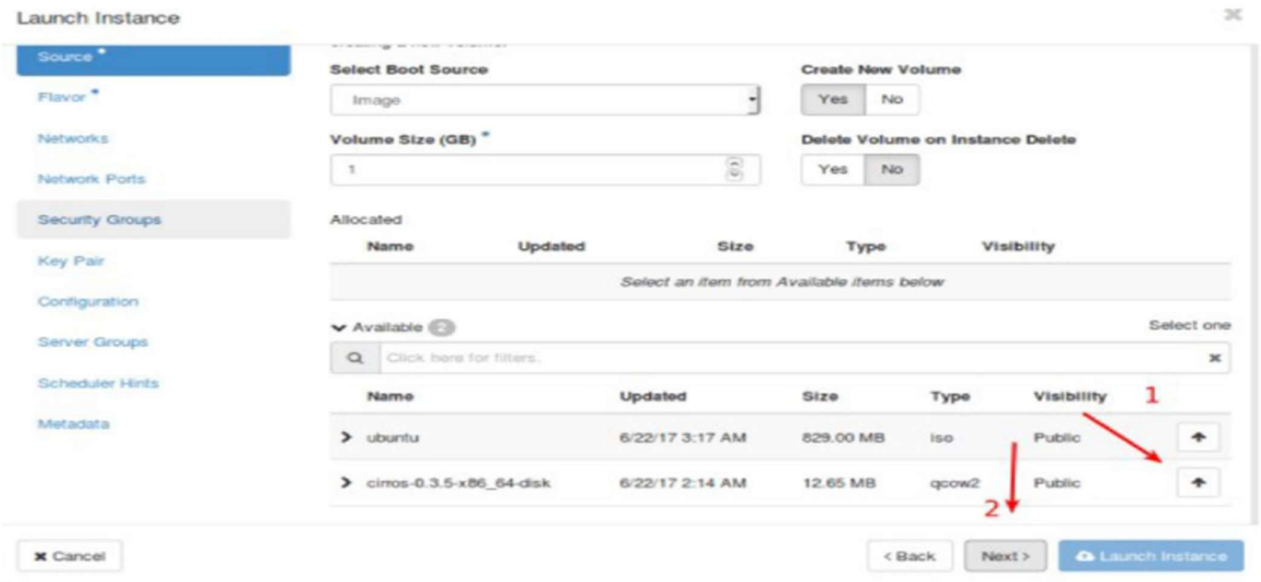
In OpenStack this is done by Source Tab, you can use various ways to launch a new virtual machine, OpenStack allows you to choose following as a source to create your instance.

Image

Snapshot of already created instance

Volume or a volume Snapshot

We are going to use “Cirros” image to create our instance.



1. Click on the icon where the first arrow is pointing, so that we can use “Cirros” to launch our virtual machine.
2. After the image is selected, just click “Next” so that we can move to “Flavor” tab.

Flavor Tab

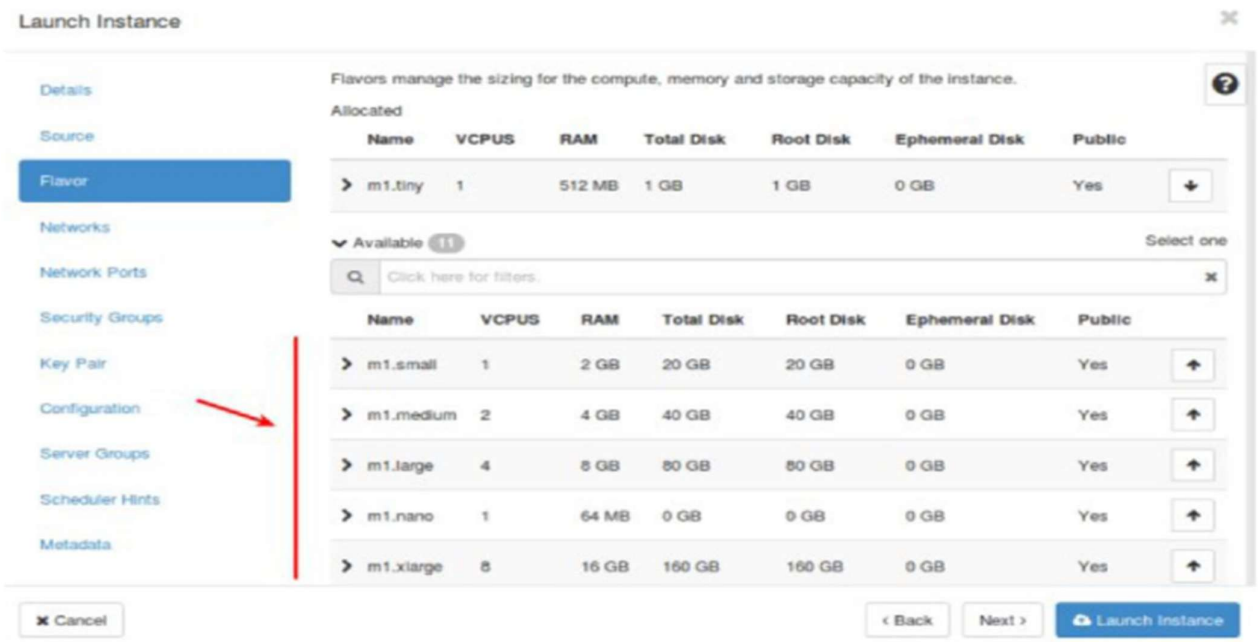
Flavor tab will allow you to allocate resource to your instance. Like:

Ram.

CPU.

Disk Space.

It is similar to giving virtual resources to the virtual machine, but OpenStack gives fancy names to everything



You can see that there are 11 available pre-configured templates to choose from. The one I choose gave following resources to the instance:

1 virtual CPU.

512 MB Ram.

1 GB Disk.

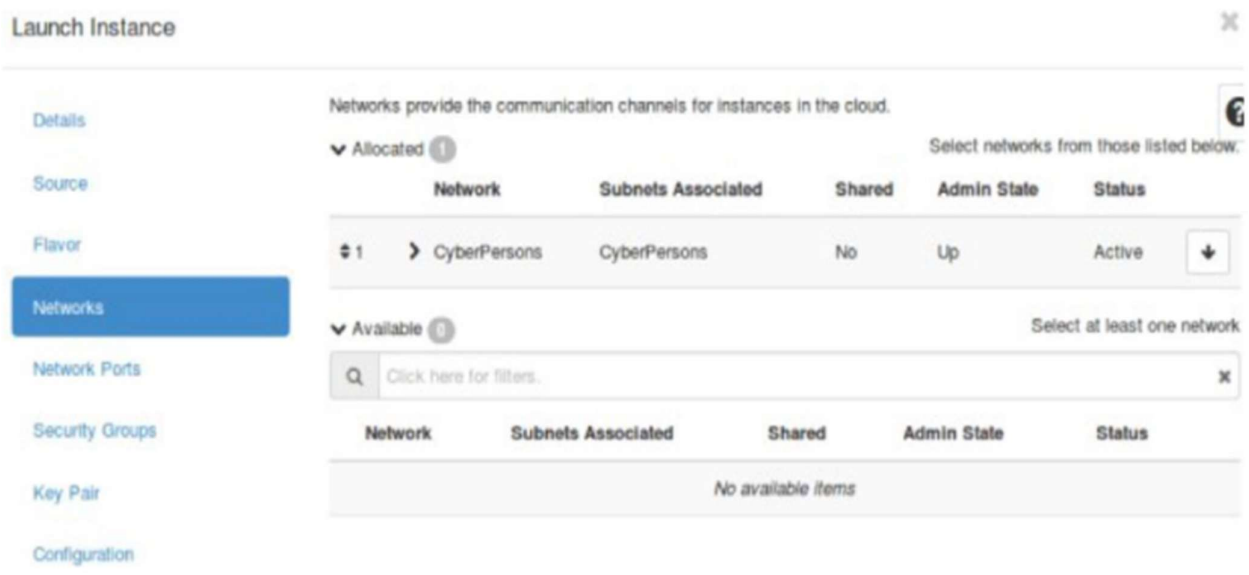
After flavor is selected, just press “Next”.

Network Tab

Network tab allows us to define a network for our virtual machine, you might have remembered that we’ve created a network above for this purpose.

Now by default, the network you have created above will be selected for this machine,

as seen in the image below:



Don’t change anything just click “Next”.

Network Ports Tab

For now, just leave the default settings on “Network Ports” tab and click next.

Security Groups Tab

Security groups define how a specific virtual machine is allowed to talk with the outer world. As for now, we are just trying to create our first virtual machine, you can leave all the defaults.

Key-Pair Tab

Leave defaults and click Next.

Configuration Tab

Leave defaults and click Next.

Server Groups Tab

Leave defaults and click Next.

Scheduler Hints Tab

Leave defaults and click Next.

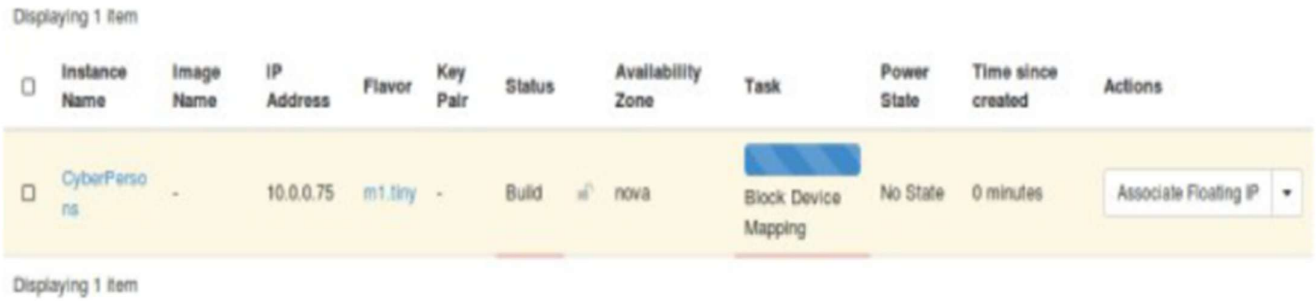
Metadata Tab

Leave defaults and click Next.

Launch Instance

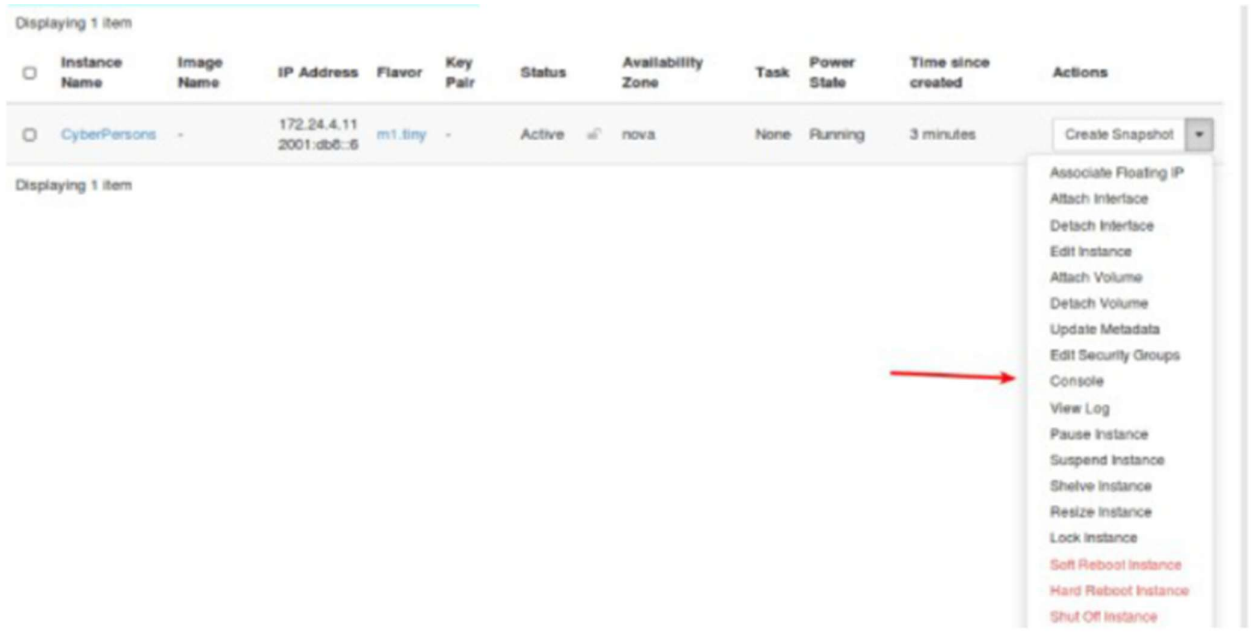
After going through all the tabs, you are now ready to press that magic “Launch Instance” button.

Once you click “Launch Instance” button, OpenStack will start creating our virtual machine, and it is going to look something like this:

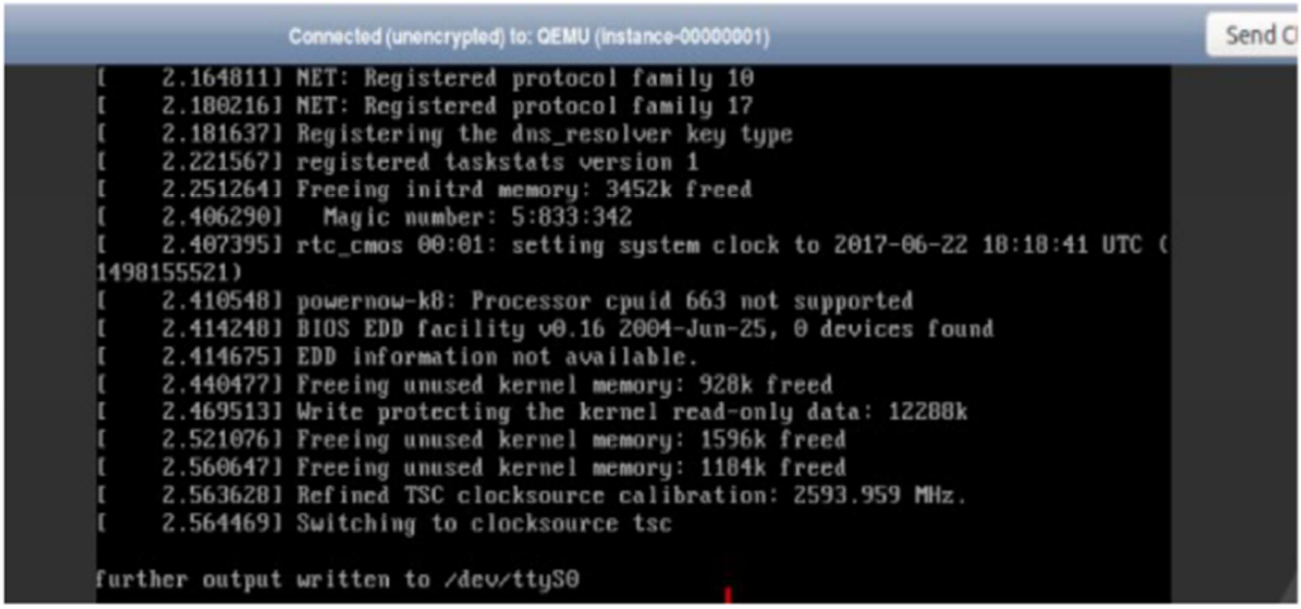


Step 5: Access Virtual Machine Console!

Once you click “Launch Instance” it will take OpenStack few seconds to create your virtual machine. Once ready you can access the console to see how the command line of your first virtual machine looks like.



Click on “Console” and OpenStack will take you to the console of the virtual machine. The console will look something like this:



RESULT:

The procedure for find a procedure to launch virtual machine using trystack(online openstack demo version) was learned and verified successfully

EXNO.:8

Install Hadoop single node cluster and run simple applications like word count.

Aim:

To Install Hadoop single node cluster and run simple Applications like word count.

Steps:

Install Hadoop

Step1: Click here to down load the Java 8 Package. Save this file in your home directory.

Step2: Extract the JavaTarFile.

***Command*:**tar -xvf jdk-8u101-linux-i586.tar.gz

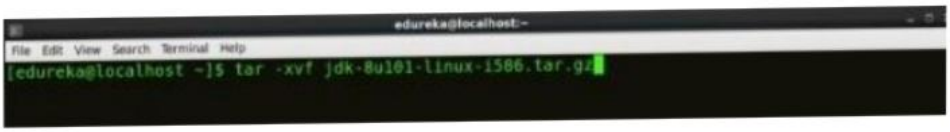


Fig:Hadoop Installation–Extracting Java Files

Step3:**DownloadtheHadoop2.7.3 Package.**

***Command*:**wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

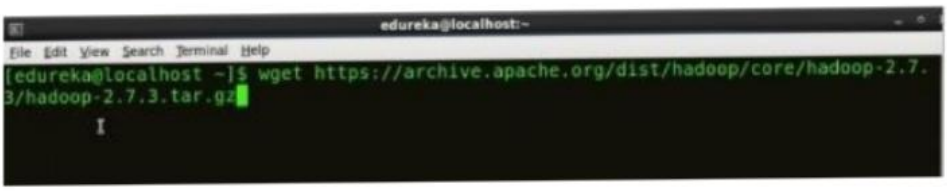


Fig: Hadoop Installation– Downloading Hadoop

Step4: Extract the Hadoop tar File.

Command:tar-xvfhadoop-2.7.3.tar.gz

Fig: Hadoop Installation – Extracting Hadoop Files **Step 5:**

Add the Hadoop and Java paths in the bash file (.bashrc). Open**.b ashrc** file.

Now, add Hadoop and Java Path as shown below.

***Command*:**vi.bashrc

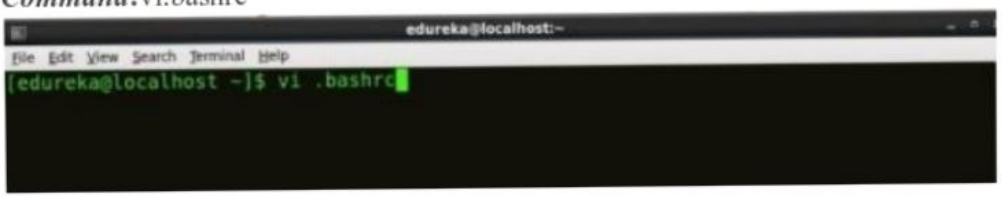


Fig:Hadoop Installation–Setting Environment Variable

Then, save the bash file and close it.

For apply in gall the sechan gesto the current Terminal, execute the source command.
***Command*:** source.bashrc

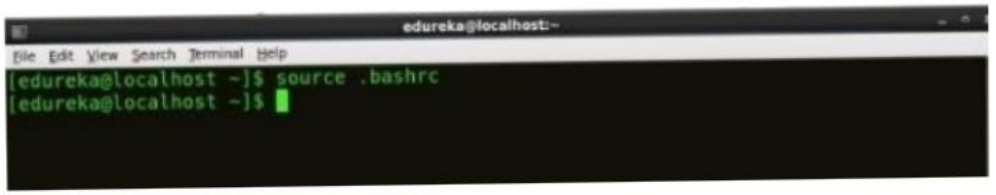
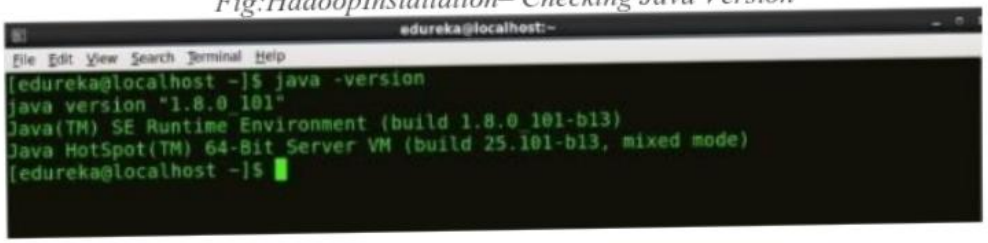


Fig:HadoopInstallation–Refreshing environment variables

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

***Command*:**java-version

Fig:HadoopInstallation– Checking Java Version



***Command*:** hadoop version

```
edureka@localhost:~
File Edit View Search Terminal Help
[edureka@localhost ~]$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be
5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-co
mmon-2.7.3.jar
[edureka@localhost ~]$
```

Fig:Hadoop Installation– Checking Hadoop Version

Step6: Edit the **Hadoop Configuration files**.

Command: cdhadoop-2.7.3/etc/hadoop/

Command:ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you can see in the snapshot below:

```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh               mapred-env.sh
configuration.xsl           httpfs-log4j.properties     mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret     mapred-site.xml.template
core-site.xml               httpfs-site.xml             slaves
hadoop-env.cmd              kms-acls.xml                ssl-client.xml.example
hadoop-env.sh               kms-env.sh                  ssl-server.xml.example
hadoop-metrics2.properties  kms-log4j.properties        yarn-env.cmd
hadoop-metrics.properties   kms-site.xml                yarn-env.sh
hadoop-policy.xml           log4j.properties            yarn-site.xml
hdfs-site.xml               mapred-env.cmd
[edureka@localhost hadoop]$
```

Fig:Hadoop Installation– Hadoop Configuration Files

Step 7: Open *core-site.xml* and edit the property mentioned below inside configuration tag:

core-site.xml informs Hadoop daemon where Name Node runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & Map Reduce.

***Command*:**vicore-site.xml

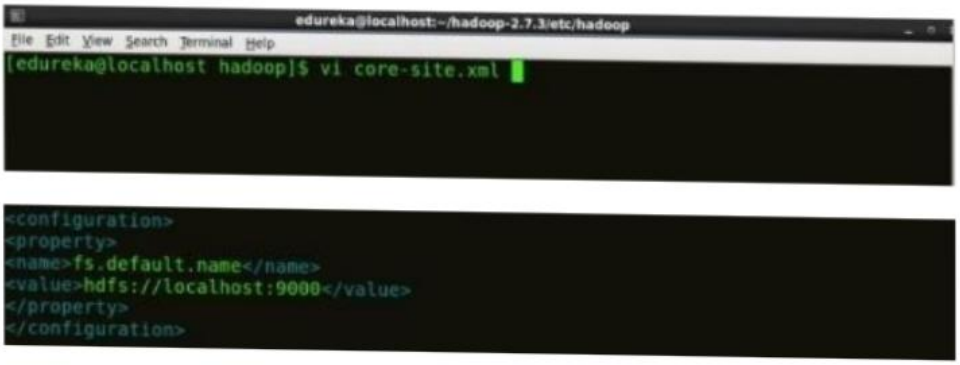


Fig:HadoopInstallation– Configuring core-site.xml

```
1
2                <?xmlversion="1.0"encoding="UTF-8"?>
3      <?xml-stylesheettype="text/xsl"href="configuration.xsl"?>
4                          <configuration>
5                            <property>
6                      <name>fs.default.name</name>
7                    <value>hdfs://localhost:9000</value>
                             </property>
                           </configuration>
```

Step 8: Edit *hdfs-site.xml* and edit the property mentioned below inside **configurationtag:**

hdfs-site.xml contains configuration setting so fHDFS daemons(i.e.NameNode,DataNode, Secondary Name Node). It also includes the replication factor and block size of HDFS.

***Command*:** vihdfs-site.xml

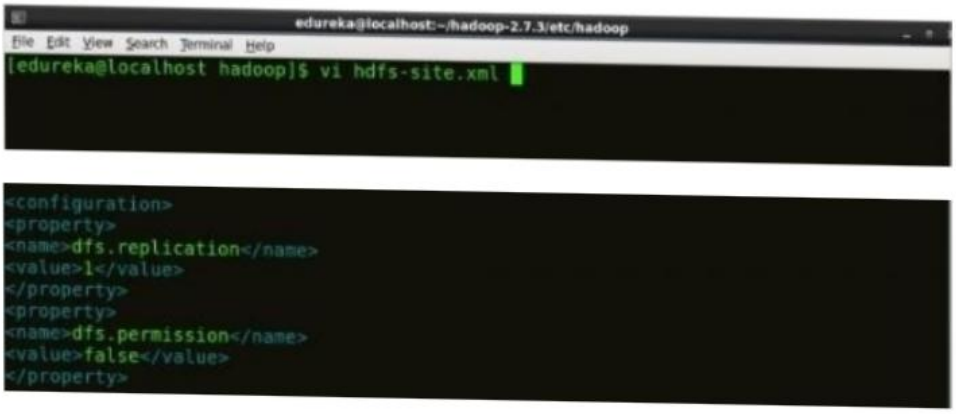


Fig:Hadoop Installation– Configuringhdfs-site.xml

```
<?xmlversion="1.0"encoding="UTF-8"?>
<?xml-stylesheettype="text/xsl"href="configuration.xsl"?>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
</configuration>
```

Step9: Edit the *mapred-site.xml* file and edit the property mentioned below

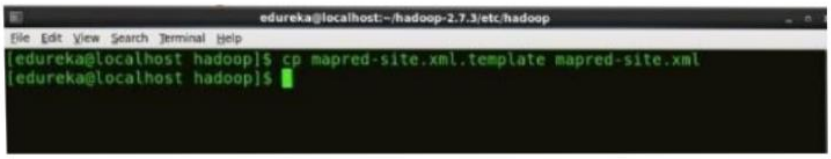
Inside configuration tag:

mapred-site.xml contains configuration settings of Map Reduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the map red- site.xml file using mapred-site.xml template.

***Command*:** cpmapred-site.xml. template mapred-site.xml

***Command*:**vimapred-site.xml.



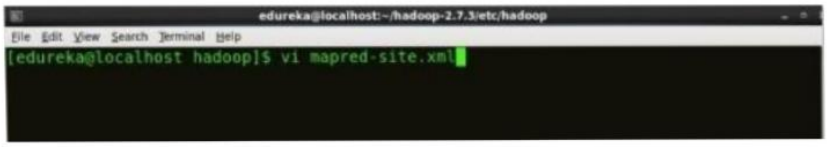


Fig:Hadoop Installation– Configuring mapred-site.xml

```
<?xmlversion="1.0"encoding="UTF-8"?>
<?xml-stylesheettype="text/xsl"href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Step 10: Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:

yarn-site.xml contains configuration settings of Resource Manager and Node Manager like application memory management size, the operation needed on program & algorithm, etc.

***Command*:**viyarn-site.xml

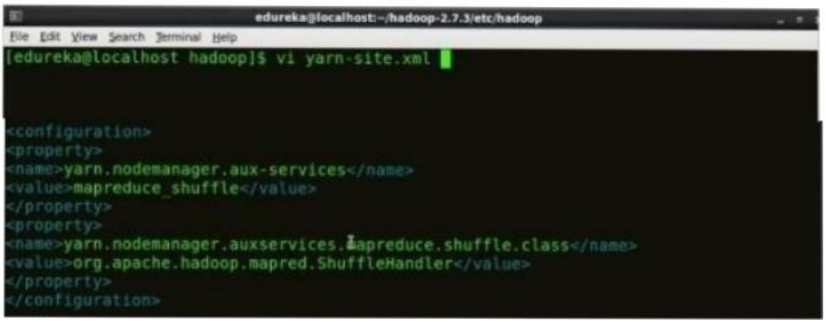


Fig:HadoopInstallation– Configuring yarn-site.xml

Step11: Edit *hadoop-env.sh* and add the JavaPath as mentioned below:

```
<?xmlversion="1.0">
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</
name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

hadoop-env.sh contains the environment variables that are used in the script to run Hadoop

like Java home path, etc.

***Command*:**vihadoop–env.sh

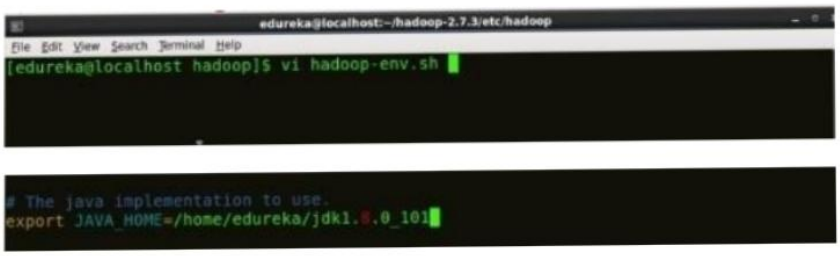


*Fig:HadoopInstallation–Configuringhadoop-env.sh***Step 12:**

Go to Hadoop home directory and format the Name Node.

***Command*:** cd

***Command*:** cd hadoop-2.7.3

***Command*:** bin/hadoop name node-format

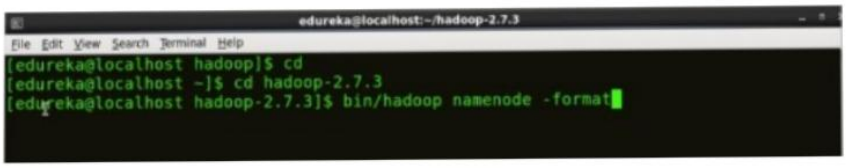


Fig:HadoopInstallation– Formatting NameNode

This formats the HDFS via Name Node.This command is only executed for the firsttime. Formatting the file system means initializing the directory specified by the dfs. name.dir variable.

Never format, up and running Hadoop file system. You will lose all your data stored in the HDFS.

Step13:**Once the Name Node is formatted, goto hadoop-2.7.3/s bin directory and start all the daemons.**

Command: cdhadoop-2.7.3/sbin

Either you can start all daemons with a single command or do it individually.

***Command*:**./start-all.sh

The above command is a combination of ***start-dfs.sh,start-yarn.sh&mr-jobhistory-daemon.sh***

Or you can run all the services individually as below:

Start Name Node:

The Name Node is the center piece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

Command: ./hadoop-daemon.sh start name node

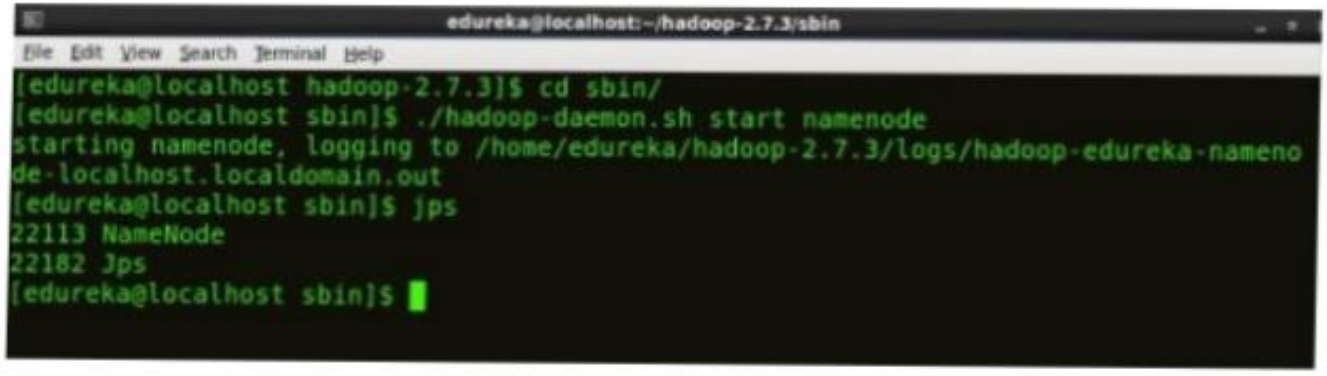


Fig: Hadoop Installation–Starting Name Node

Start Data Node:

On startup, a Data Node connects to the Name node and it responds to the requests from the Name node for different operations.

Command:./hadoop-daemon.sh start data node

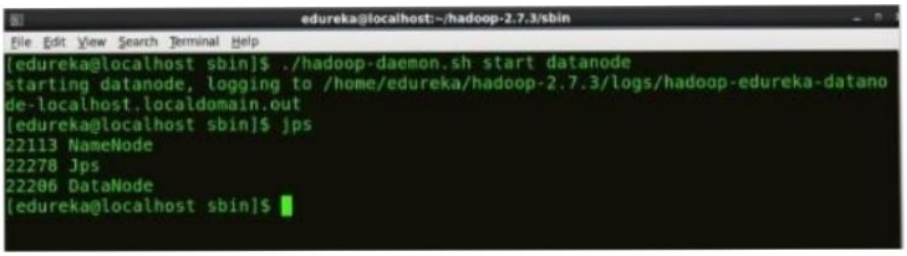


Fig: Hadoop Installation–StartingDataNode

Start Resource Manager:

Resource Manager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system.Its work is to manage each Node Managers and the each application's Application Master.

Command:./yarn-daemon.shstart resourcemanager

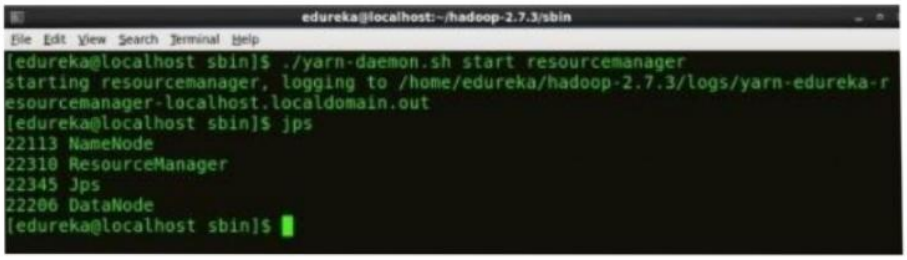


Fig:Hadoop Installation–StartingResourceManager

Start Node Manager:

The Node Manager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the Resource Manager.

Command:./yarn-daemon.sh start node manager

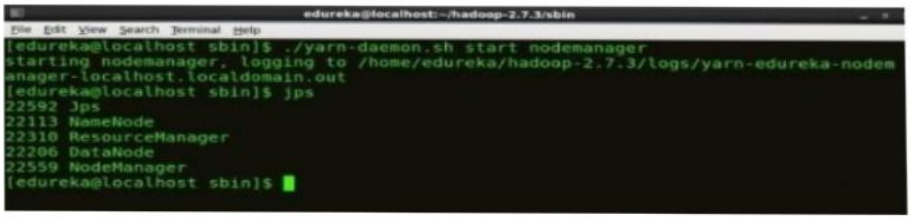


Fig:Hadoop Installation–Starting Node Manager

Start Job History Server:

Job History Server is responsible for servicing all job history related requests from client.

***Command*:**./mr-jobhistory-daemon.sh start history server

Step14:**To check that all the Hadoops ervices are up and running, run the below command.**

Command:jps

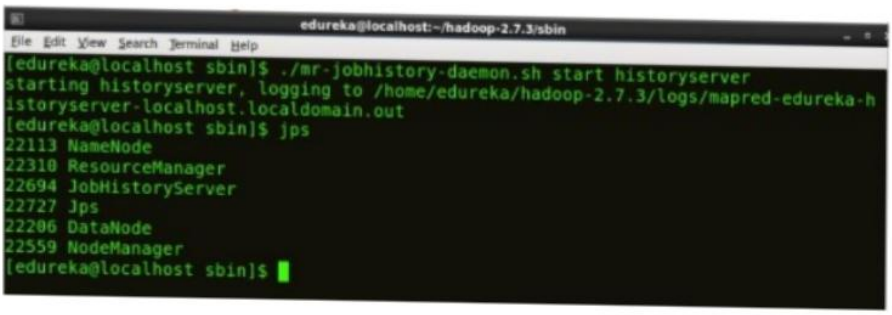


Fig:Hadoop Installation– Checking Daemons

Step15: Now open the Mozilla browser and go
To **localhost:50070/dfshealth.html** to check the NameNode interface.

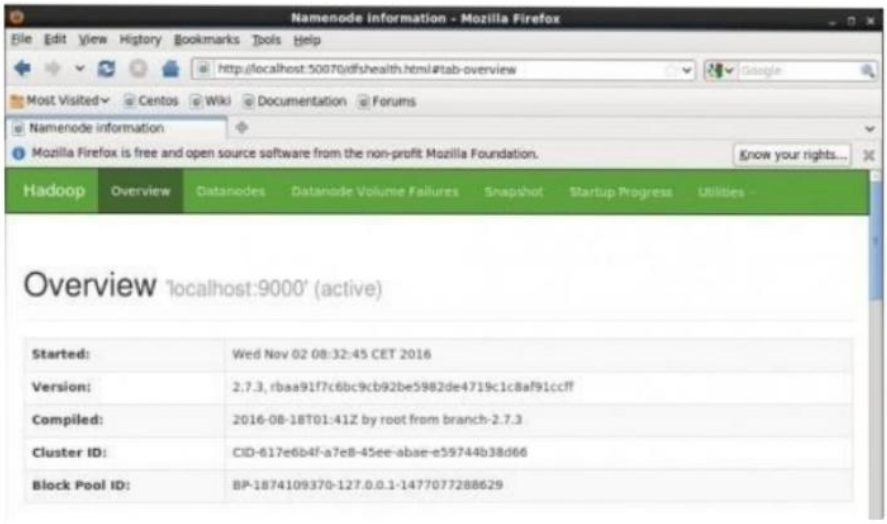


Fig: Hadoop Installation–StartingWebUI

Congratulations, you have successfully installed a single node Hadoop cluster

Result:

Thus the Hadoop one cluster was installed and simple applications executed successfully.

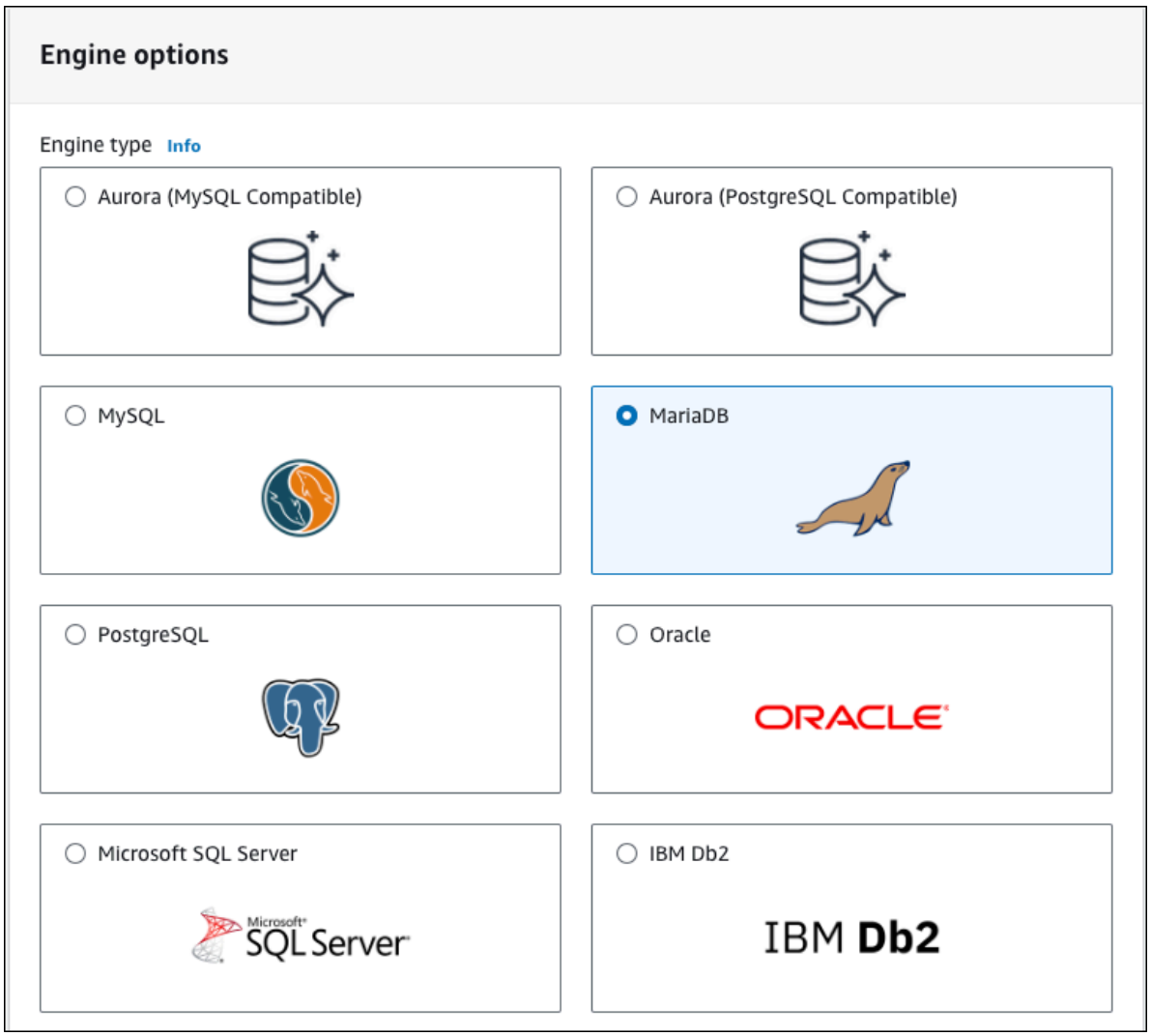
Create an Amazon RDS DB instance

Create an RDS for MariaDB, RDS for MySQL, or RDS for PostgreSQL DB instance that maintains the data used by a web application.

RDS for MariaDB | RDS for MySQL | RDS for PostgreSQL

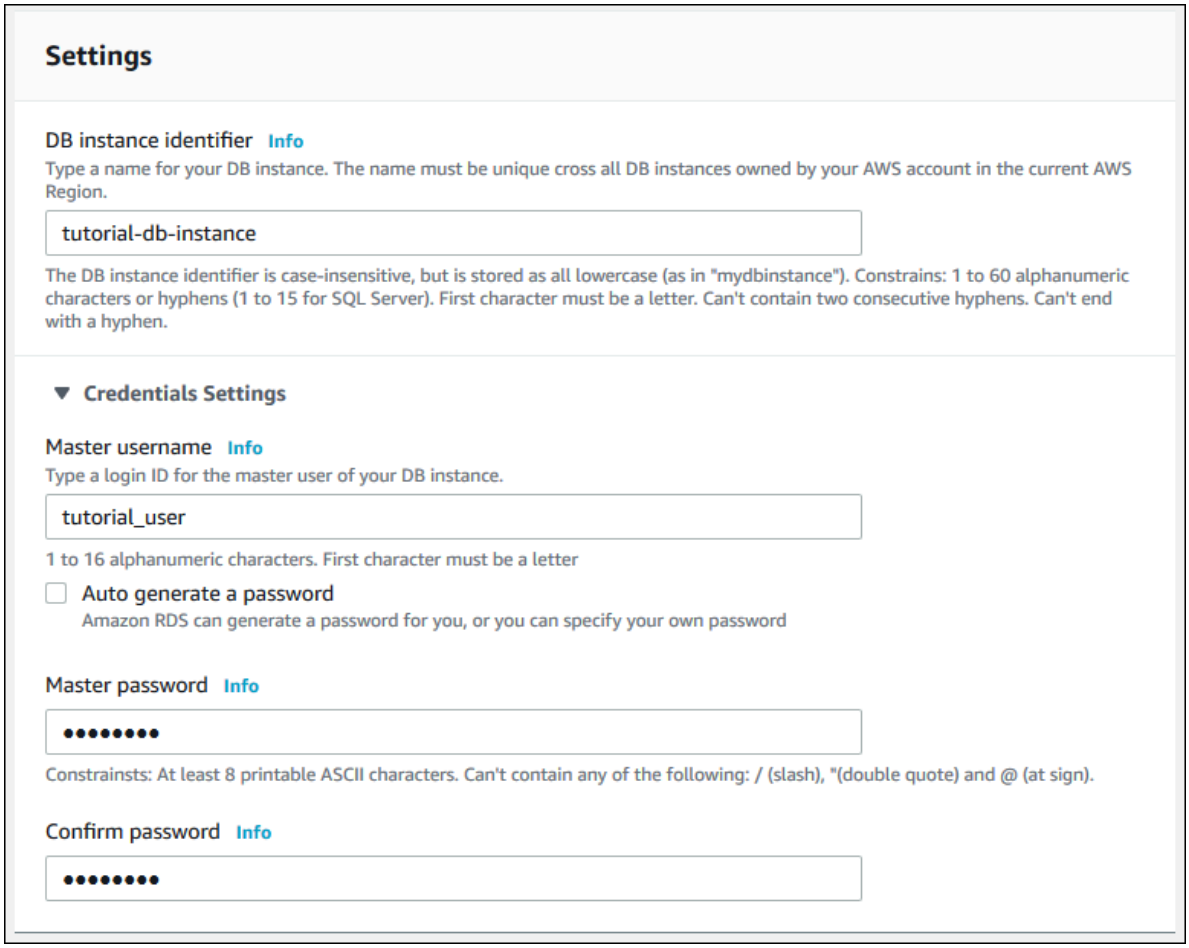
To create a MariaDB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/ (https://console.aws.amazon.com/rds/) .
2. In the upper-right corner of the AWS Management Console, check the AWS Region. It should be the same as the one where you created your EC2 instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database**.
5. On the **Create database** page, choose **Standard create**.
6. For **Engine options**, choose **MariaDB**.

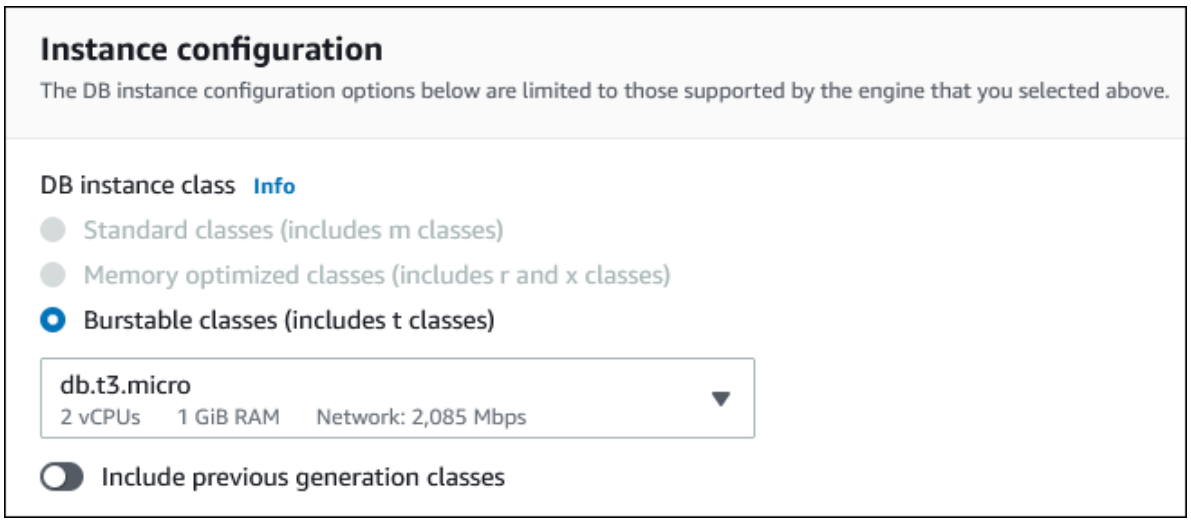


7. For **Templates**, choose **Free tier** or **Sandbox**. **Free tier** appears for free plan accounts. **Sandbox** appears for paid plan accounts.

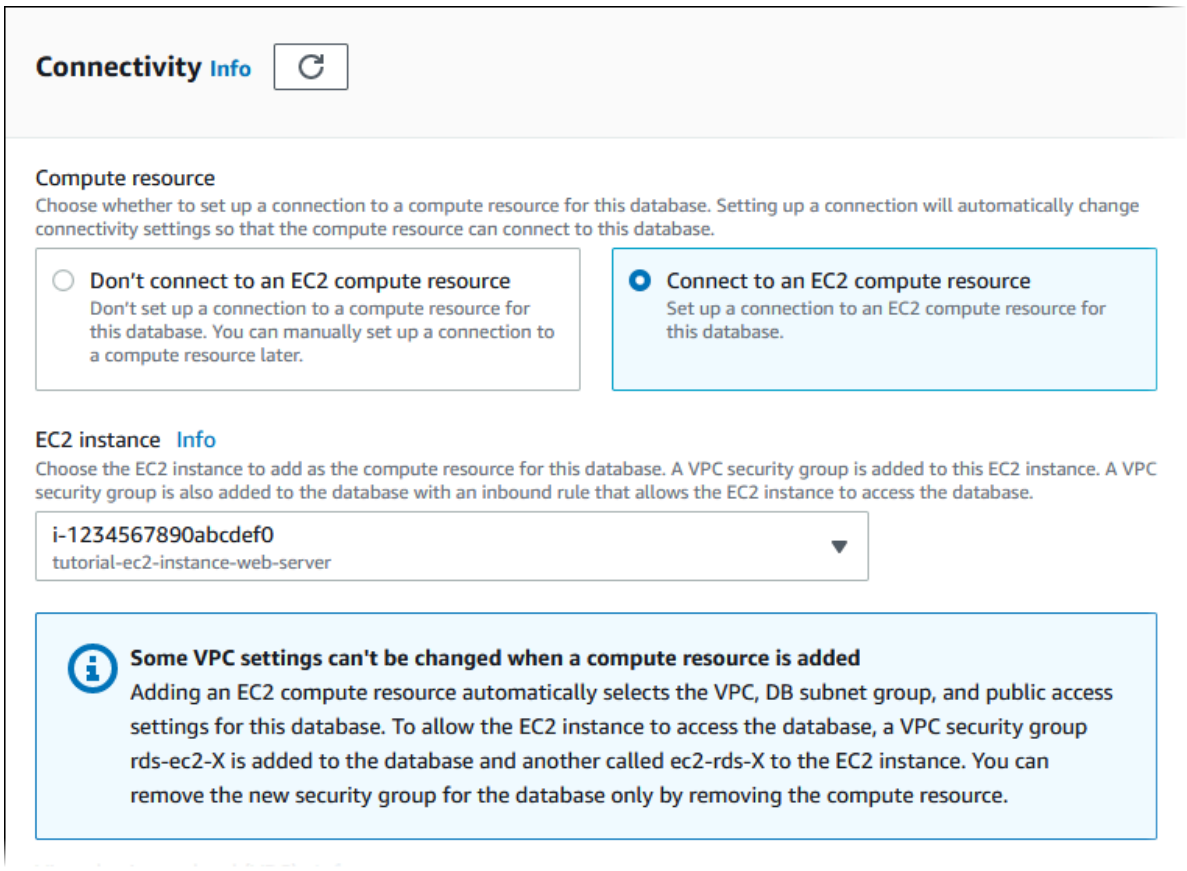
8. In the **Availability and durability** section, keep the defaults.
9. In the **Settings** section, set these values:
    - **DB instance identifier** – Type `tutorial-db-instance`.
    - **Master username** – Type `tutorial_user`.
    - **Auto generate a password** – Leave the option turned off.
    - **Master password** – Type a password.
    - **Confirm password** – Retype the password.



10. In the **Instance configuration** section, set these values:
    - **Burstable classes (includes t classes)**
    - **db.t3.micro**

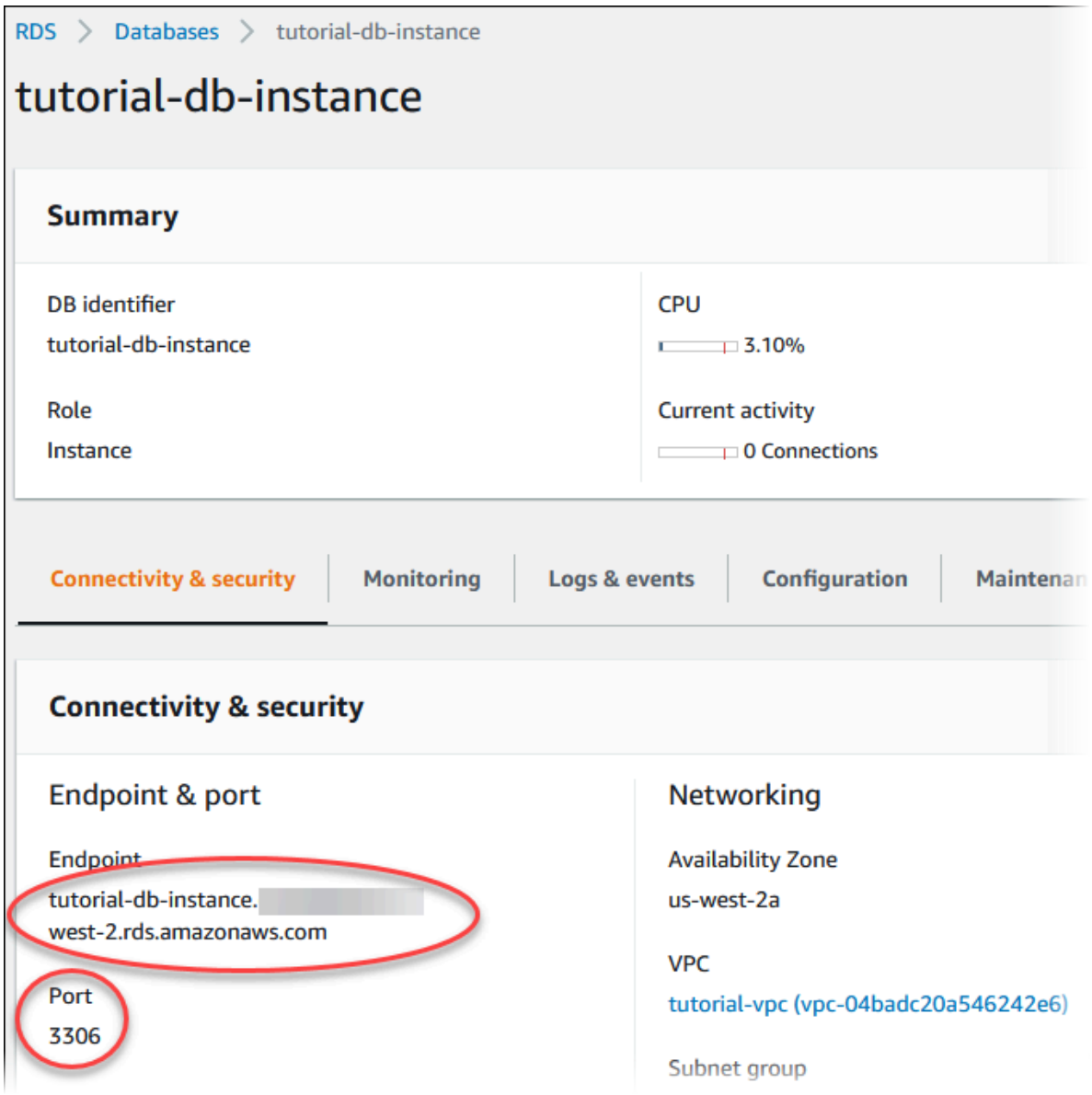


11. In the **Storage** section, keep the defaults.
12. In the **Connectivity** section, set these values and keep the other values as their defaults:
    - For **Compute resource**, choose **Connect to an EC2 compute resource**.
    - For **EC2 instance**, choose the EC2 instance you created previously, such as **tutorial-ec2-instance-web-server**.



13. In the **Database authentication** section, make sure **Password authentication** is selected.
14. Open the **Additional configuration** section, and enter `sample` for **Initial database name**. Keep the default settings for the other options.
15. To create your MariaDB instance, choose **Create database**.

    Your new DB instance appears in the **Databases** list with the status **Creating**.
16. Wait for the **Status** of your new DB instance to show as **Available**. Then choose the DB instance name to show its details.
17. In the **Connectivity & security** section, view the **Endpoint** and **Port** of the DB instance.



Note the endpoint and port for your DB instance. You use this information to connect your web server to your DB instance.

18. Complete Install a web server on your EC2 instance (./CHAP_Tutorials.WebServerDB.CreateWebServer.html) .

Related resources

Amazon RDS API Reference (https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/index.html)

AWS CLI commands for Amazon RDS (https://docs.aws.amazon.com/cli/latest/reference/rds/)

SDKs & Tools (https://aws.amazon.com/tools/)

Recommended tasks

How to

Install or update the latest version of the AWS CLI
(https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)

Connect CodeBuild to GitLab repositories for build projects
(https://docs.aws.amazon.com/codebuild/latest/userguide/access-tokens-gitlab-overview.html)

Connect to Linux instances using SSH for remote access
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connect-to-linux-instance.html)

Create an EC2 instance to connect with your DB instance
(https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.LaunchEC2.html)

Install and configure a web server on an EC2 instance
(https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html)

Learn about

Understand the features and capabilities of Amazon EC2
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html)

Configure retry options for Firehose delivery
(https://docs.aws.amazon.com/firehose/latest/APIReference/API_AmazonOpenSearchServerlessRetryOptions.html)

Recently added to this guide

View related pages

Abstracts generated by AI

1 2 3

AmazonRDS › AuroraUserGuide
Create an Amazon Aurora DB cluster
(https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_Tutorials.WebServerD…

Create an Aurora MySQL or PostgreSQL DB cluster, connect to an EC2 instance, and install a web server to power your app.

25 January 2024

Timestream › developerguide

Creating and connecting to a Timestream for InfluxDB instance

(https://docs.aws.amazon.com/timestream/latest/developerguide/timestream-for-influx-getting-…

Tutorial creates EC2 instance, InfluxDB DB instance, sends Telegraf data to InfluxDB, accesses InfluxDB UI, deletes instances.

6 November 2025

Kendra › dg

Getting started with a MySQL database data source (console)

(https://docs.aws.amazon.com/kendra/latest/dg/getting-started-mysql.html)

MySQL database configured with VPC, security group, credentials, connection, table, and data source for Amazon Kendra indexing.

5 November 2025

Discover highly rated pages

Abstracts generated by AI

AmazonRDS › UserGuide

Regions, Availability Zones, and Local Zones

(https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.RegionsAndAvailabilityZo…

Amazon RDS enables placing resources like DB instances in multiple locations including Regions, Availability Zones, and Local Zones for low-latency access.

29 September 2024

Create instances

Release Notes

MySQL | PostgreSQL (/sql/docs/postgres/create-instance) | SQL Server (/sql/docs/sqlserver/create-instance)

This page describes how to create a Cloud SQL for MySQL instance.

For detailed information about all instance settings, see Instance settings (/sql/docs/mysql/instance-settings).

A newly-created instance has four system databases:

- `information_schema` (https://dev.mysql.com/doc/refman/8.0/en/information-schema.html): Provides access to database metadata, information about the MySQL server.
- `mysql` (https://dev.mysql.com/doc/refman/8.0/en/system-schema.html): The system schema. It contains tables that store information required by the MySQL server as it runs.
- `performance_schema` (https://dev.mysql.com/doc/refman/8.0/en/performance-schema.html): A feature for monitoring MySQL Server execution at a low level.
- `sys` (https://dev.mysql.com/doc/refman/8.0/en/sys-schema.html): Contains a set of objects that helps DBAs and developers interpret data collected by the performance schema.

The maximum number of instances you can have in a single project depends on the network architecture (/sql/docs/mysql/upgrade-cloud-sql-instance-new-network-architecture#overview) of those instances:

- New SQL network architecture: You can have up to 1000 instances per project.
- Old SQL network architecture: You can have up to 100 instances per project.
- Using both architectures: Your limit will be somewhere between 100 and 1000, depending on the distribution of your instances across the two architectures.

File a support case (https://support.google.com/googlecloud/answer/1041916) to request an increase. Read replicas are counted as instances.

Note: This page contains features related to Cloud SQL editions. For more information about Cloud SQL editions, see Introduction to Cloud SQL editions (/sql/docs/mysql/editions-intro).

Before you begin

1. Start by creating a Google Cloud account. With this account, you get $300 in free credits, plus free usage of over 20 products, up to monthly limits.

   Create an account (https://console.cloud.google.com/freetrial)

2. In the Google Cloud console, on the project selector page, select or create a Google Cloud project.

   Roles required to select or create a project

   - **Select a project**: Selecting a project doesn't require a specific IAM role—you can select any project that you've been granted a role on.
   - **Create a project**: To create a project, you need the Project Creator (`roles/resourcemanager.projectCreator`), which contains the `resourcemanager.projects.create` permission. Learn how to grant roles (/iam/docs/granting-changing-revoking-access).

★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

   Go to project selector (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Verify that billing is enabled for your Google Cloud project (/billing/docs/how-to/verify-billing-enabled#confirm_billing_is_enabled_on_a_project).
4. Install (/sdk/docs/install) the gcloud CLI (/sdk/gcloud).
5. If you're using an external identity provider (IdP), you must first sign in to the gcloud CLI with your federated identity (/iam/docs/workforce-log-in-gcloud).
6. To initialize (/sdk/docs/initializing) the gcloud CLI, run the following command:

```
gcloud init
```

7. Make sure you have the Cloud SQL Admin and Compute Viewer roles on your user account.

   Go to the IAM page (https://console.cloud.google.com/iam-admin/iam)

   Learn more (/sql/docs/mysql/roles-and-permissions) about roles and permissions.

Create a MySQL instance

Important: For your Cloud SQL Enterprise Plus edition instance, Cloud SQL can generate a write endpoint automatically. For more information about this endpoint, including requirements for generating one automatically, see Generate the write endpoint (#generate-write-endpoint).

Tip: If you plan on using private networking, then you can deploy both the private networking setup of your choice and the Cloud SQL instance along with clients such as Compute Engine VMs by using Terraform. For more information, see Simplified Cloud Networking Configuration Solutions (https://github.com/GoogleCloudPlatform/cloudnetworking-config-solutions).

Console (#console) gcloud (#gcloud) Terraform (#terraform) More

1. In the Google Cloud console, go to the **Cloud SQL Instances** page.

   Go to Cloud SQL Instances (https://console.cloud.google.com/sql)

2. Click **Create instance**.
3. On the **Choose your database engine** panel of the **Create an instance** page, click **Choose MySQL**.
4. In the **Choose a Cloud SQL edition** section of the **Create a SQL Server instance** page, select the Cloud SQL edition for your instance: `Enterprise` or `Enterprise Plus`.

For more information about Cloud SQL editions, see Introduction to Cloud SQL editions (/sql/docs/mysql/editions-intro).

5. Select the edition preset for your instance. To see the available presets, click the **Edition preset** menu.

★ **Note:** To learn about how edition presets differ from one another, click **Compare edition presets**.

6. In the **Instance info** section, select the database version for your instance. To see the available versions, click the **Database version** menu.

   If you select MySQL 8.0 without a minor version, then automatic minor version upgrade is enabled for your instance. For more information about automatic minor version upgrade, see Automatic upgrade (/sql/docs/mysql/upgrade-minor-db-version#auto-upgrade).

7. Optional: If you're installing MySQL 8.0, then select **Show minor versions**. You can choose a minor version other than the default minor version. If you select a minor version for your instance, then automatic minor version upgrade is disabled for your instance.

   The database version can't be edited after the instance has been created.

★ Note: Only MySQL 8.0.31 and later are compatible with Cloud SQL Enterprise Plus edition. By default, MySQL 8.4 is Cloud SQL Enterprise Plus edition.

8. In the **Instance ID** field of the **Instance info** pane, enter an ID for your instance.

❗ Don't include sensitive or personally identifiable information in your instance name.

   You do not need to include the project ID in the instance name. This is done automatically where appropriate (for example, in the log files).

9. Set a password for the **root** user.

   Although there's an option to set **No password**, this isn't recommended for security reasons.

To see the password in clear text, click the **Show password** icon.

You can either enter the password manually or click **Generate** to have Cloud SQL create a password for you automatically.

10. Optional: Configure a password policy for the instance as follows:

    a. Select the **Enable password policies** checkbox.

    ★ Note: When you enable a password policy, statements that create users or change user passwords can cause additional latency due to password policy verification.

    b. Click the **Set password policy** button, set one or more of the following options, and click **Save**.

    - **Minimum length**: Specifies the minimum number of characters that the password must have.
    - **Password complexity**: Checks if the password is a combination of lowercase, uppercase, numeric, and non-alphanumeric characters.
    - **Restrict password reuse**: Specifies the number of previous passwords that you can't reuse.

      Supported only on Cloud SQL for MySQL 8.0 and later.
    - **Disallow username**: Prevents the use of the username in the password.

    ★ Note: When you deselect the **Enable password policies** checkbox, the password policy parameters are reset.

11. In the **Choose region and zonal availability** section, select the region and zone for your instance. Region availability might be different based on your Cloud SQL for MySQL edition. For more information, see About instance settings (/sql/docs/mysql/instance-settings).

    Place your instance in the same region as the resources that access it. The region you select can't be modified in the future. In most cases, you don't need to specify a zone.

★ **Note:** If there is a resource location constraint on your organization policy, you must select one of the regions that the organization policy allows. You see a message about Resource Location Restriction in the **Choose region and zonal availability** section if a constraint exists. Learn more (/resource-manager/docs/organization-policy/defining-locations).

If you are configuring your instance for high availability (/sql/docs/mysql/high-availability), you can select both a primary and secondary zone.

The following conditions apply when the secondary zone is used during instance creation:

- The zones default to `Any` for the primary zone and `Any (different from primary)` for the secondary zone.
- If both the primary and secondary zones are specified, they must be distinct zones.

12. In the **Customize your instance** section, update settings for your instance. Begin by clicking **SHOW CONFIGURATION OPTIONS** to display the groups of settings. Then, expand the groups you want to review and customize settings. A **Summary** of all the options you select is shown on the right. Customizing these instance settings is optional. Defaults are assigned in every case where no customizations are made.

    The following table is a quick reference to instance settings. For more details about each setting, see the instance settings (/sql/docs/mysql/instance-settings) page.

Setting	Notes
Machine type	
Machine type	Select from Shared core or Dedicated core. For Shared core, each machine type is classified by the number of CPUs (cores) and amount of memory for your instance.
Cores	The number of vCPUs for your instance. Learn more (/sql/docs/postgres/instance-settings#cpus-postgres).

Setting	Notes
Memory	The amount of memory for your instance, in GBs. Learn more (/sql/docs/postgres/instance-settings#memory-postgres).
Custom	For the Dedicated core machine type, instead of selecting a predefined configuration, select the **Custom** button to create an instance with a custom configuration. When you select this option, you need to select the number of cores and amount of memory for your instance. Learn more (/sql/docs/mysql/create-instance#machine-types).
Data cache	
Enable data cache	By default, the option to enable data cache is selected automatically for Cloud SQL for MySQL Enterprise Plus edition instances. If you don't want to enable data cache, then clear the **Enable data cache** checkbox. For more information about data cache, see data cache (/sql/docs/mysql/instance-settings#data-cache).
Storage	
Storage type	Determines whether your instance uses SSD or HDD storage. Learn more (/sql/docs/mysql/instance-settings#storage-type-2ndgen).
Storage capacity	The amount of storage provisioned for the instance. Learn more (/sql/docs/mysql/instance-settings#storage-capacity-2ndgen) .
Enable automatic storage increases	Determines whether Cloud SQL automatically provides more storage for your instance when free space runs low. Learn more (/sql/docs/mysql/instance-settings#automatic-storage-increase-2ndgen) .
Encryption	
Google-managed encryption	The default option.
Customer key-managed encryption key (CMEK)	Select to use your key with Google Cloud Key Management Service. Learn more (/sql/docs/mysql/cmek).

Setting	Notes
Connections	
Private IP	Adds a private IP address for your instance. To enable connecting to the instance, additional configuration is required. Optionally, you can specify an allocated IP range for your instances to use for connections. a. Expand **Show allocated IP range option**. b. Select an IP range from the drop-down menu. Your instance can have both a public and a private IP address. ★ **Note:** Cloud SQL generates a write endpoint automatically for your Cloud SQL Enterprise Plus edition instance if you do the following: a. If you haven't already enabled the Cloud DNS API, enable the Cloud DNS API (/sql/docs/mysql/configure-private-ip#connect-write-endpoint) for your Google Cloud project. b. Enable the Cloud DNS API (/sql/docs/mysql/configure-private-ip#connect-write-endpoint) for your Google Cloud project (if this API isn't enabled). c. Add a private IP address to the instance. d. Specify an associated network for the instance. e. Optionally, specify an allocated IP range for the instance. • Learn more about using private IP (/sql/docs/mysql/connect-overview#private-ip). • Learn more about allocated IP address ranges (/sql/docs/mysql/private-ip#allocated_ip_address_ranges) .

Setting	Notes
Public IP	Adds a public IP address for your instance. You can then add authorized networks to connect to the instance. Your instance can have both a public and a private IP address. Learn more about using public IP (/sql/docs/mysql/connect-overview#public_ip).
Authorized networks	Add the name for the new network and the Network address. Learn more (/sql/docs/postgres/authorize-networks#authorized-networks) .
Private path for Google Cloud services	By selecting this check box, you allow other Google Cloud services, such as BigQuery, to access data in Cloud SQL and make queries against this data over a private connection. ★ **Note:** This check box is enabled only if you select the **Private IP** check box, and you add or select an authorized network to create a private connection.
Enable Managed Connection Pooling	By selecting this checkbox, you enable Managed Connection Pooling for your instance. Managed Connection Pooling lets you scale your workloads by optimizing resource utilization and connection latency Cloud SQL instances using pooling and multiplexing. For more information about Managed Connection Pooling, see Managed Connection Pooling overview (/sql/docs/mysql/managed-connection-pooling).
Security	
Server certificate authority mode	Choose the type of certificate authority (CA) that signs the server certificate for this Cloud SQL instance. Learn more (/sql/docs/mysql/authorize-ssl#certificate_authority_ca_hierarchies) . By default, when you create an instance in Google Cloud console, the instance uses the **Google managed internal certificate authority** (`GOOGLE_MANAGED_INTERNAL_CA`), which is the per-instance CA option.

Setting	Notes
Data protection	
Backup tier	The backup option (/sql/docs/mysql/backup-recovery/backup-options) of your instance. You can choose between enhanced backups (/sql/docs/mysql/backup-recovery/backup-options#enhanced-backups) and standard backups (/sql/docs/mysql/backup-recovery/backup-options#standard-backups) .
Automate backups	The window of time when you would like backups to start. Learn more (/sql/docs/mysql/instance-settings#backups-and-binary-logging-2ndgen) .
Choose where to store your backups	Select Multi-region for most use cases. If you need to store backups in a specific region, for example, if there are regulatory reasons to do so, select Region and select your region from the Location drop-down menu.
Choose how many automated backups to store	The number of automated backups you would like to retain (from 1 to 365 days). Learn more (/sql/docs/mysql/instance-settings#backup-retention).
Enable point-in-time recovery	Enables point-in-time recovery and write-ahead logging. Learn more (/sql/docs/mysql/instance-settings#backups-and-binary-logging-2ndgen) .

★ **Note:** The following default behavior applies:

- If you create a Cloud SQL Enterprise Plus edition instance, then PITR is enabled by default, regardless of how the instance was created.
- If you create a Cloud SQL Enterprise edition instance in the Google Cloud console, then PITR is enabled by default. Otherwise, if you create the instance by using the gcloud CLI (/sdk/gcloud), Terraform, or the Cloud SQL Admin API, then PITR is disabled by default. In this

Setting	Notes
	case, if you want to use PITR, you must enable it manually.
Enable deletion protection	Determines whether to protect an instance against accidental deletion. Learn more (/sql/docs/mysql/deletion-protection).
Enable retained backups after instance deletion	Determines whether automated and on-demand backups are retained after an instance is deleted. Learn more (/sql/docs/mysql/backup-recovery/backups#retained-backups) .
Choose how many days of logs to retain	Configure write-ahead log retention from 1 to 7 days. The default setting is 7 days. Learn more (/sql/docs/mysql/instance-settings#transaction-log-retention) .
Maintenance	
Preferred window	Determines a one-hour window when Cloud SQL can perform disruptive maintenance on your instance. If you do not set the window, then disruptive maintenance can be done at any time. Learn more (/sql/docs/mysql/instance-settings#maintenance-window-2ndgen) .
Order of updates	Your preferred timing for instance updates, relative to other instances in the same project. Learn more (/sql/docs/mysql/instance-settings#maintenance-timing-2ndgen) .
Flags	
ADD FLAG	You can use database flags to control settings and parameters for your instance. Learn more (/sql/docs/mysql/flags).
Labels	

Setting	Notes
ADD LABEL	Add a key and value for each label that you add. You use labels to help organize your instances.

13. Click **Create Instance**.

    Note: It might take a few minutes to create your instance. However, you can view information about the instance (/sql/docs/mysql/instance-info) while it's being created.

To see how the underlying REST API request (/sql/docs/mysql/admin-api/rest/v1beta4/instances/insert) is constructed for this task, see the APIs Explorer on the instances:insert page (/sql/docs/mysql/admin-api/rest/v1beta4/instances/insert).

Generate the write endpoint

A write endpoint is a global domain name service (DNS) name that resolves to the IP address of the current primary instance automatically. This endpoint redirects incoming connections to the new primary instance automatically in case of a replica failover or switchover (/sql/docs/mysql/replication/cross-region-replicas) operation. You can use the write endpoint in a SQL connection string instead of an IP address. By using a write endpoint, you can avoid having to make application connection changes when a region outage occurs.

For more information about using a write endpoint to connect to an instance, see Connect to an instance using a write endpoint (/sql/docs/mysql/connect-to-instance-using-write-endpoint).

Specify the database minor version for MySQL 8.0

You can specify the minor version of an existing MySQL 8.0 instance by using **`gcloud`** or the REST API.

gcloudREST v1 (#rest-v1)REST v1beta4 (#rest-v1beta4)
(#gcloud)

Use the **`gcloud sql instances create`** (/sdk/gcloud/reference/sql/instances/create) command with the **`--database-version`** flag.

Replace the following variables before running the command:

- ***INSTANCE_NAME***: The name of the instance.
- ***DATABASE_VERSION***: The database minor version of the instance: `MYSQL_8_0_18`, `MYSQL_8_0_26`, `MYSQL_8_0_27`, `MYSQL_8_0_28`, `MYSQL_8_0_29`, `MYSQL_8_0_30`, `MYSQL_8_0_31`, `MYSQL_8_0_32`, `MYSQL_8_0_33`, `MYSQL_8_0_34`, `MYSQL_8_0_35`, `MYSQL_8_0_36`, `MYSQL_8_0_37`, `MYSQL_8_0_39`, `MYSQL_8_0_40`, `MYSQL_8_0_41` (default minor version for MySQL 8.0), `MYSQL_8_0_42`, or `MYSQL_8_0_43`. If you specify `MYSQL_8_0`, the default minor version is used.

  If you don't specify this flag, then the default major version, `MYSQL_8_0`, is used.

```
gcloud sql instances create INSTANCE_NAME \
--database-version=DATABASE_VERSION
```

For detailed information, see the documentation on creating an instance by using **`gcloud`** (/sql/docs/mysql/create-instance#gcloud).

Database minor version for read replicas, clones, and PITR

When creating a read replica (/sql/docs/mysql/replication/create-replica), you can specify the database minor version of the read replica. By default, new read replicas are created on the default minor version.

When cloning an instance (/sql/docs/mysql/clone-instance#cloning-an-instance), the newly created instance has that same minor version as that of the source.

When performing a point-in-time recovery (/sql/docs/mysql/backup-recovery/pitr#perform-pitr), the newly created instance has the same database minor version as that of the source.

Custom instance configurations

Determines memory and virtual cores available for your Cloud SQL instance. Machine types are part of a machine series, and machine series availability is determined by your Cloud SQL edition.

For Cloud SQL Enterprise Plus edition instances, Cloud SQL offers predefined machine types for your instances in the `N2` and `C4A` machine series.

For Cloud SQL Enterprise edition instances, Cloud SQL offers the `general purpose shared core`, `general purpose dedicated core`, and the `N4` machine series.

If you require real-time processing, such as online transaction processing (OLTP), make sure that your instance has enough memory to contain the entire working set. However, there are other factors that can impact memory requirements, such as number of active connections, and internal overhead processes. Perform load testing to avoid performance issues in your production environment.

When you configure your instance, select sufficient memory and vCPUs to handle your needs, and scale up your instance as your requirements increase. A machine configuration with insufficient vCPUs might lose its SLA coverage. For more information, see Operational guidelines (/sql/docs/mysql/operational-guidelines).

To learn more about the machine types and machine series available for your Cloud SQL instance, see Machine series overview (/sql/docs/mysql/machine-series-overview).

Tip: If you plan on using private networking, then you can deploy both the private networking setup of your choice and the Cloud SQL instance by using Terraform.

For more information, see Cloud SQL Simplified Networking (https://github.com/GoogleCloudPlatform/terraform-google-cloudsqlnetworking).

Troubleshoot

Issue	Troubleshooting
Error message: `Failed to create subnetwork. Couldn't find free blocks in allocated IP ranges. Please allocate new ranges for this service provider`.	There are no more available addr several possible scenarios: • The size of the allocated IP ra than /24. • The size of the allocated IP ra for the number of Cloud SQL i

Issue	Troubleshooting

	• The requirement on the size c
	created in multiple regions. S
	(/sql/docs/mysql/private-ip#
	
	To resolve this issue, you can eit
	allocate an additional IP range tc
	information, see Allocate an IP a
	(/sql/docs/mysql/configure-priv
	
	If you used the **--allocated-i**
	instance, you may only expand th
	
	If you're allocating a new range, t
	existing allocations.
	
	After creating a new IP range, up
	
	`gcloud services vpc-peer`
	`--service=servicenetwork`
	`--ranges=`*`OLD_RESERVED_RA`*
	`--network=`*`VPC_NETWORK`*
	`--project=`*`PROJECT_ID`* `\`
	`--force`
	
	If you're expanding an existing al
	range and not decrease it. For ex
	then make the new allocation at
	
	In general, if starting from a /24 a
	condition (additional instance ty
	For example, if trying to create b
	going from /24 to /23 is enough.
	
	After expanding an existing IP ra
	command:
	
	`gcloud services vpc-peer`
	`--service=servicenetwork`

Issue	Troubleshooting
	`--ranges=`*`RESERVED_RANGE_`* `--network=`*`VPC_NETWORK`* `--project=`*`PROJECT_ID`*
Error message: **`Failed to create subnetwork. Router status is temporarily unavailable. Please try again later. Help Token: `***`[token-ID]`**.	Try to create the Cloud SQL insta
Error message: **`HTTPError 400: Invalid request: Incorrect Service Networking config for instance: `***`PROJECT_ID`***`:`***`INSTANCE_NAME`***`:SERVICE_NETWORKING_NOT_ENABLED.`	Enable the Service Networking A (/service-infrastructure/docs/se following command and try to cr `gcloud services enable s` `--project=`*`PROJECT_ID`*
Error message: **`Failed to create subnetwork. Required 'compute.projects.get' permission for `***`PROJECT_ID`**.	When you create an instance usi created just-in-time using the Se enabled the Service Networking A and the instance creation fails. In propagate throughout the systen
Error message: **`More than 3 subject alternative names are not allowed.`**	You're trying to use a custom SA certificate of a Cloud SQL instan instance.
Error message: **`Subject alternative names %s is too long. The maximum length is 253 characters.`**	Make sure that any DNS names t Cloud SQL instance don't have m
Error message: **`Subject alternative name %s is invalid.`**	Verify that the DNS names that y SQL instance meet the following • They don't have wildcard char • They don't have trailing dots. • They meet RFC 1034 (https:// specifications.

What's next

1. Create a MySQL database on the instance (/sql/docs/mysql/create-manage-databases).
2. Create MySQL users on the instance (/sql/docs/mysql/create-manage-users).
3. Secure and control access to the instance (/sql/docs/mysql/instance-access-control).
4. Connect to the instance with a MySQL client (/sql/docs/mysql/connect-admin-ip).
5. Import data into the database (/sql/docs/mysql/import-export).
6. Learn about instance settings (/sql/docs/mysql/instance-settings).

Last updated 2025-11-04 UTC.