

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.
-

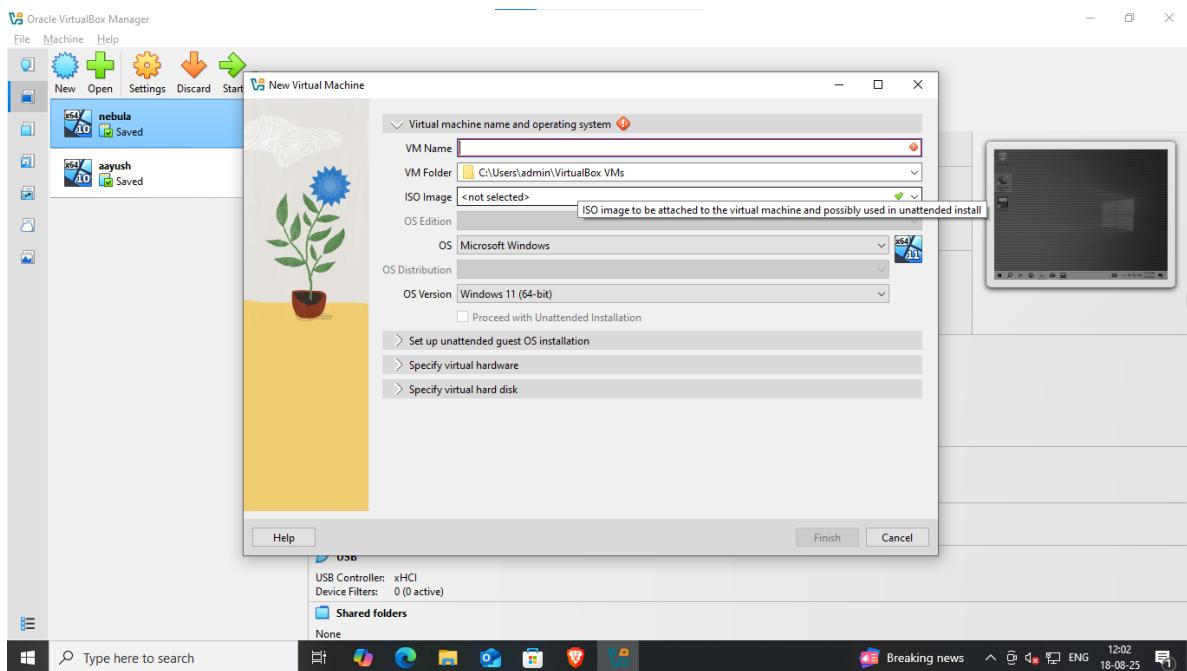
Step 2: Download Microsoft Media Creation Tool for Windows 10

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**.

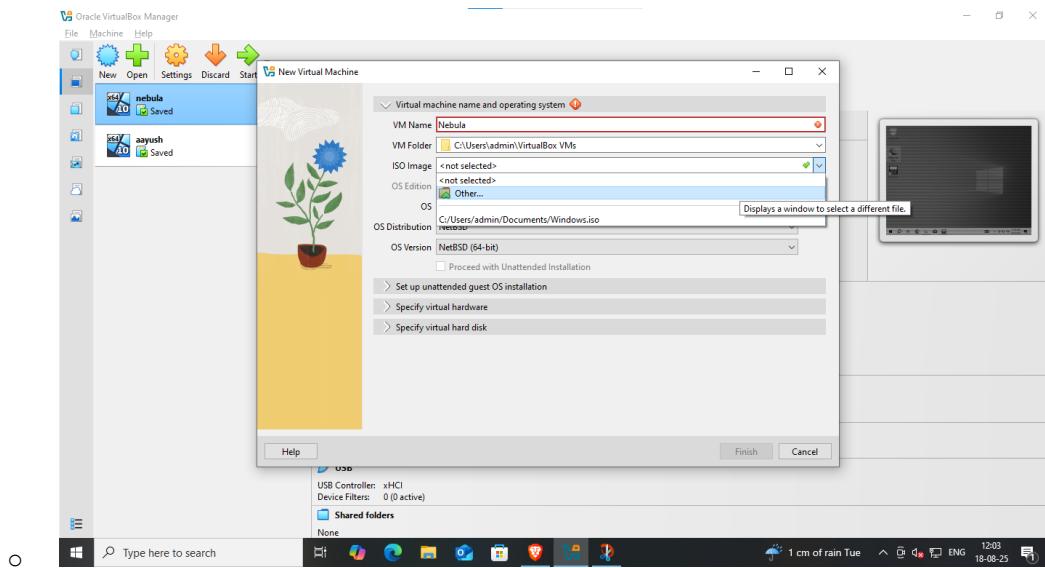
8. Choose a location to save the Windows 10 ISO file and click **Save**.
 9. Wait while the tool downloads the Windows 10 ISO file.
 10. Once completed, click **Finish** to close the tool.
-

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.

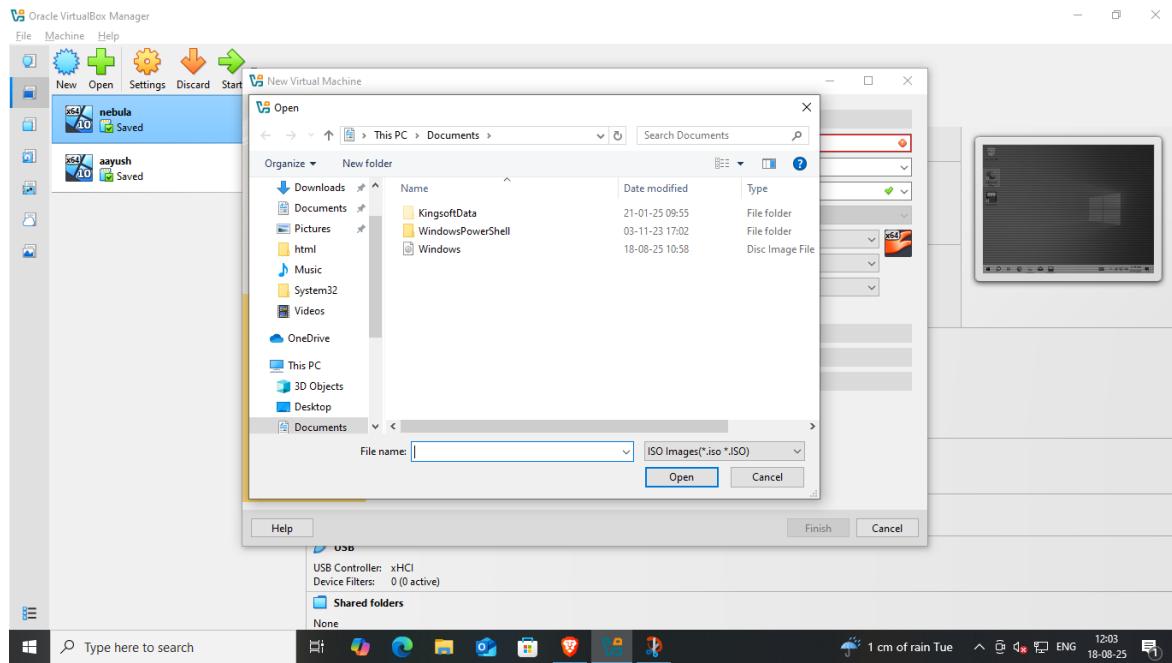


3. In the **Name and Operating System** window:
 - Enter a name for your VM (e.g., "Windows 10").
 - Select **Microsoft Windows** as the type.
 - Select **Windows 10 (64-bit)** or **Windows 10 (32-bit)** depending on the ISO you downloaded.
 - Click **Next**.

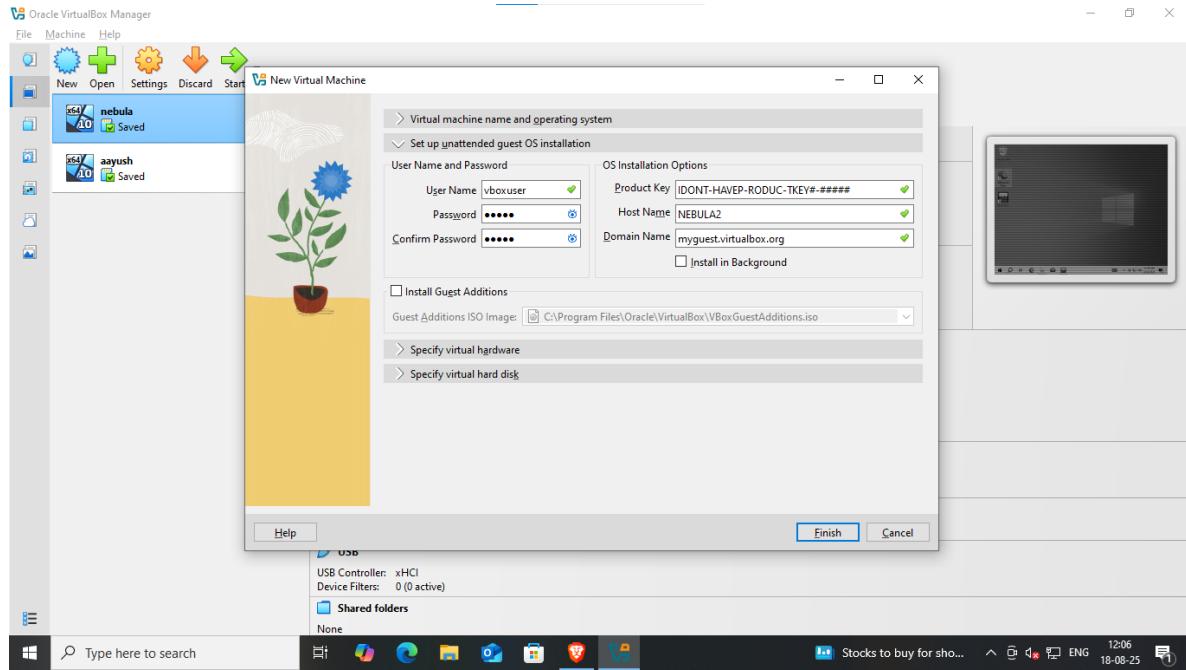


4. Navigate to ISO image and select the Windows 10 ISO file you downloaded.

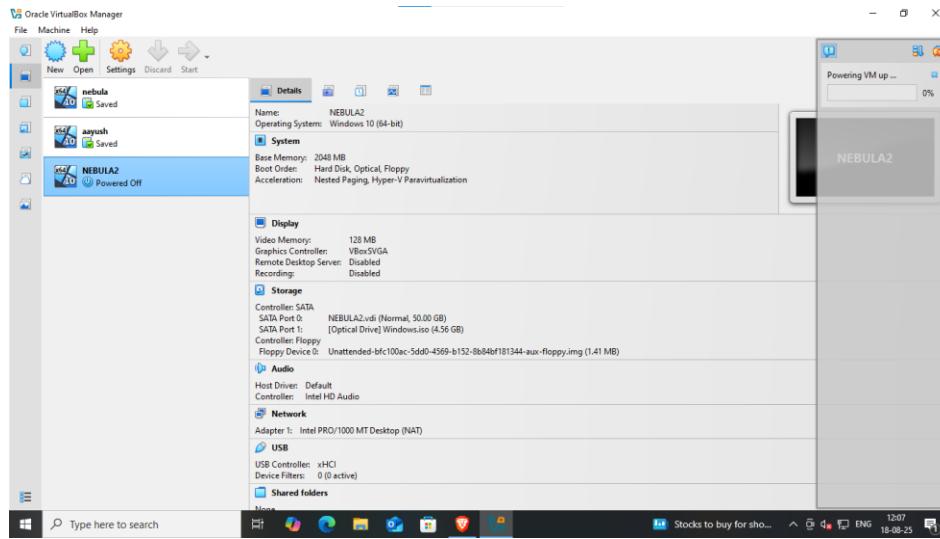
Click **OPEN**



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.
 - o 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**.
 - o 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.

The screenshot shows the AWS sign-up process. On the left, there's a promotional message about Free Tier products and a hand icon holding three cubes. A red arrow points from this icon to the 'Verify email address' button on the right. The right side has fields for 'Root user email address' (containing '@k21academy.com') and 'AWS account name' (with a placeholder). Below these are 'Verify email address' and 'Sign in to an existing AWS account' buttons, separated by an 'OR' link.

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.

Sign up for AWS

Root user email address
Used for account recovery and some administrative functions
@k21academy.com

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.
[Placeholder]

Verify email address

OR

Sign in to an existing AWS account

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

The screenshot shows the AWS sign-up process for creating a password. It includes a success message about email verification, instructions for password creation, and fields for entering a root user password and confirming it. A red arrow points from the 'Continue (step 1 of 5)' button back to the 'Verify email address' step on the previous page.

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.

Sign up for AWS

Create your password

It's you! Your email address has been successfully verified. X

Your password provides you with sign in access to AWS, so it's important we get it right.

Root user password

Confirm root user password

Continue (step 1 of 5)

OR

Sign in to an existing AWS account

- 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.

Sign up for AWS

Choose your account plan



Free (6 months)

Learn, experiment, and build prototypes

- Receive up to \$200 in credits
 - Includes free usage of select services
 - Workloads scale beyond credit thresholds
 - Access to all AWS services and features
- ⓘ After the 6 month free period or when all credits are used, you can choose to upgrade to a paid plan. Otherwise, your account closes automatically.

[Choose free plan](#)



Paid

Develop production-ready workloads

- Receive up to \$200 in credits
- Includes free usage of select services
- Workloads scale beyond credit thresholds
- Access to all AWS services and features

ⓘ After all of your credits are used, you are charged using pay-as-you-go pricing.

[Choose paid plan](#)

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

Free Tier offers

All AWS accounts can explore 3 different types of free offers, depending on the product used.



Always free
Never expires



12 months free
Start from initial sign-up date



Trials
Start from service activation date

Sign up for AWS

Contact Information

How do you plan to use AWS?

- Business - for your work, school, or organization
- Personal - for your own projects

Who should we contact about this account?

Full Name

Phone Number

Country or Region

Address

Apartment, suite, unit, building, floor, etc.

City

State, Province, or Region

Postal Code

Customers with an Indian contact address are served by Amazon Web Services India Private Limited, the local seller for AWS services in India.

I have read and agree to the terms of the AWS Customer Agreement [\[Link\]](#).

[Continue \(step 2 of 5\)](#)

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.

Secure verification

We will not charge you for usage below AWS Free Tier limits. We may temporarily hold up to \$1 USD (or an equivalent amount in local currency) as a pending transaction for 3-5 days to verify your identity.

Billing Information

Credit or Debit card number

VISA

AWS accepts most major credit and debit cards. To learn more about payment options, review our FAQ.

Expiration date: March 2027

Security code

Cardholder's name

Save card information for faster future payments.
Securely save card information payments as per RBI guidelines. Learn more.

Billing address:

Use my contact address

Use a new address

Do you have a PAN?

Permanent Account Number (PAN) is a ten-digit alphanumeric number issued by the Indian Income Tax Department. This 10-digit number is printed on the front of your PAN card.

Yes
 No
You can go on the Tax Settings Page on Billing and Cost Management Console to update your PAN information.

Verify and Continue (step 3 of 5)

You might be redirected to your bank's website to authorize the verification charge.

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 on your Country. Here I have provided India, so Amazon charged 2 INR.

Verified by VISA 06 Feb 2022 12:09:55

ICICI Bank

Card Number XXXX XXXX XXXX 8208

An OTP (One Time Password) has been sent to your registered mobile number. Please authenticate the transaction using this OTP.

Enter OTP

Resend OTP

OTPs are SECRET. DO NOT disclose it to anyone. Bank NEVER asks for OTP.

Submit Cancel

This page will automatically timeout after 180 seconds.

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).



Sign up for AWS

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

Text message (SMS)

Voice call

Country or region code

India (+91)

Mobile phone number

7017083421

Security check



Type the characters as shown above

Send SMS (step 4 of 5)

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.



Sign up for AWS

Confirm your identity

Verify code

7674

Continue (step 4 of 5)

Having trouble? Sometimes it takes up to 10 minutes to retrieve a verification code. If it's been longer than that, [return to the previous page](#) and try again.

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

Select your suitable plan then click continue.



Sign up for AWS

Select a support plan

Choose a support plan for your business or personal account. Compare plans and pricing examples [\[?\]](#). You can change your plan anytime in the AWS Management Console.

- | | | |
|--|--|---|
| <input checked="" type="radio"/> Basic support - Free <ul style="list-style-type: none">Recommended for new users just getting started with AWS24x7 self-service access to AWS resourcesFor account and billing issues onlyAccess to Personal Health Dashboard & Trusted Advisor  | <input type="radio"/> Developer support - From \$29/month <ul style="list-style-type: none">Recommended for developers experimenting with AWSEmail access to AWS Support during business hours12 (business)-hour response times  | <input type="radio"/> Business support - From \$100/month <ul style="list-style-type: none">Recommended for running production workloads on AWS24x7 tech support via email, phone, and chat1-hour response timesFull set of Trusted Advisor best-practice recommendations  |
|--|--|---|



Need Enterprise level support?

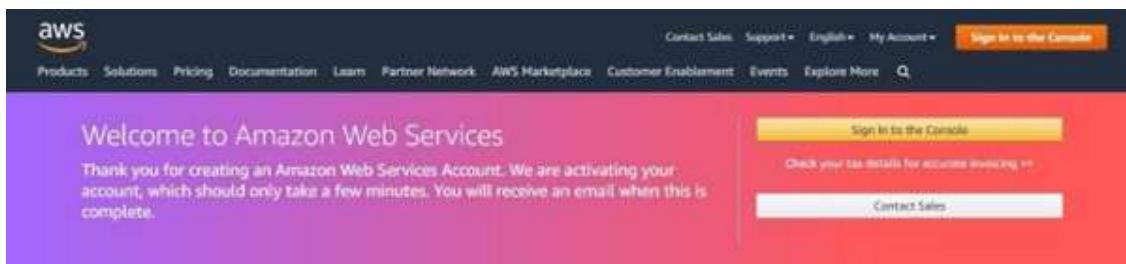
From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. [Learn more \[?\]](#)

[Complete sign up](#)

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.

Search results for 'ec2'

Try searching with longer queries for more relevant results

Services (13)

Features (54)

Resources New

Documentation (33,710)

Knowledge Articles (20)

Marketplace (3,133)

Blogs (2,058)

See all 13 results ▶

EC2 ★
Virtual Servers in the Cloud

EC2 Image Builder ★

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

EC2 Dashboard

EC2 Global View

Events

Instances

Instances 1

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Instances Info

Find Instance by attribute or tag (case-sensitive)

Name Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4 DNS

No instances

You do not have any instances in this region

Launch instances

Select an instance

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

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

my-web-server

Add additional tags

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

The screenshot shows the 'Application and OS Images (Amazon Machine Image)' section. It features a search bar at the top with the placeholder 'Search our full catalog including 1000s of application and OS images'. Below the search bar is a 'Quick Start' section with icons for various operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. To the right of this section is a 'Browse more AMIs' button with the text 'Including AMIs from AWS, Marketplace and the Community'. The main content area displays the 'Amazon Machine Image (AMI)' section for the 'Amazon Linux 2023 AMI'. The details shown are: Name: Amazon Linux 2023 AMI, ID: ami-0dbc3d7bc646e8516 (64-bit (x86)) / ami-055859c8e0f361065 (64-bit (Arm)), Virtualization: hvm, ENA enabled: true, Root device type: ebs. A 'Free tier eligible' badge is present. Below this, there is a 'Description' section with the text 'Amazon Linux 2023 AMI 2023.2.20231018.2 x86_64 HVM kernel-6.1'.

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

The screenshot shows the 'Instance type' section. It lists the 't2.micro' instance type with the following details: Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true. Pricing information is provided for On-Demand Windows, SUSE, RHEL, and Linux base pricing. A note states 'Additional costs apply for AMIs with pre-installed software'. To the right of the instance type list are two buttons: 'All generations' and 'Compare instance types'.

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

The screenshot shows the 'Key pair (login)' section. It contains a note: 'You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.' Below this is a 'Key pair name - required' field with a dropdown menu containing the word 'Select'. To the right of the dropdown is a 'Create new key pair' button with a circular arrow icon.

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**.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

1 web-server-key-pair
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

2 RSA
RSA encrypted private and public key pair

3 ED25519
ED25519 encrypted private and public key pair

Private key file format

4 .pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠️ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

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**.

The screenshot shows a user interface for managing network settings. At the top left is a dropdown menu labeled "Network settings" with a blue "Info" link next to it. To the right is an "Edit" button, which is highlighted with a yellow border. The background is white with some light gray shadows.

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*'.

The screenshot shows the "Firewall (security groups)" configuration page. At the top left is a "Firewall (security groups)" label with a blue "Info" link. Below it is a brief description: "A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance." Two radio buttons are present: "Create security group" (selected, highlighted with a blue border) and "Select existing security group".

The "Security group name - required" field contains "WebServerSG". A note below it states: "This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=;&{}!\$*".

The "Description - required" field contains "Allows SSH and HTTP access".

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

Inbound Security Group Rules

The screenshot shows the "Inbound Security Group Rules" configuration page. It lists two security group rules:

1. **Security group rule 1 (TCP, 22, 15/32, Allow SSH from my IP)**:
 - Type: ssh (highlighted with a yellow border)
 - Protocol: TCP
 - Port range: 22
 - Source type: My IP (highlighted with a yellow border)
 - Name: 15/32
 - Description: Allow SSH from my IP

2. **Security group rule 2 (TCP, 80, 0.0.0.0/0, Allow web traffic on port 80)**:
 - Type: HTTP (highlighted with a yellow border)
 - Protocol: TCP
 - Port range: 80
 - Source type: Anywhere (highlighted with a yellow border)
 - Name: 0.0.0.0/0
 - Description: Allow web traffic on port 80

At the bottom left is a "Add security group rule" button.

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.

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar with placeholder text 'Find Instance by attribute or tag (case-sensitive)'. Below it, a filter bar has 'Instance ID = i-0b554dc7418b6dd89' selected. A 'Clear filters' button is also present. The main table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is highlighted with a yellow background, showing 'my-web-server' in the Name column, 'i-0b554dc7418b6dd89' in the Instance ID column, 'Running' in the Instance state column, 't2.micro' in the Instance type column, '2/2 checks passed' in the Status check column, 'No alarms' in the Alarm status column, and 'us-east-1b' in the Availability Zone column. A circled '1' is next to the checked checkbox in the Name column. A circled '2' is next to the 'Public IPv4 address' field, which contains '34.229.194.201 | open address'.

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
```

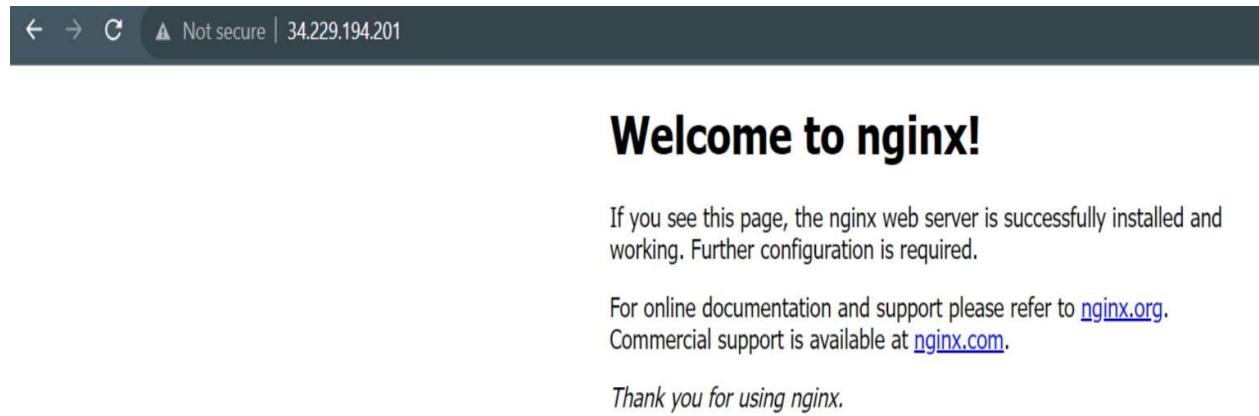
Step-5: Once the update is completed, install Nginx.

```
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

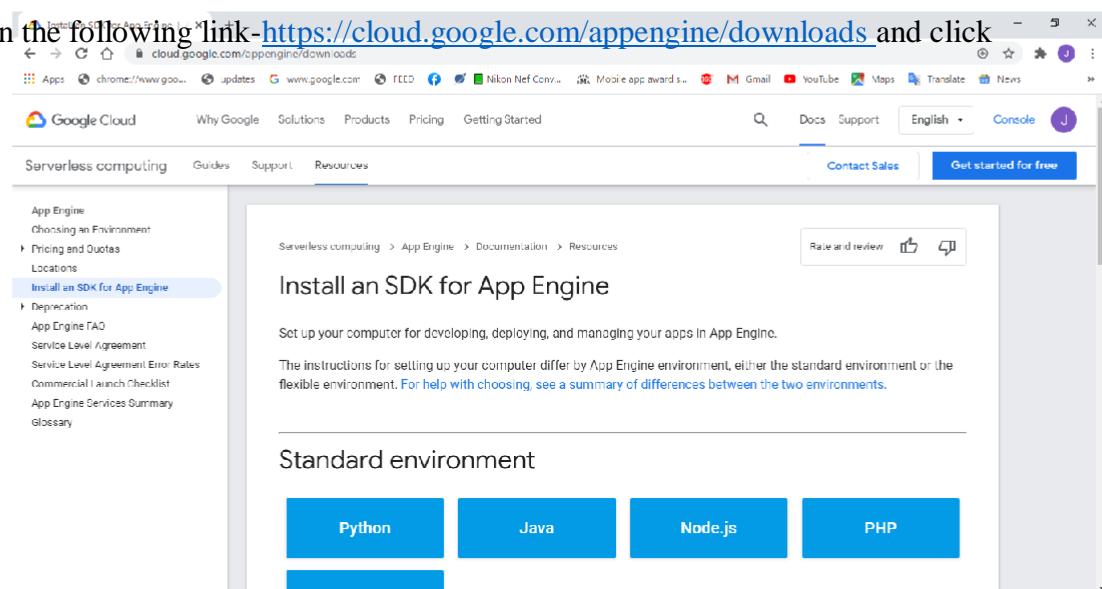
Date:

Aim:

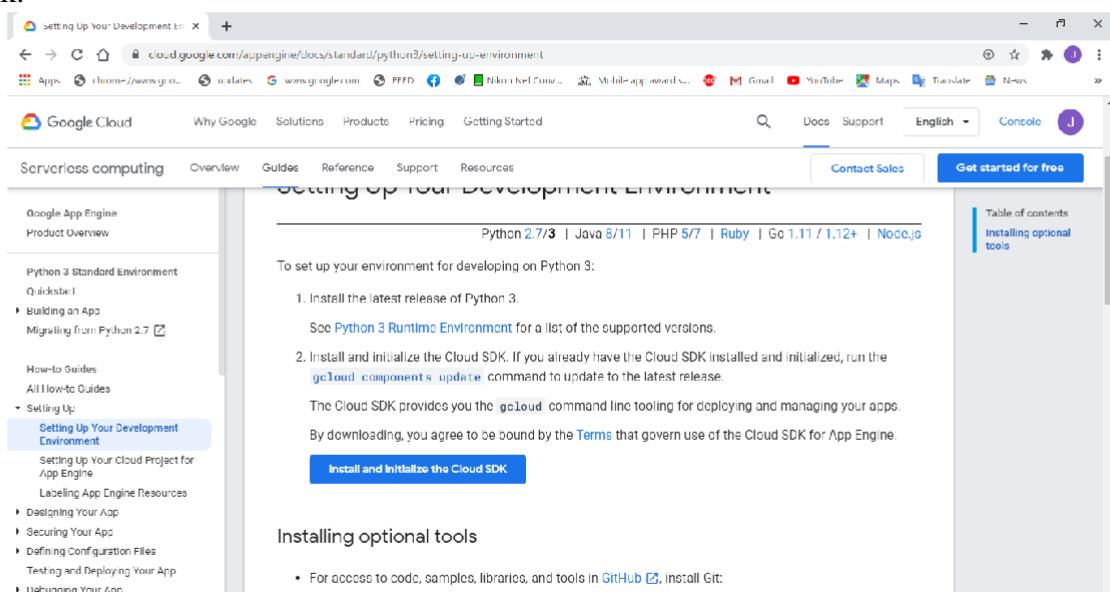
To install the Google app engine in the system.

Procedure:

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



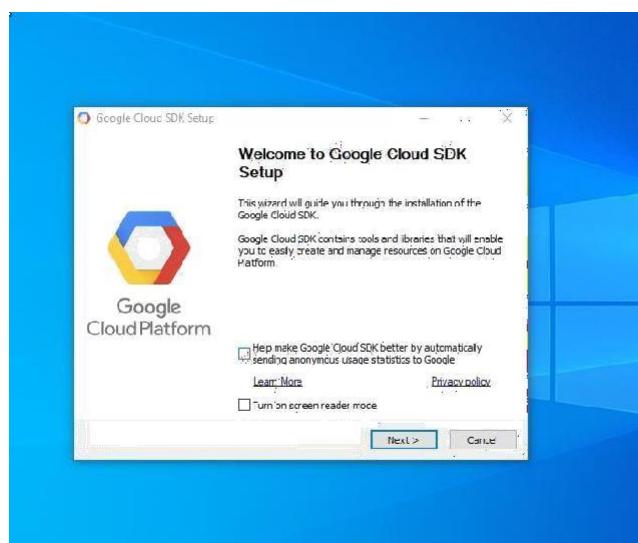
Step2: Select setting up your environment development and click on install and initialize the cloud sdk.



Step3:Download the sdk installer and install it.

The screenshot shows a web browser window with the URL cloud.google.com/sdk/docs/install. The page is titled "Cloud SDK: Command Line Interface" and has a sidebar with links like "cloud SDK", "Quickstarts", "How-to guides", and "Installing the SDK". The main content area is titled "Installation instructions" and provides steps for Linux, Debian/Ubuntu, Red Hat/Fedora/CentOS, macOS, and Windows. The Windows tab is selected, showing PowerShell commands to download the installer and run it. A note at the top says: "★ Note: If you are behind a proxy/firewall, see the [proxy settings](#) page for more information on installation."

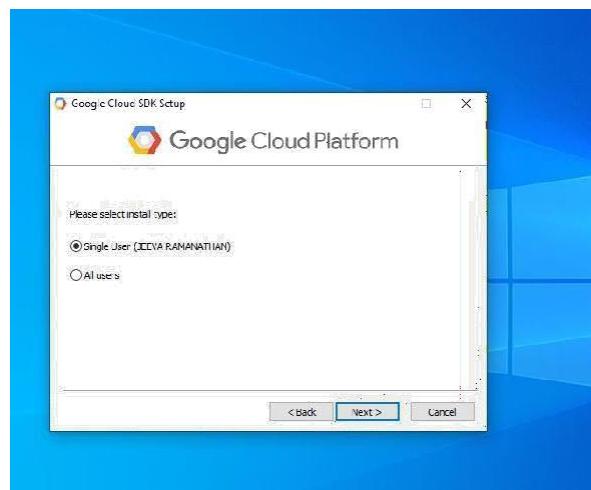
Step4: Click Next.



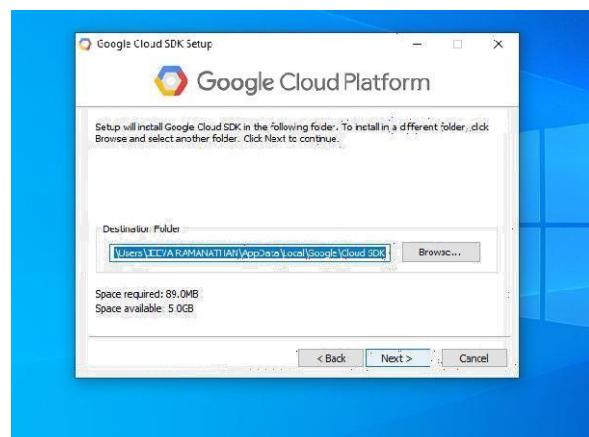
Step5: Click I Agree.



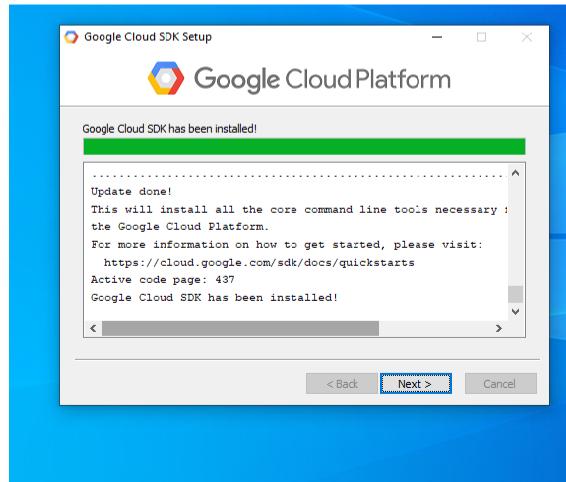
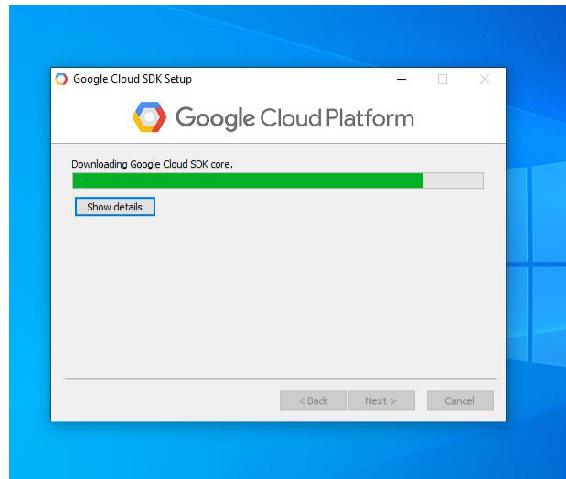
Step6: Select single user and click Next.



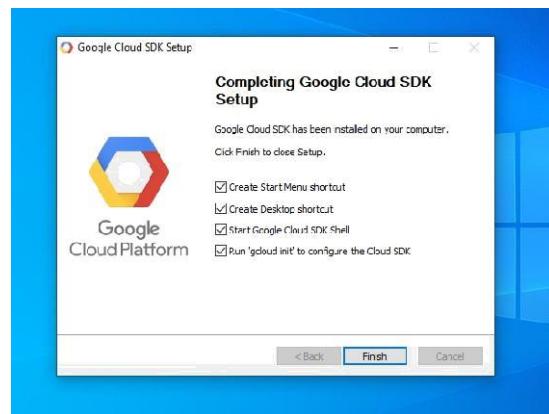
Step7: Select the destination location and click Next.



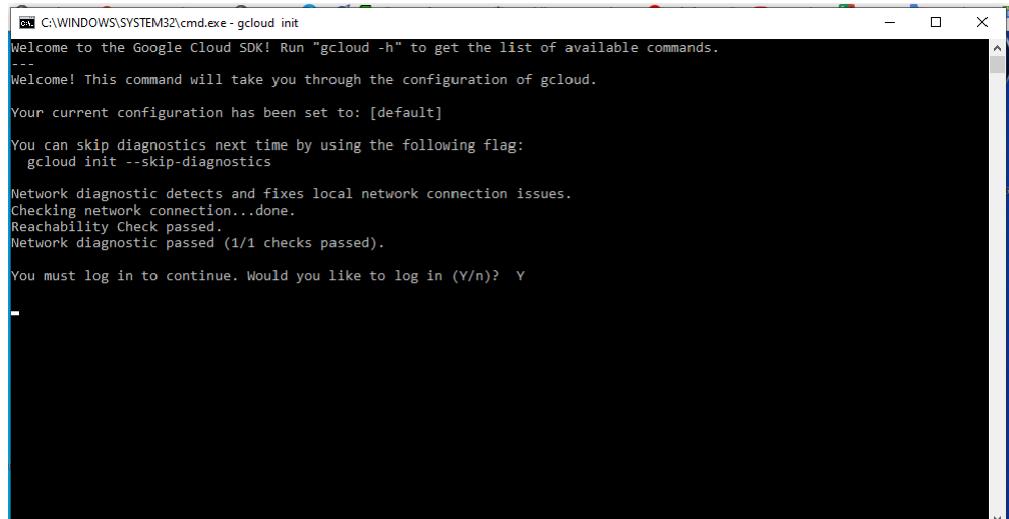
Step8: Downloading all the requirements and installing



Step9: Click Finish.



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



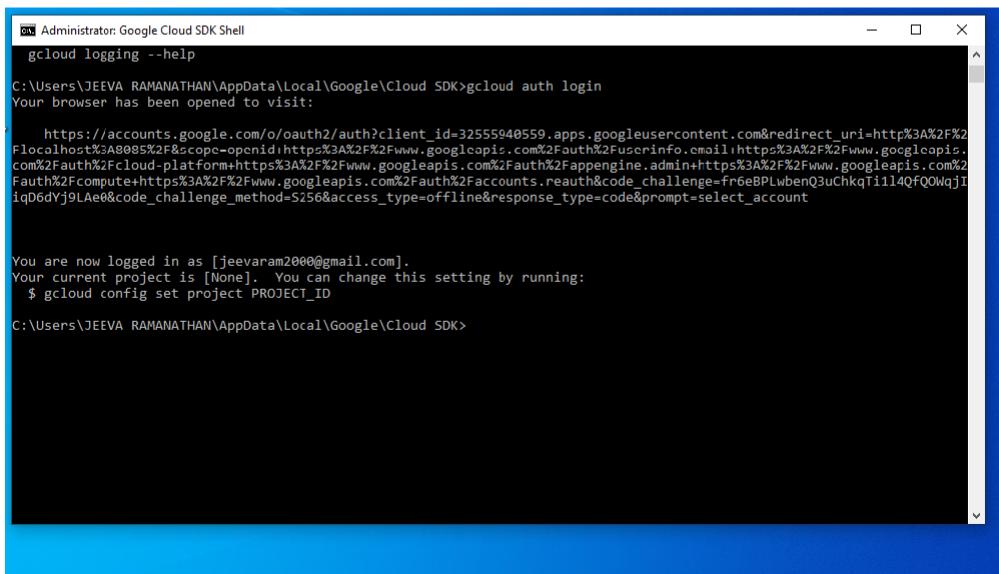
```
C:\WINDOWS\SYSTEM32\cmd.exe - gcloud init
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.
...
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
  Checking network connection...done.
  Reachability Check passed.
  Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)? Y
```



```
Administrator: Google Cloud SDK Shell
gcloud logging --help
C:\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A9085%2F&scope=openid+http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&code_challenge=fr6eBPlwbenQ3uChkqTii14Qf00WqjIiqD6dYj9Lae0&code_challenge_method=s256&access_type=offline&response_type=code&prompt=select_account

You are now logged in as [jeevaram2000@gmail.com].
Your current project is [None]. You can change this setting by running:
$ gcloud config set project PROJECT_ID

C:\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK>
```

Result:

Thus googleapp engine is installed successfully in the system.

EX.No:4

Launch a web application using GAE launcher

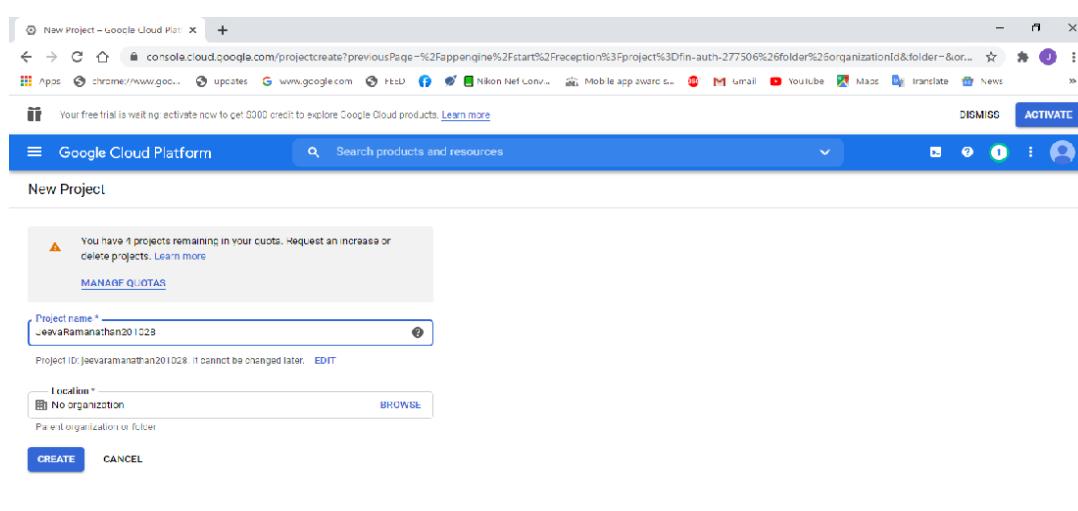
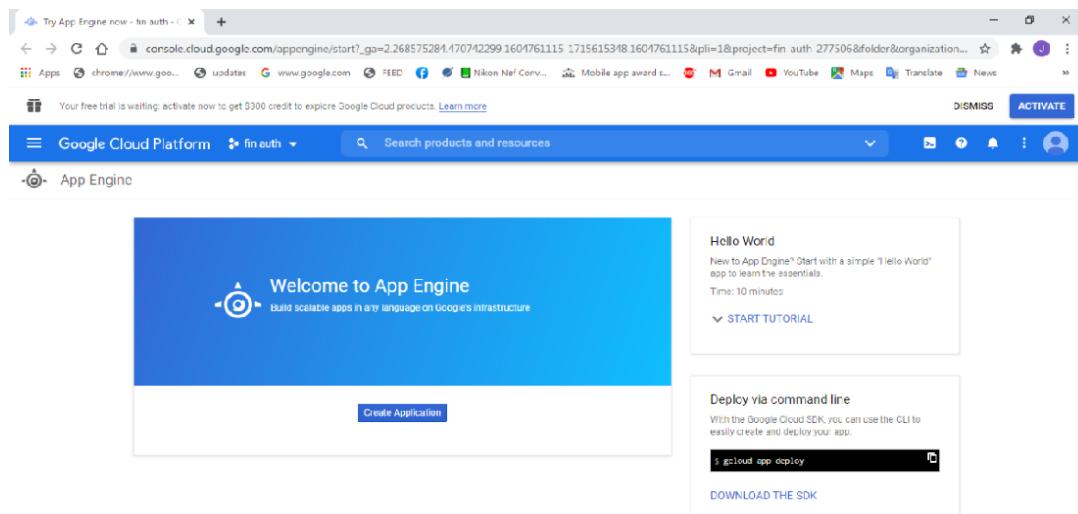
Aim:

To launch the web applications by using the GAE launcher.

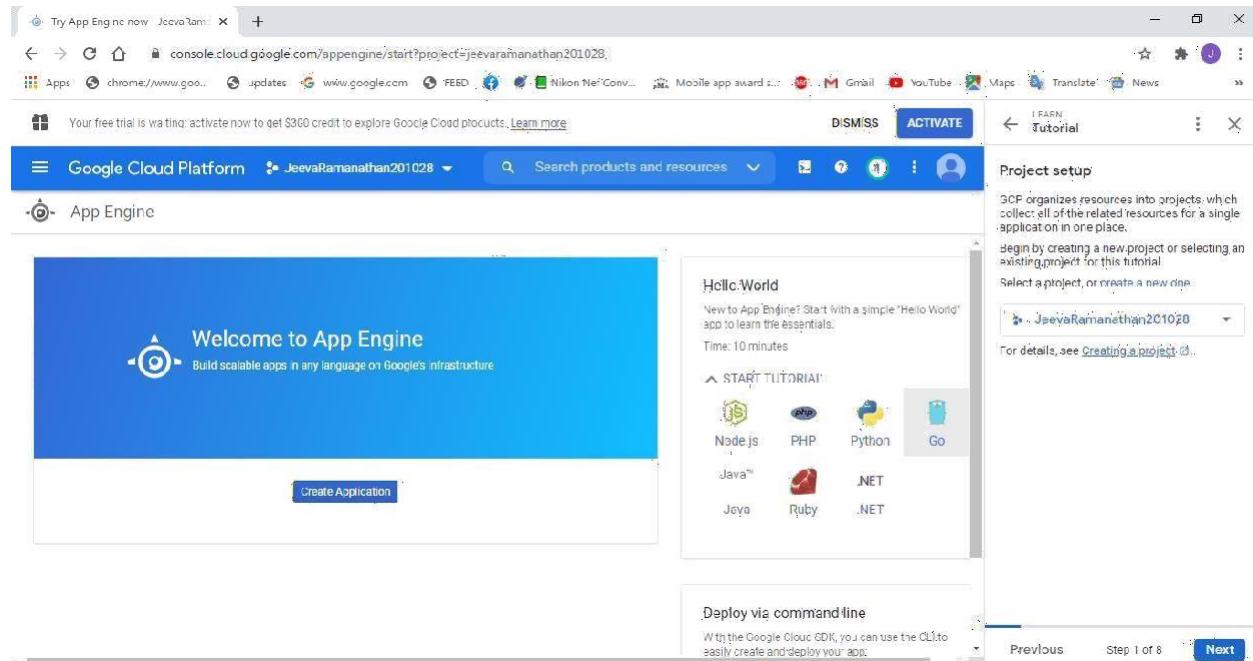
Procedure:

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.

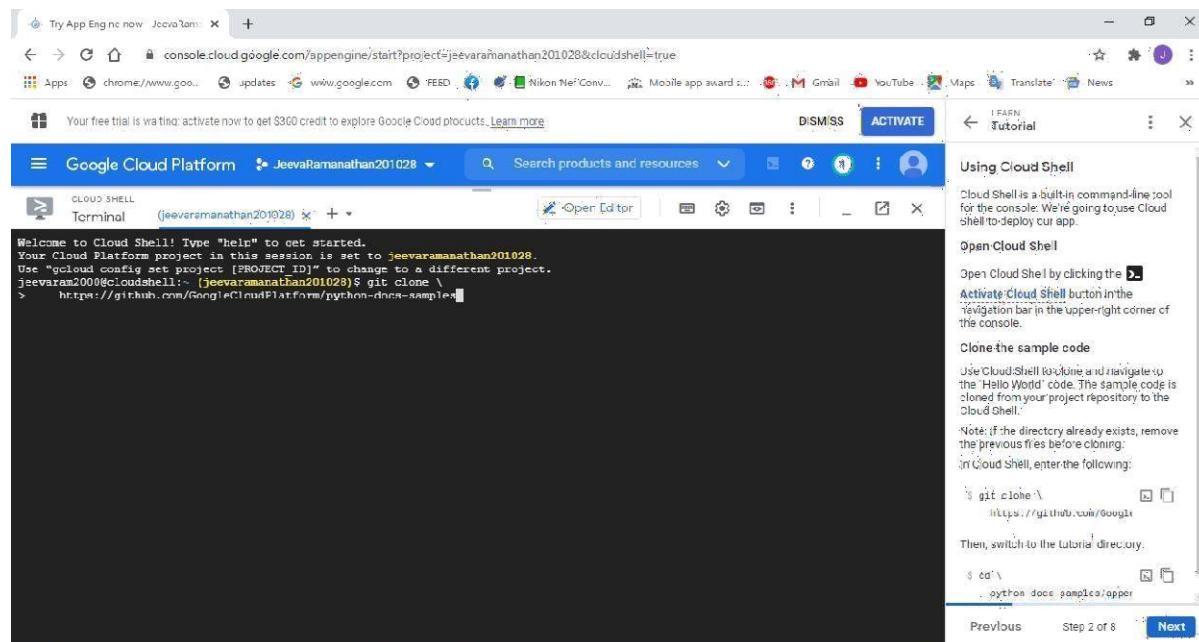


Step2:Select python and click next.



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

The screenshot shows a Google Cloud Platform Cloud Shell terminal window. The terminal output shows the creation of a virtual environment named 'envs/hello_world'. It includes the command to create the environment, the contents of the app.yaml file, and the command to run the application.

```
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# [START gae_python38_app]
from flask import Flask

# If 'entrypoint' is not defined in app.yaml, App Engine will look for an app
# called 'app' in 'main.py'.
app = Flask(__name__)

@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'

if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    # app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python3 \
>     ~envs/hello_world
```

The right side of the interface has a 'Tutorial' sidebar with steps for testing the app on Cloud Shell, activating the virtual environment, using pip to install dependencies, and finally running the app in Cloud Shell.

Step5: Activate your virtual environment.

The screenshot shows a Google Cloud Platform Cloud Shell terminal window. The terminal output shows the activation of the previously created virtual environment 'envs/hello_world'. It includes the command to activate the environment, the creation of a virtual environment named 'CPython3.7.3.final.0-64 in 701ms', and the addition of seed packages.

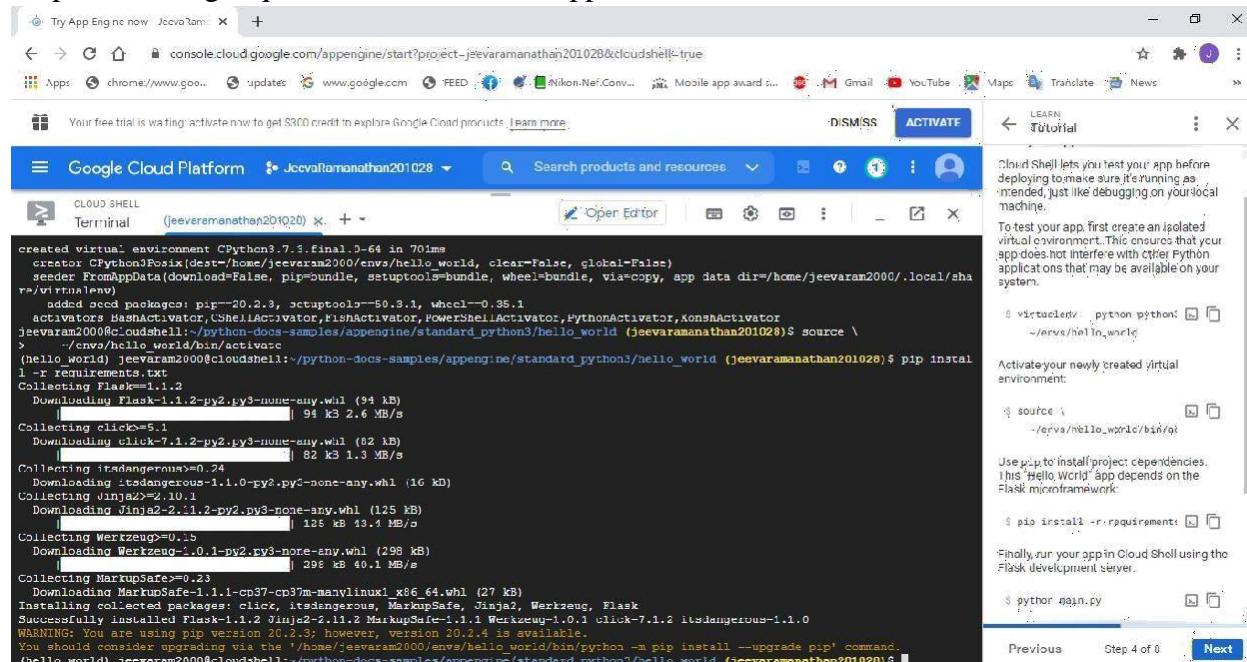
```
# If 'entrypoint' is not defined in app.yaml, App Engine will look for an app
# called 'app' in 'main.py'.
app = Flask(__name__)

@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'

if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    # app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python3 \
>     ~envs/hello_world
created virtual environment CPython3.7.3.final.0-64 in 701ms
  creator CPython3Posix(dest=/home/jeevaraman2000/envs/hello_world, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/jeevaraman2000/.local/share/virtualenvs)
  added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
  activators BashActivator, CShellActivator, FishActivator, PowerShellActivator, PythonActivator, XonshActivator
jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
>     ~envs/hello_world/bin/activate
(hello_world) jeevaraman2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$
```

The right side of the interface has a 'Tutorial' sidebar with steps for testing the app on Cloud Shell, activating the virtual environment, using pip to install dependencies, and finally running the app in Cloud Shell using the Flask development server. A message at the bottom indicates that the code has been copied to the clipboard.

Step6: Installing requirements and run the app.



```

created virtual environment CPython3.7.3.final.0-64 in 701ms
  creator CPython3Posix(dest=/home/jeevaram2000/envs/hello_world, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/jeevaram2000/.local/share/virtualenvs)
  added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
activators BashActivator, CShellActivator, FishActivator, PowerShellActivator, PythonActivator, KonshActivator
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
> ~/envs/hello_world/bin/activate
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip install -r requirements.txt
Collecting Flask<1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
|██████████| 94 kB 2.6 MB/s
Collecting click<5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (62 kB)
|██████████| 62 kB 1.3 MB/s
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2<2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
|██████████| 125 kB 13.4 MB/s
Collecting Werkzeug<0.15
  Downloading Werkzeug-20.0.1-py2.py3-none-any.whl (298 kB)
|██████████| 298 kB 40.1 MB/s
Collecting MarkupSafe<0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-20.0.1 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$

```

Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.

To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.

\$ virtualenv --python python3 ~/envs/hello_world

Activate your newly created virtual environment:

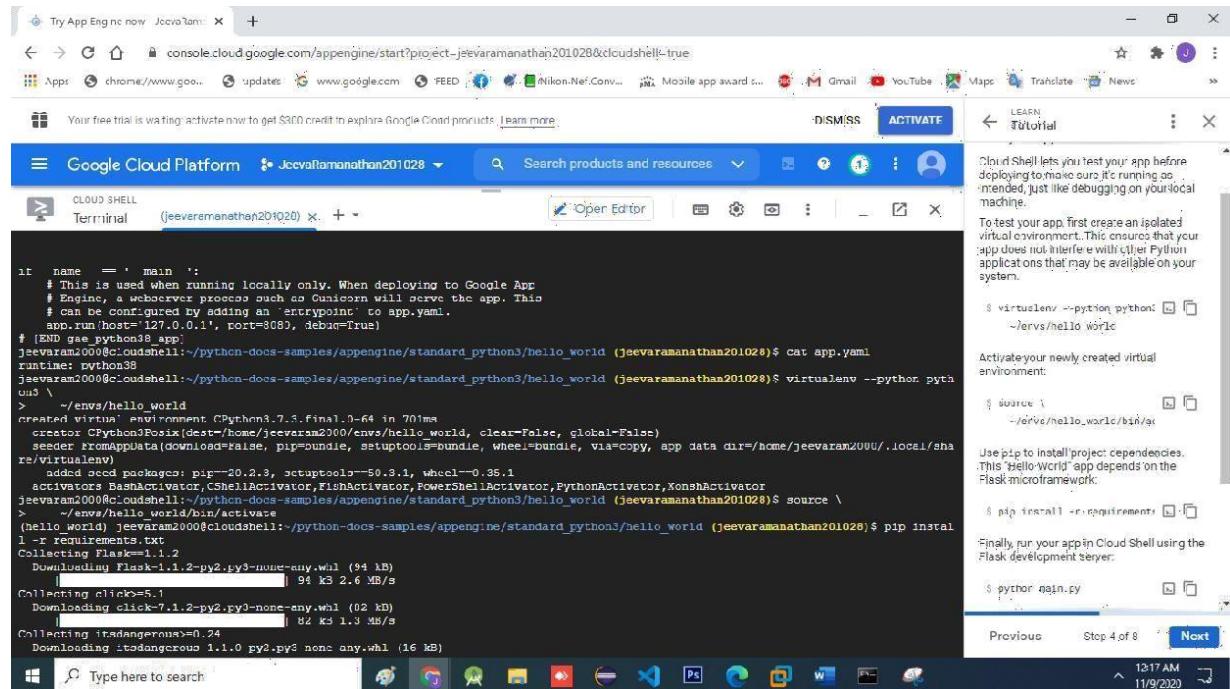
\$ source venvs/hello_world/bin/activate

Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:

\$ pip install -r requirements.txt

Finally, run your app in Cloud Shell using the Flask development server:

\$ python main.py



```

if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Unicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    app.run(host='127.0.0.1', port=8080, debug=True)
# [START app]
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python3 \
> ~/envs/hello_world
created virtual environment CPython3.7.3.final.0-64 in 701ms
  creator CPython3Posix(dest=/home/jeevaram2000/envs/hello_world, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/jeevaram2000/.local/share/virtualenvs)
  added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
activators BashActivator, CShellActivator, FishActivator, PowerShellActivator, PythonActivator, KonshActivator
jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
> ~/envs/hello_world/bin/activate
(hello_world) jeevaram2000@CloudShell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip install -r requirements.txt
Collecting Flask<1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
|██████████| 94 kB 2.6 MB/s
Collecting click<5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (62 kB)
|██████████| 62 kB 1.3 MB/s
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)

```

Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.

To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.

\$ virtualenv --python python3 ~/envs/hello_world

Activate your newly created virtual environment:

\$ source venvs/hello_world/bin/activate

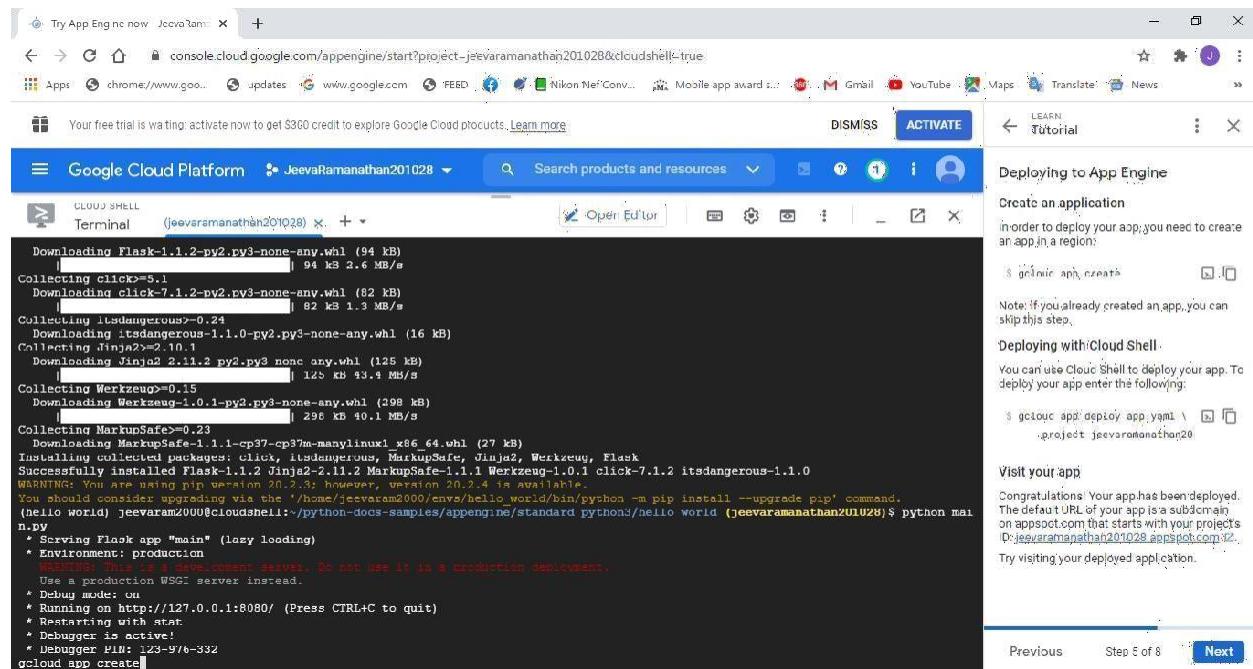
Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:

\$ pip install -r requirements.txt

Finally, run your app in Cloud Shell using the Flask development server:

\$ python main.py

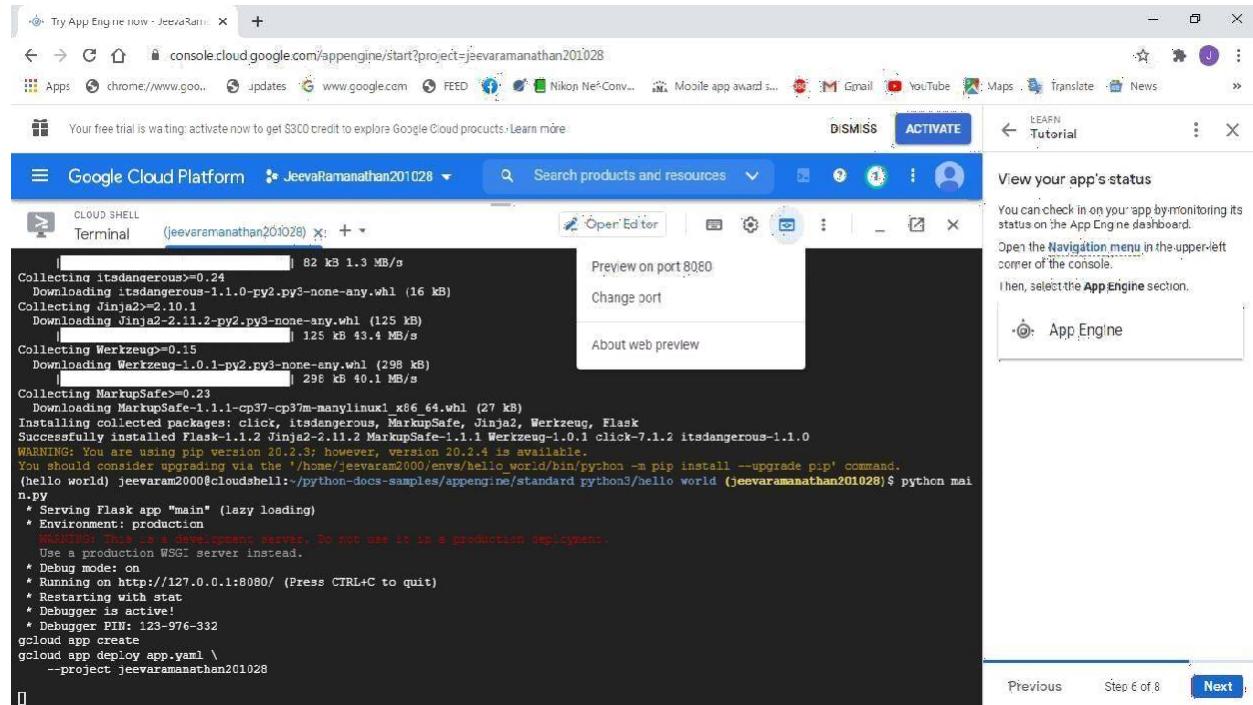
Step7: Create an application and deploy it in cloud shell.



The screenshot shows a Google Cloud Platform Cloud Shell terminal window. The terminal output displays the deployment process for a Python application:

```
Downloading Flask==1.1.2-py2.py3-none-any.whl (94 kB)
Collecting click==5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting itsdangerous==0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2==2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug==0.15
  Downloading Werkzeug-0.1.0-py2.py3-none-any.whl (298 kB)
Collecting MarkupSafe==0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-0.1.0 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(hello-world) jeevaramanathan200@cloudshell:~/python-docs-samples/appengine-standard/python3/hello_world (jeevaramanathan201028)$ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  * READING: This is a development server. Do not use it in a production deployment.
  * Debug mode: on
  * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 123-976-332
gcloud app create
```

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.

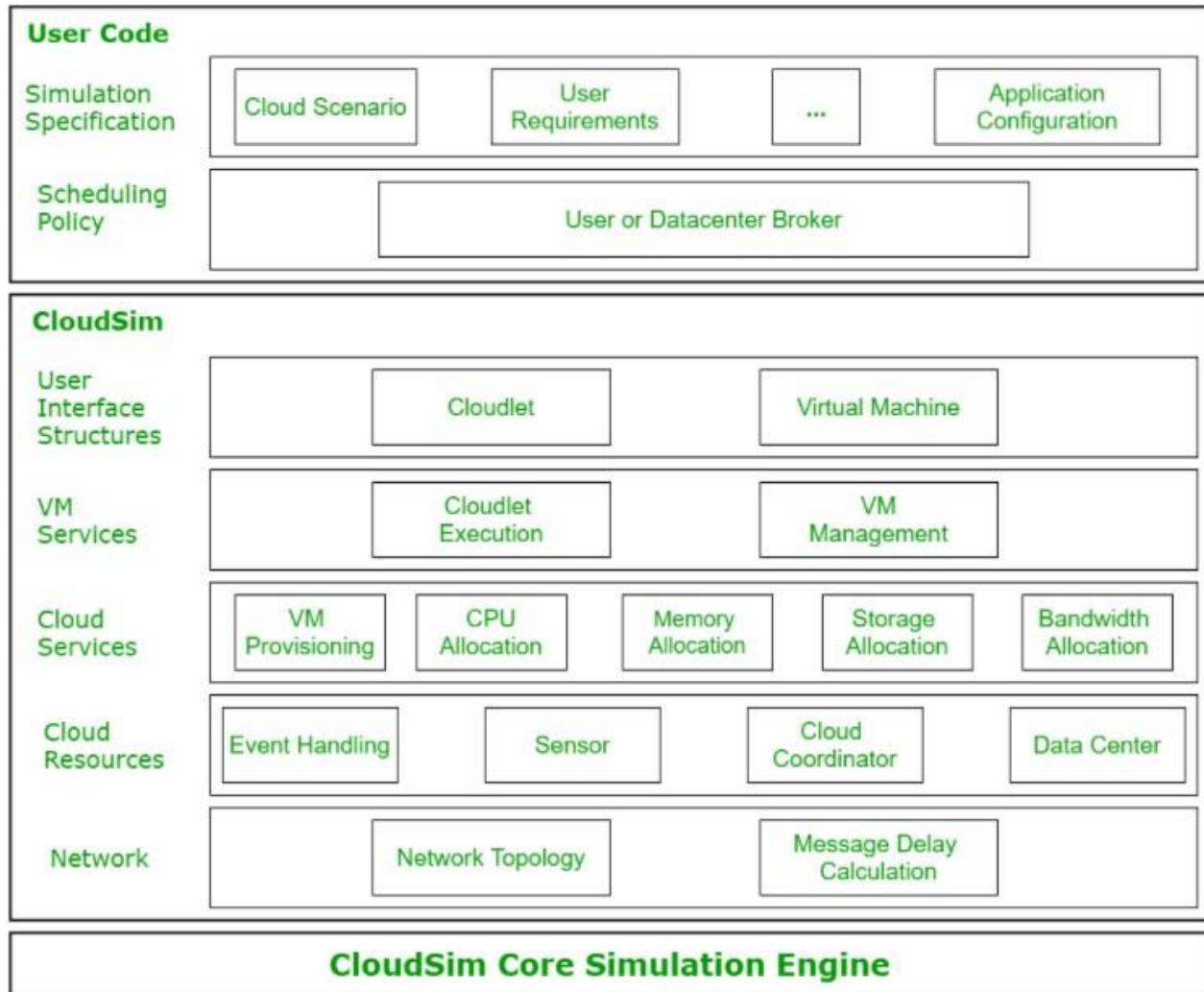
B) Benefits of CloudSim

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



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

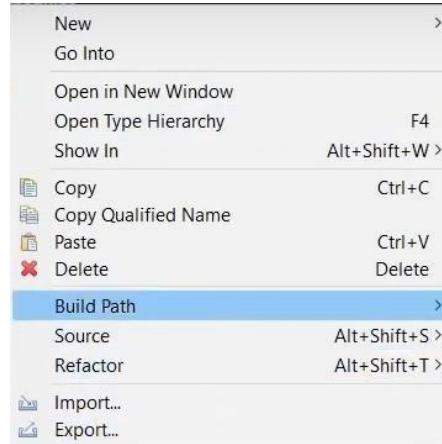
1. SJF stands for Shortest Job First
2. Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.
3. It is a Greedy Algorithm.
4. It may cause starvation if shorter processes keep coming. This problem can be solved using the concept of ageing.
5. It 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

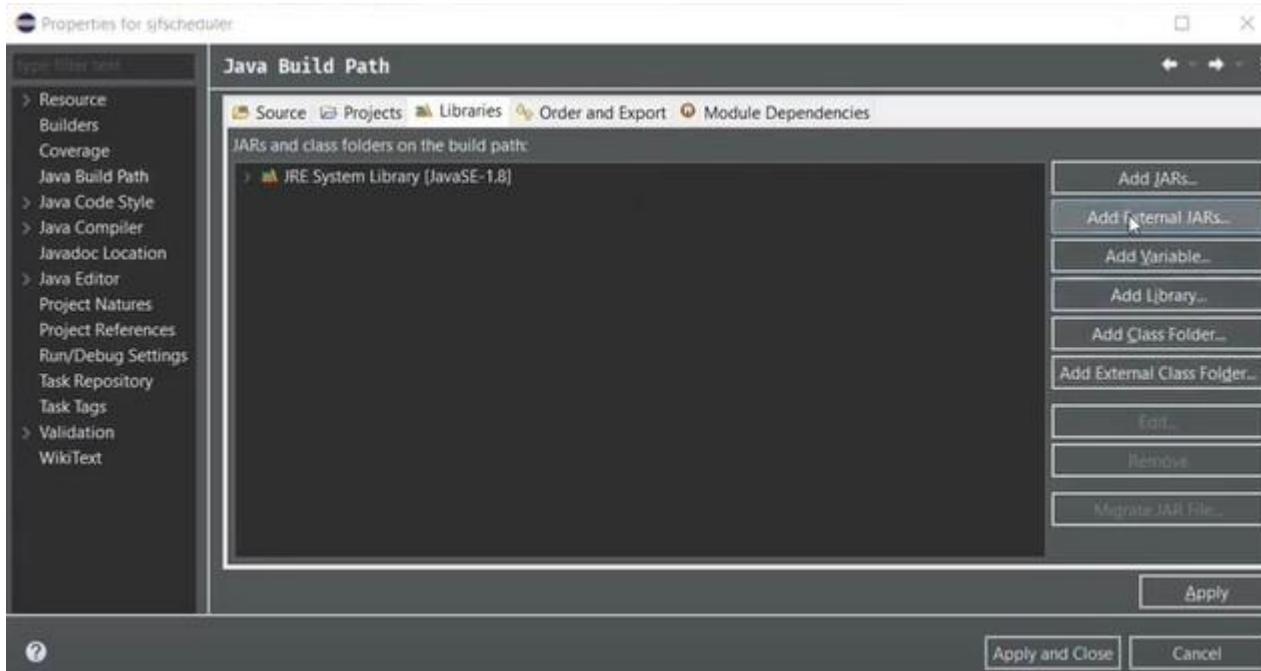
1. Visit <https://github.com/Cloudslab/cloudsim/releases> to download the CloudSim archives for CloudSim 4.
2. Extract the archive.
3. The jars folder of the extracted archive should contain the following files:
 - a) cloudsim-4.0.jar
 - b) cloudsim-examples.jar
4. Create a new Java Project using the 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



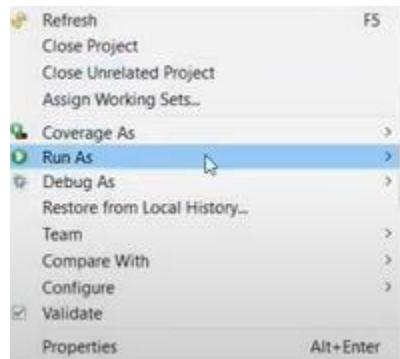
7. Select the Libraries section and click on Add External JARs field on the pop up



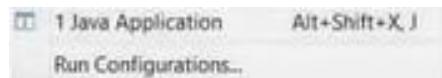
8. Navigate to the jars directory of the CloudSim archive and include the 2 jars in the project.
9. Create a new package in the src directory of the project.
10. Copy the code files for the constants, Data Center Creator, Data Center Broker, Matrix Generator and the SJF scheduler files from the <https://github.com/suyash-more/Cloud-Computing-Projects/tree/master/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...
Initialising 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 I
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.

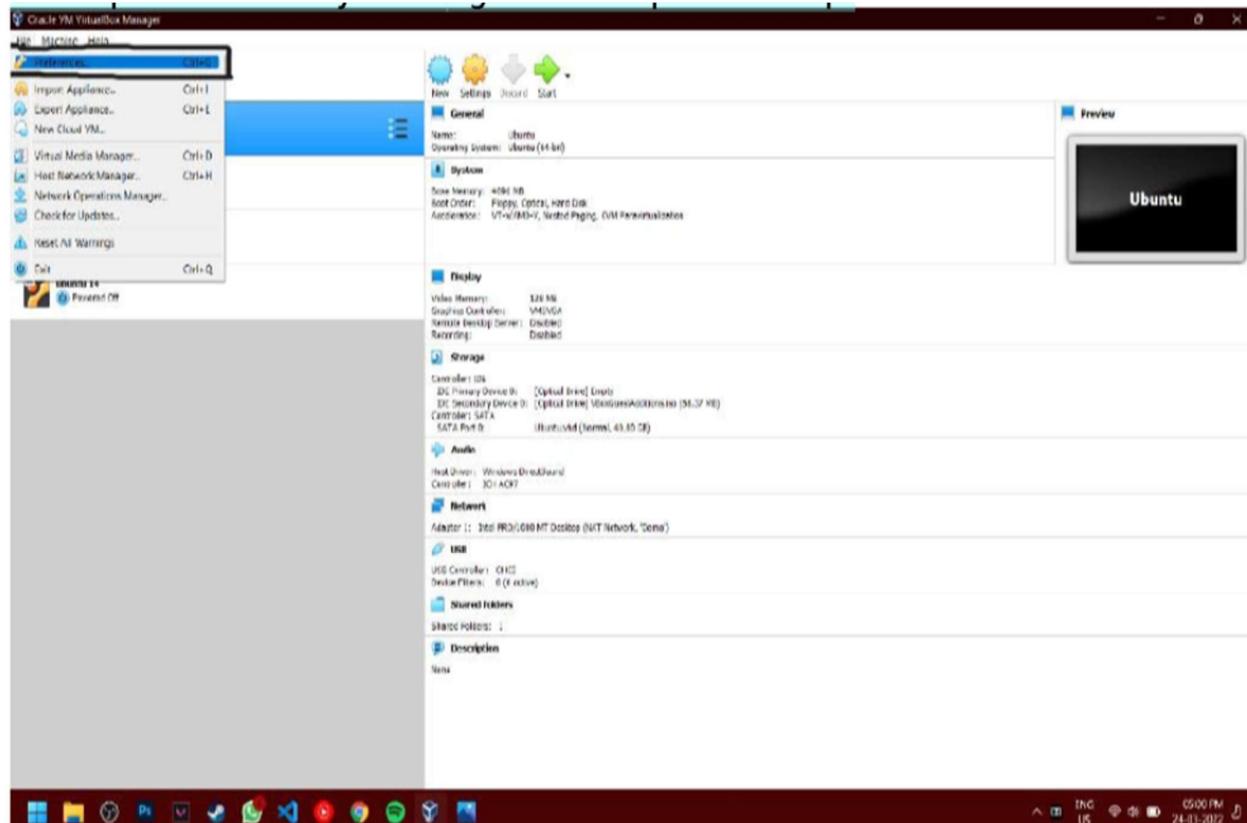
Process Virtual Machine:

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.

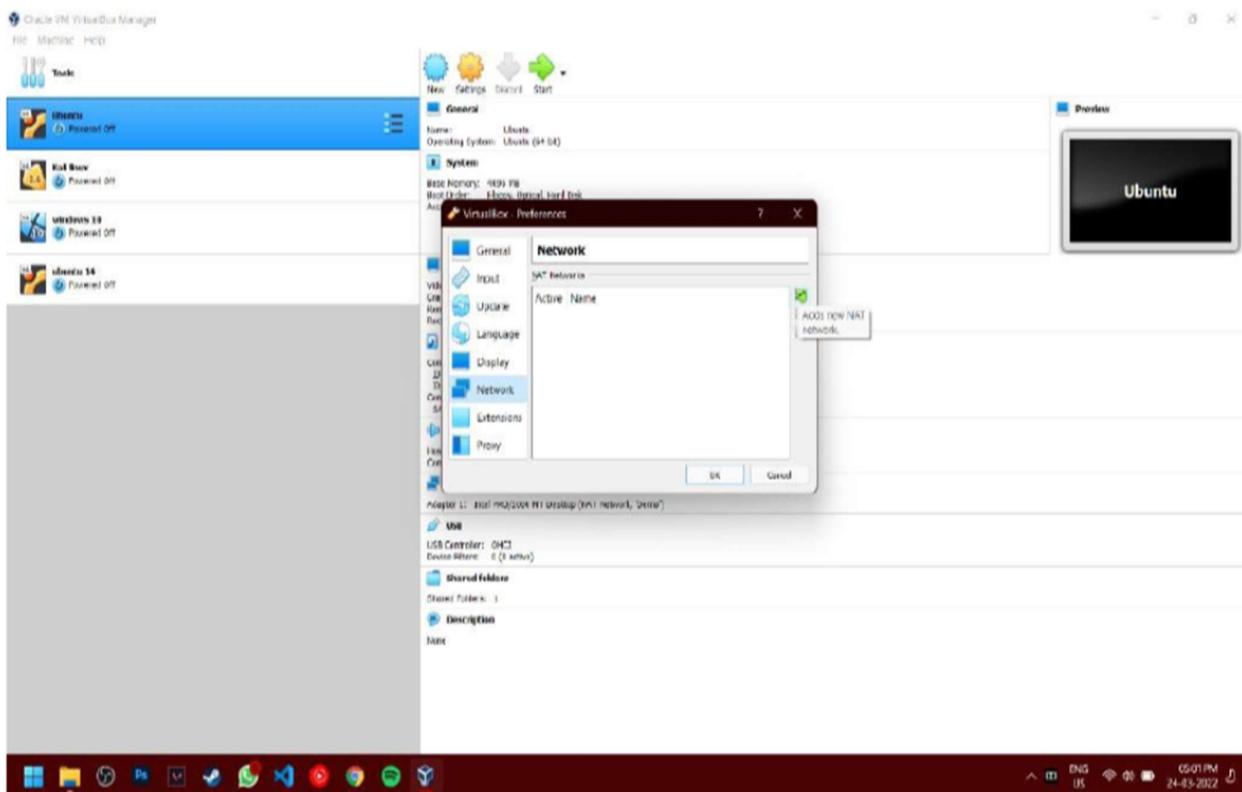
Steps:

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

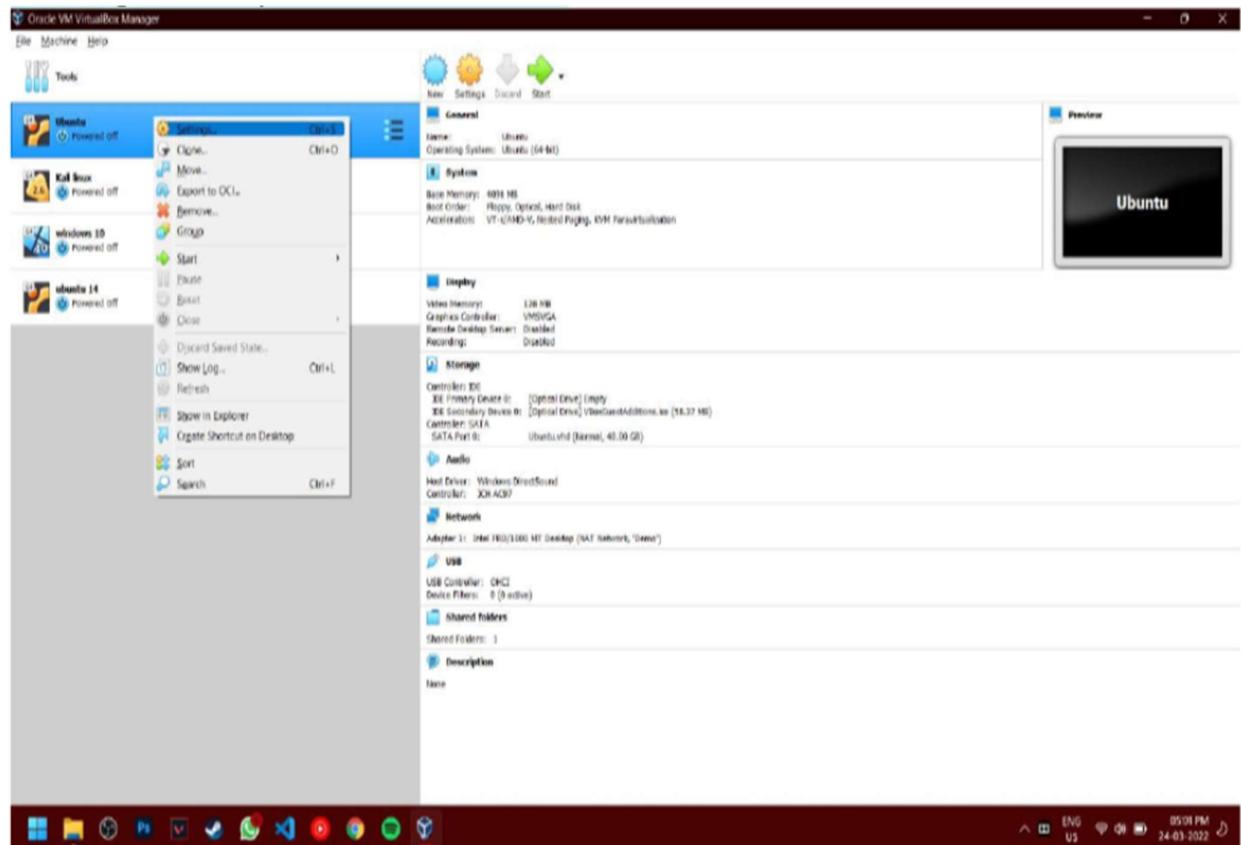
Go to preferences by clicking the File option in Top.



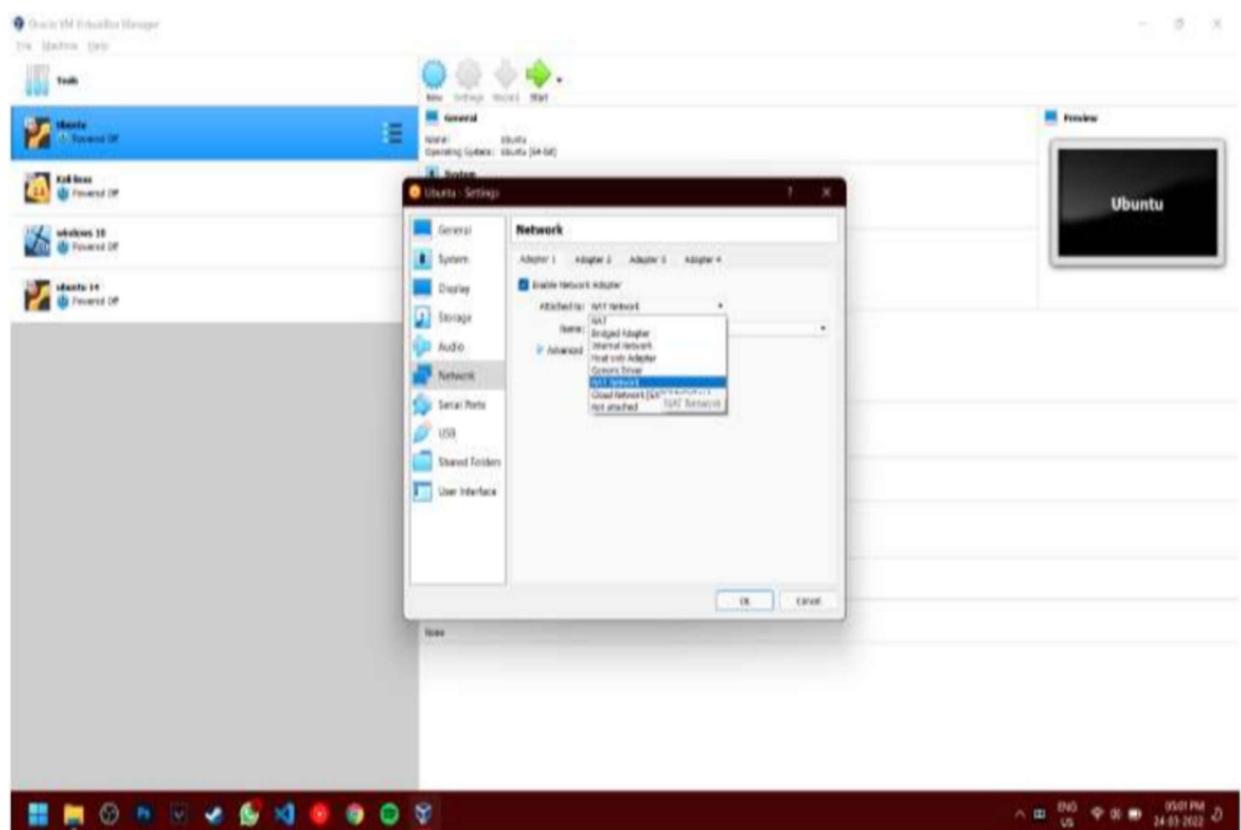
Now select the network option and create a new NAT network

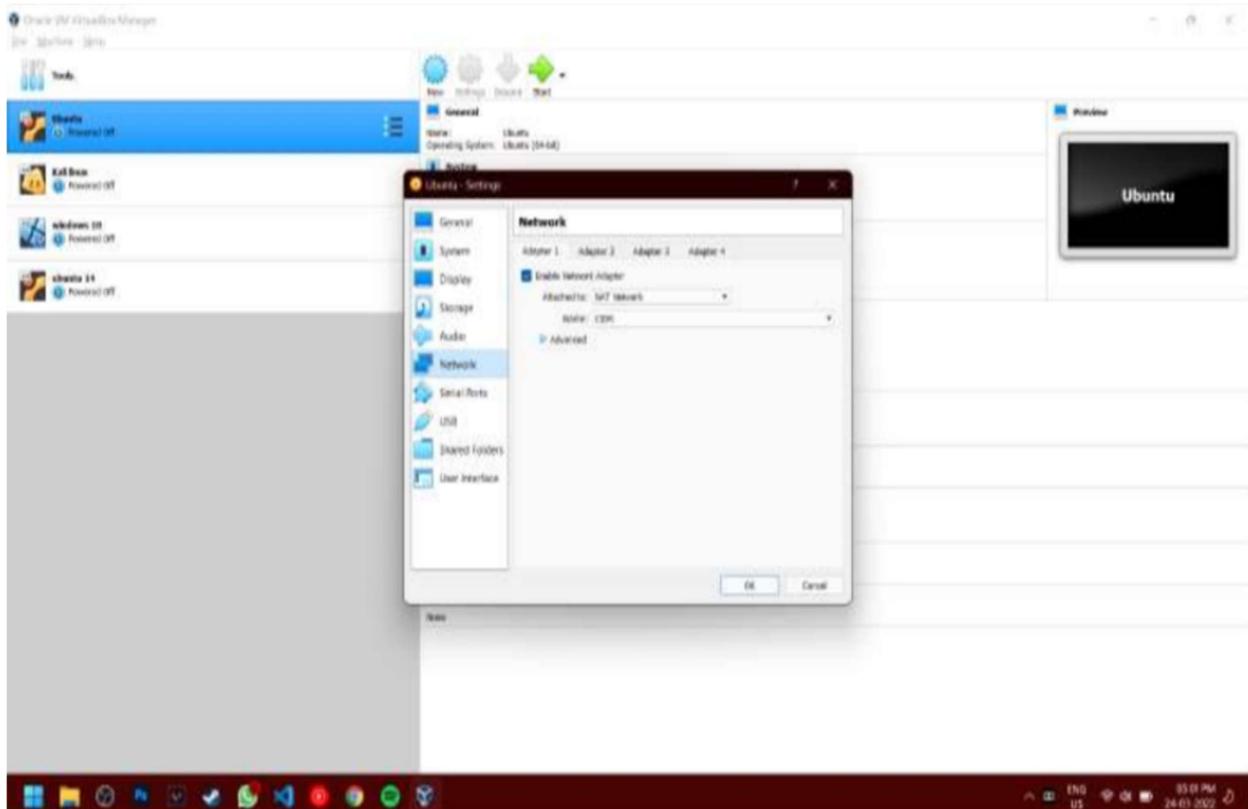


After creating a NAT network, now go to virtual machine setting by right clicking on the preferred machine.



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



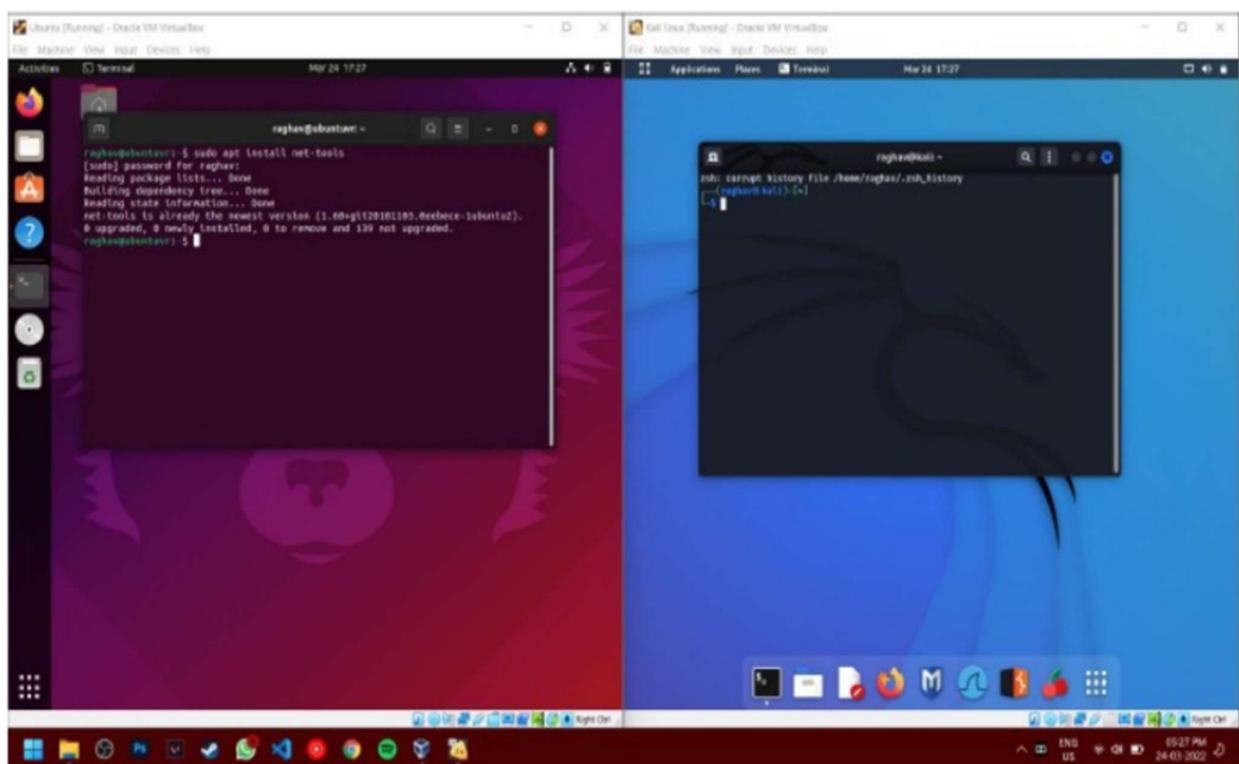


Now repeat the same process for another machine

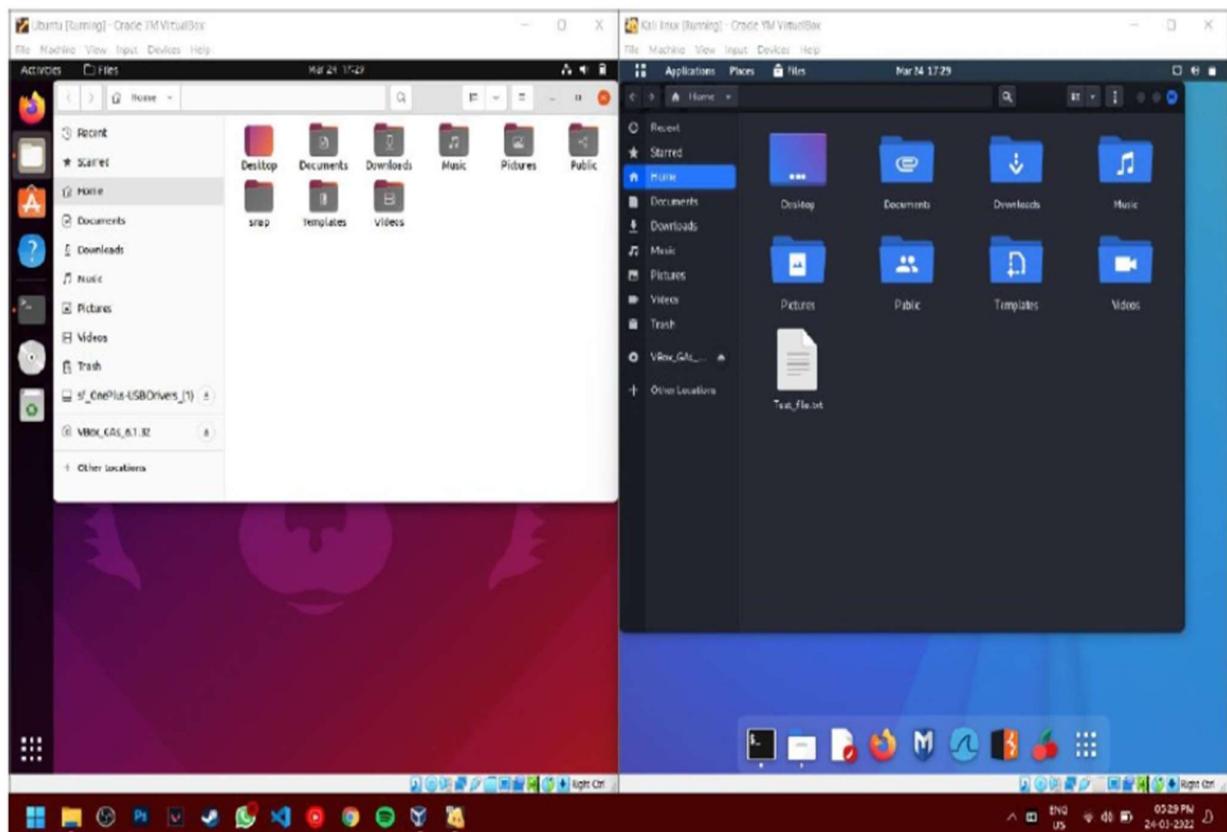
Step 2. Launch both virtual machines

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

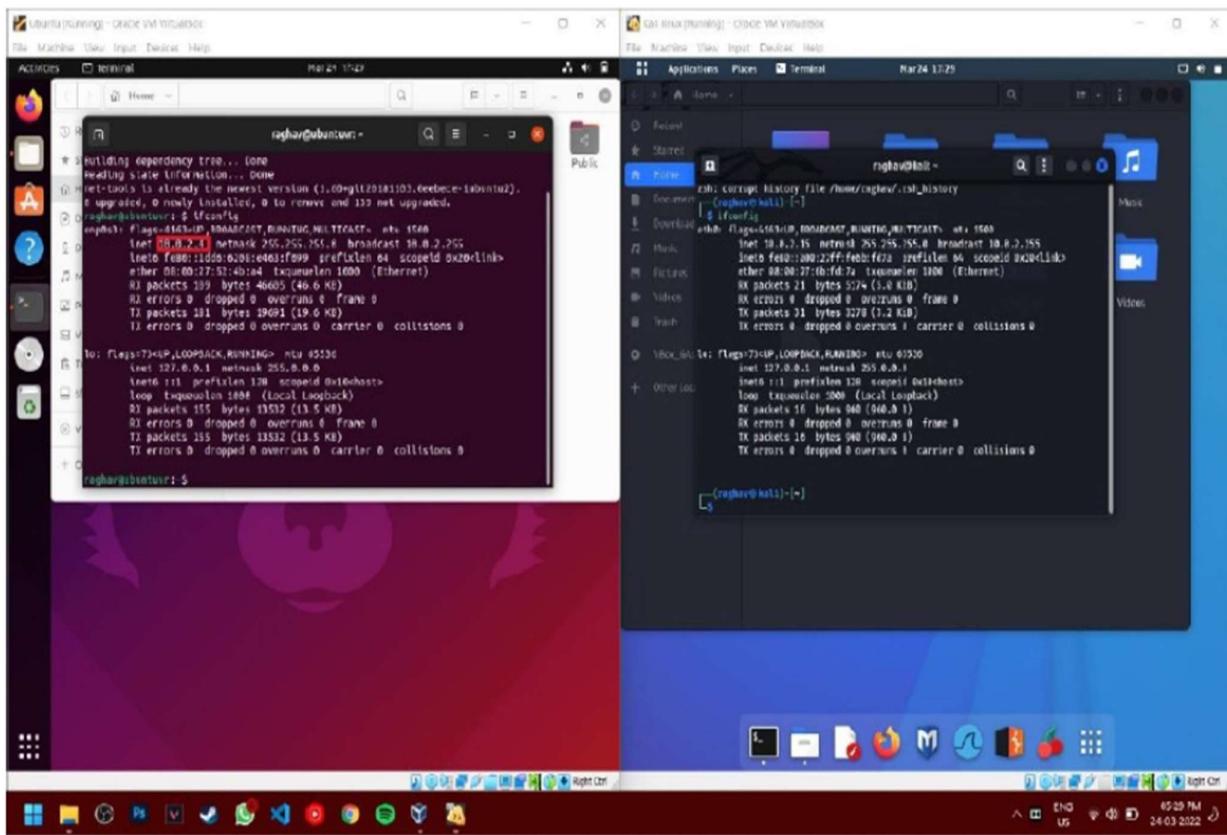
Command: sudo apt install net-tools



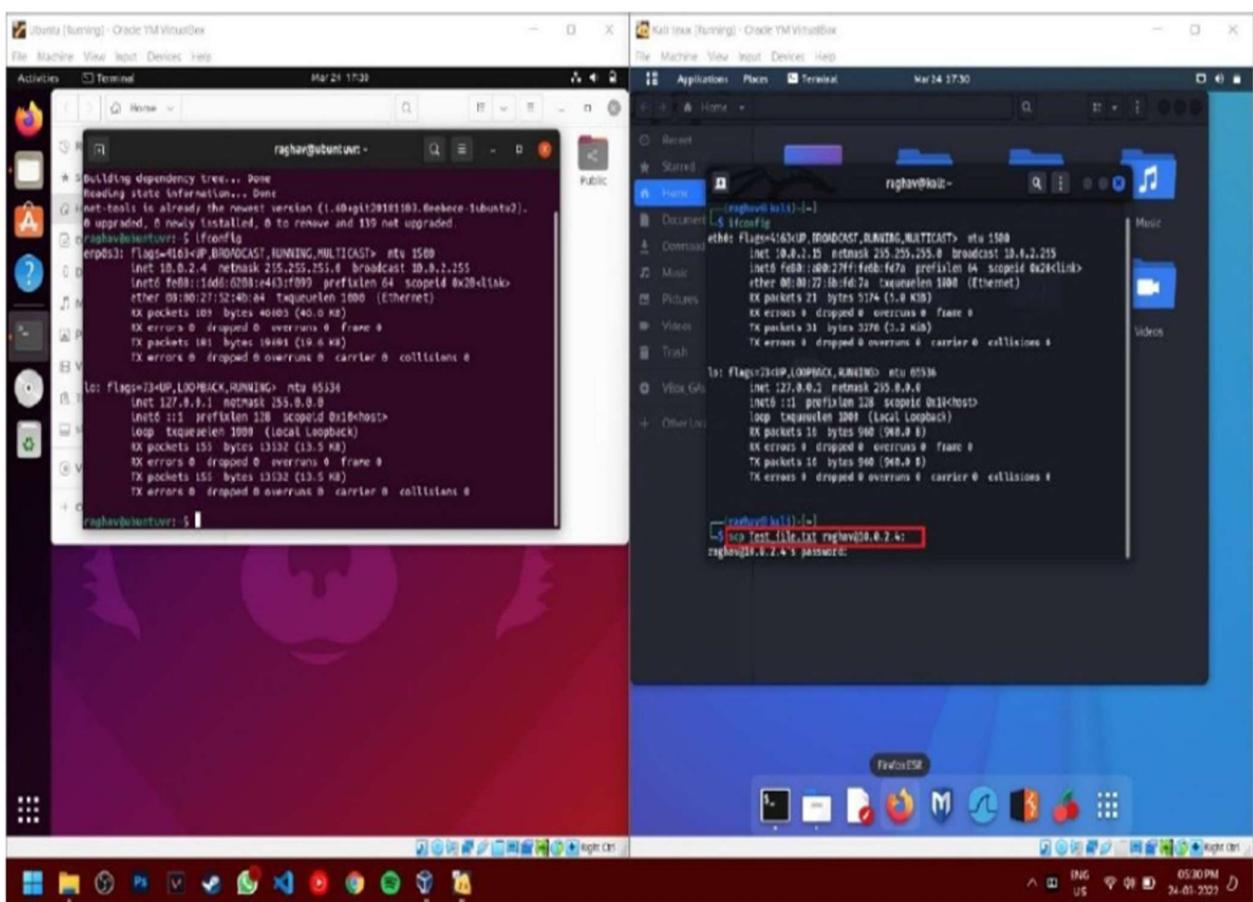
Now create a file in Home folder using any text editor. Here we have used “Test_file.txt” and will transfer from Kali Linux (Right Machine) to Ubuntu (left Machine)



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:

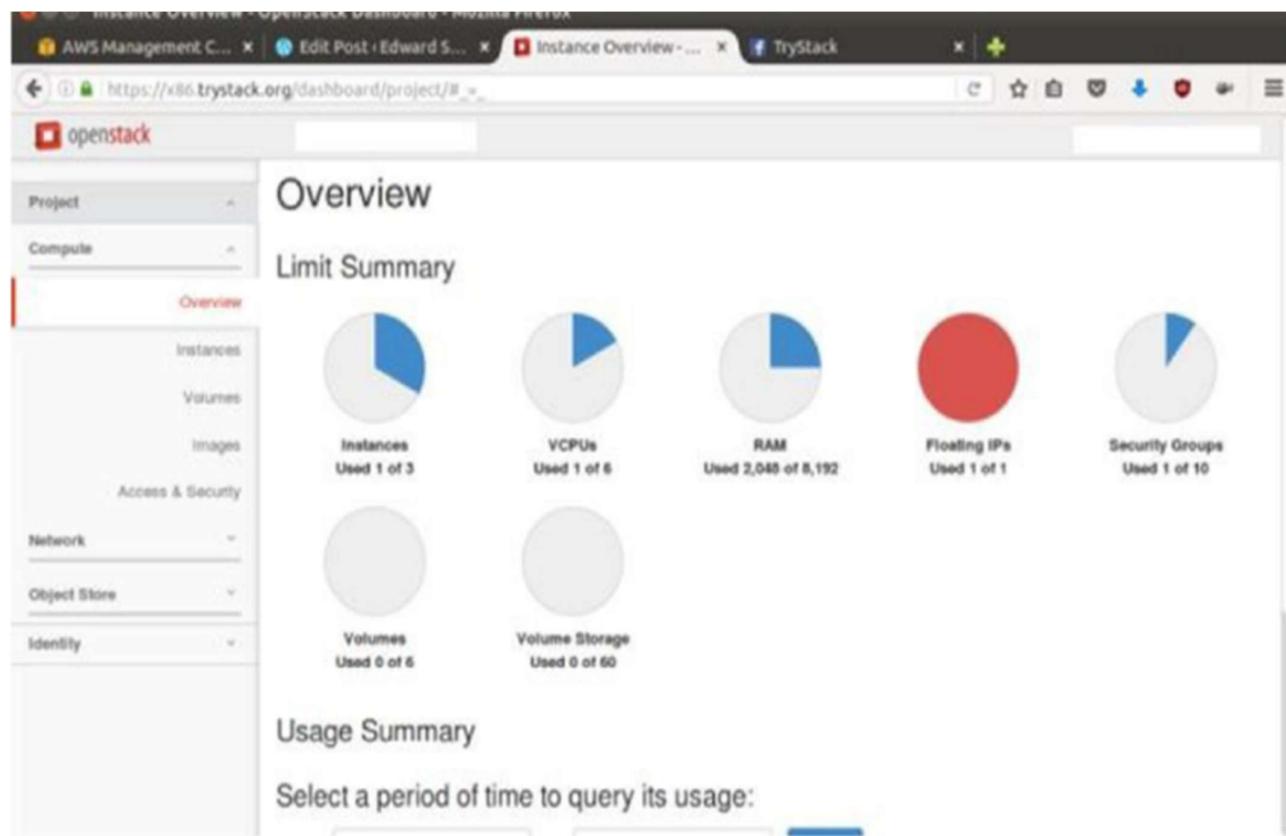
```
./stack.sh
```

Once installation is complete your screen should be :

```
This is your host IP address: [REDACTED]
This is your host IPv6 address: [REDACTED]
Horizon is now available at http://[REDACTED]/dashboard
Keystone is serving at http://[REDACTED]/identity/
The default users are: admin and demo
The password: secret
Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/developer/devstack/systemd.html
2017-06-19 16:13:18.557 | WARNING:
2017-06-19 16:13:18.557 | Using lib/neutron-legacy is deprecated, and it will be removed in the future
2017-06-19 16:13:18.557 | stack.sh completed in 1933 seconds.
stack@devstack:~/devstack$
```

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:

Identity / Projects

Projects

Project	Name	Description	Project ID	Domain Name	Enabled	Actions
al_demo	al_demo		04fe75225674c009fa1a1fb87cd7e4	Default	Yes	Manage Members
demo	demo		329acaa3fb3461bb7288071cf1d988	Default	Yes	Manage Members

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

The screenshot shows the 'Networks' section of a cloud provider's management console. On the left, there's a sidebar with 'Project' dropdown, 'API Access', 'Compute', 'Volumes', 'Network' dropdown (which is expanded to show 'Network Topology', 'Networks' (selected), 'Routers', 'Security Groups', 'Floating IPs', 'Admin', and 'Identity'), and 'Displaying 1 item'. The main area has a breadcrumb 'Project / Network / Networks' with red arrow 1 pointing to the first slash. It shows a table with one item: Name: public, Subnets Associated: IPv6-public-subnet 2001:db8::64, public-subnet 172.24.4.0/24, Shared: No, External: Yes, Status: Active, Admin State: UP, Actions: Edit Network. Red arrow 2 points to the 'Networks' link in the sidebar. Red arrow 3 points to the 'Networks' tab in the main header. Red arrow 4 points to the 'Create Network' button in the top right of the main area.

The screenshot shows the 'Create Network' wizard. The title bar says 'Create Network'. Below it, there are three tabs: 'Network' (selected), 'Subnet', and 'Subnet Details'. The 'Network Name' field contains 'CyberPersons'. A descriptive text on the right says: 'Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.' Below the name field are several checkboxes:

- Enable Admin State
- Shared
- Create Subnet

At the bottom right are 'Cancel', '< Back', and 'Next >' buttons.

Create Network

Network Subnet Subnet Details

Subnet Name
CyberPersons

Network Address Source
Allocate Network Address from a pool

Address pool
shared-default-subnetpool (10.0.0.0/22)

Network Mask
26 (pool default)

IP Version
IPv4

Gateway IP (Optional)

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel Back Next

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.

Create Network

Network Subnet Subnet Details

Enable DHCP

Allocation Pools (Optional)

DNS Name Servers (Optional)

Host Routes (Optional)

Specify additional attributes for the subnet.

Cancel Back Create

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 “Launch Instance”.

The screenshot shows the OpenStack dashboard interface. At the top, there is a header with the OpenStack logo, the project name "alt_demo", and a user icon labeled "admin". Below the header, there is a navigation bar with dropdown menus for "Project", "API Access", "Compute", and "Instances". The "Compute" menu is expanded, showing "Instances" as the selected option. A red arrow labeled "1" points to the "Project" dropdown. A red arrow labeled "2" points to the "Instances" option in the "Compute" dropdown. A red arrow labeled "3" points to the "Instances" tab in the main content area. A red arrow labeled "4" points to the "Launch Instance" button at the bottom right of the content area. The main content area displays an "Overview" section with tabs for "Instances", "Images", "Key Pairs", "Volumes", and "Network". The "Instances" tab is selected, showing a table with columns: Instance Name, Image Name, IP Address, Flavor, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. The table currently displays "No items to display".

The screenshot shows the "Launch Instance" dialog box. The left sidebar lists several tabs: "Details" (selected), "Source", "Flavor", "Networks", "Network Ports", "Security Groups", "Key Pair", "Configuration", "Server Groups", "Scheduler Hints", and "Metadata". The main content area has fields for "Instance Name" (set to "CyberPersons"), "Availability Zone" (set to "nova"), and "Count" (set to "1"). To the right of these fields is a circular progress bar labeled "Total Instances (10 Max)" with "10%" completed. A legend below the progress bar indicates "0 Current Usage", "1 Added", and "9 Remaining". At the bottom of the dialog box are buttons for "Cancel", "Back", "Next", and "Launch 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.

Launch Instance

Details

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

Instance Name * CyberPersons

Availability Zone nova

Total Instances (10 Max) 10%
0 Current Usage
1 Added
9 Remaining

Count * 1

Source *
Flavor *
Networks
Network Ports
Security Groups
Key Pair
Configuration
Server Groups
Scheduler Hints
Metadata

< Back Next > Launch Instance

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.

Launch Instance

Source *	Select Boot Source	Create New Volume			
Flavor *	Image	Yes No			
Networks:	Volume Size (GB) *	Delete Volume on Instance Delete			
Network Ports	1	Yes No			
Security Groups	Allocated				
	Name	Updated	Size	Type	Visibility
	Select an item from Available items below				
	Available				
	Q Click here for filters.				
	Name	Updated	Size	Type	Visibility
	ubuntu	6/22/17 3:17 AM	829.00 MB	iso	Public 1
	cirros-0.3.5-x86_64-disk	6/22/17 2:14 AM	12.65 MB	qcow2	Public 2
	Select one				
Key Pair					
Configuration					
Server Groups					
Scheduler Hints					
Metadata					
<input type="button" value="Cancel"/> <input type="button" value="Back"/> <input type="button" value="Next >"/> <input type="button" value="Launch 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

Launch Instance

Details	Flavors manage the sizing for the compute, memory and storage capacity of the instance.						
Source	Allocated						
Flavor	Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
	m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
Networks	Available						
Network Ports	Select one						
Security Groups	Available						
Key Pair							
Configuration							
Server Groups							
Scheduler Hints							
Metadata							
<input type="button" value="Cancel"/> <input type="button" value="Back"/> <input type="button" value="Next >"/> <input type="button" value="Launch Instance"/>							

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:

The screenshot shows the 'Launch Instance' wizard with the 'Networks' tab selected. The 'Allocated' section shows one network named 'CyberPersons' associated with the 'CyberPersons' subnet, marked as 'No' for shared status, 'Up' for admin state, and 'Active' for status. The 'Available' section is empty, with a note to select at least one network. Other tabs visible include 'Details', 'Source', 'Flavor', 'Network Ports', 'Security Groups', 'Key Pair', and 'Configuration'.

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:

Displaying 1 item											Actions
	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	CyberPersons	-	10.0.0.75	m1.tiny	-	Build	■ nova	Block Device Mapping	No State	0 minutes	<button>Associate Floating IP</button> ▾

Displaying 1 item

Step 5: Access Virtual Machine Console!

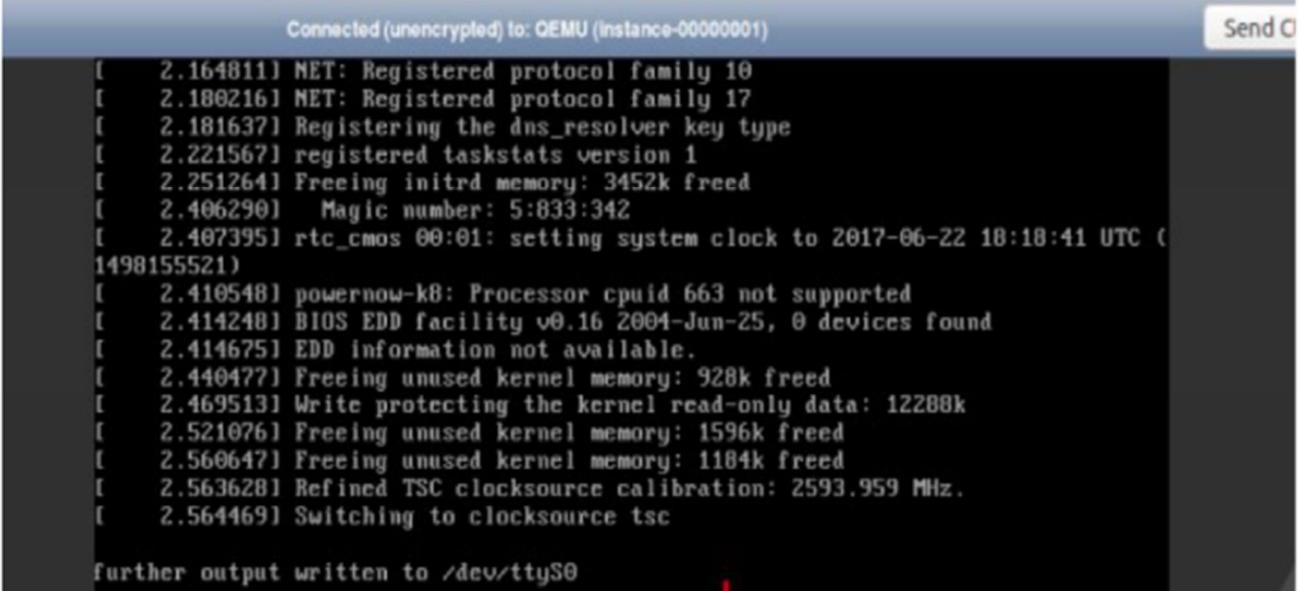
Once you click “Launch Instance” it will take OpenStack few seconds to create your virtual machine. Once ready you can access the console to see how the command line of your first virtual machine looks like.

Displaying 1 item											Actions
	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	CyberPersons	-	172.24.4.11 2001:db8::6	m1.tiny	-	Active	■ nova	None	Running	3 minutes	<button>Create Snapshot</button> ▾

Displaying 1 item

- Associate Floating IP
- Attach Interface
- Detach Interface
- Edit Instance
- Attach Volume
- Detach Volume
- Update Metadata
- Edit Security Groups
- Console →
- View Log
- Pause Instance
- Suspend Instance
- Shelve Instance
- Resize Instance
- Lock Instance
- Soft Reboot Instance
- Hard Reboot Instance
- Shut Off Instance

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



The screenshot shows a terminal window titled "Connected (unencrypted) to: QEMU (Instance-00000001)". The window contains a large amount of text representing the kernel boot process. The text is in white on a black background and includes various system initialization messages such as network registration, DNS resolver key type, taskstats version, memory freeing, BIOS EDD facility, and TSC clocksource calibration. A red vertical bar is visible on the right side of the terminal window.

```
Connected (unencrypted) to: QEMU (Instance-00000001)
[    2.164811] NET: Registered protocol family 10
[    2.180216] NET: Registered protocol family 17
[    2.181637] Registering the dns_resolver key type
[    2.221567] registered taskstats version 1
[    2.251264] Freeing initrd memory: 3452k freed
[    2.406290] Magic number: 5:833:342
[    2.407395] rtc_cmos 00:01: setting system clock to 2017-06-22 18:18:41 UTC (1498155521)
[    2.410548] powernow-k8: Processor cpuid 663 not supported
[    2.414240] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[    2.414675] EDD information not available.
[    2.440477] Freeing unused kernel memory: 928k freed
[    2.469513] Write protecting the kernel read-only data: 12288k
[    2.521076] Freeing unused kernel memory: 1596k freed
[    2.560647] Freeing unused kernel memory: 1184k freed
[    2.563628] Refined TSC clocksource calibration: 2593.959 MHz.
[    2.564469] Switching to clocksource tsc

further output written to /dev/ttyS0
```

RESULT:

The procedure for finding 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.

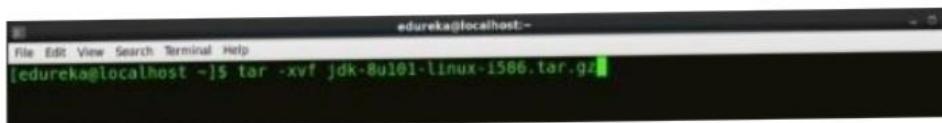
Steps:

Install Hadoop

Step1: [Click here](#) to download the Java 8 Package. Save this file in your home directory.

Step2: Extract the JavaTarFile.

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

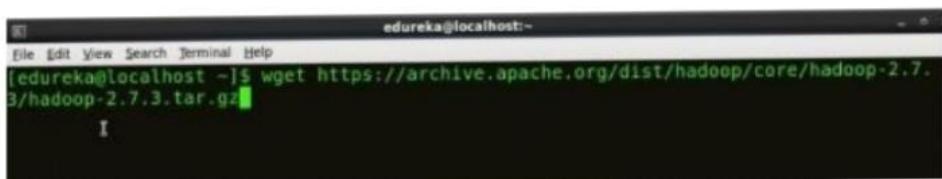


```
edureka@localhost:~$ tar -xvf jdk-8u101-linux-i586.tar.gz
```

Fig:Hadoop Installation–Extracting Java Files

Step3: Download the Hadoop 2.7.3 Package.

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

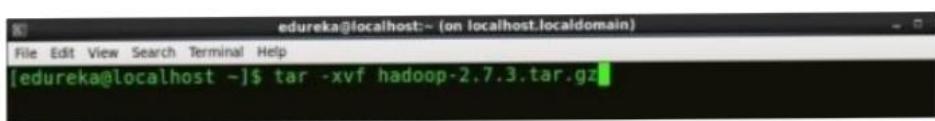


```
edureka@localhost:~$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation– Downloading Hadoop

Step4: Extract the Hadoop tar File.

Command: tar -xvf hadoop-2.7.3.tar.gz



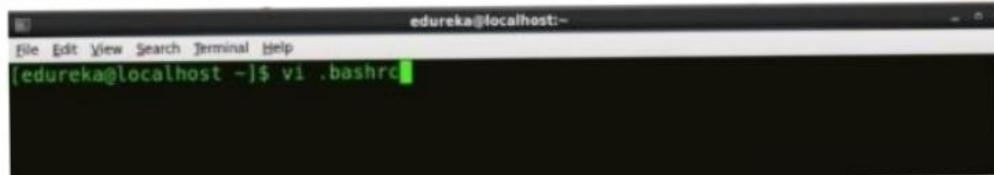
```
edureka@localhost:~$ tar -xvf hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Extracting Hadoop Files Step 5:

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

Now, add Hadoop and Java Path as shown below.

Command: vi.bashrc



```
[edureka@localhost ~]$ vi .bashrc
```

```
# User specific aliases and functions

export HADOOP_HOME=$HOME/hadoop-2.7.3
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME/hadoop-2.7.3
export HADOOP_COMMON_HOME=$HADOOP_HOME/hadoop-2.7.3
export HADOOP_HDFS_HOME=$HADOOP_HOME/hadoop-2.7.3
export YARN_HOME=$HADOOP_HOME/hadoop-2.7.3
export PATH=$PATH:$HADOOP_HOME/hadoop-2.7.3/bin

# Set JAVA_HOME

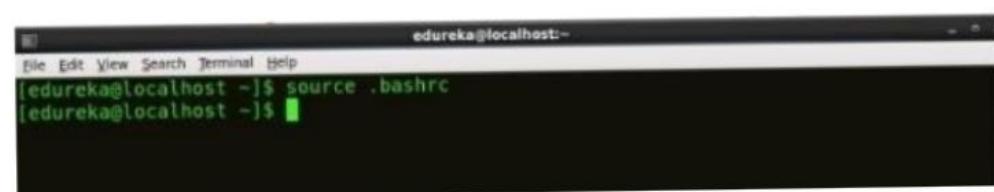
export JAVA_HOME=/home/edureka/jdk1.8.0_101
export PATH=/home/edureka/jdk1.8.0_101/bin:$PATH
```

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



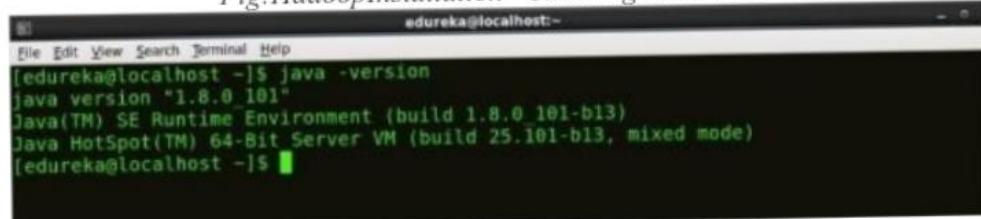
```
[edureka@localhost ~]$ source .bashrc
[edureka@localhost ~]$
```

Fig:Hadoop Installation–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.

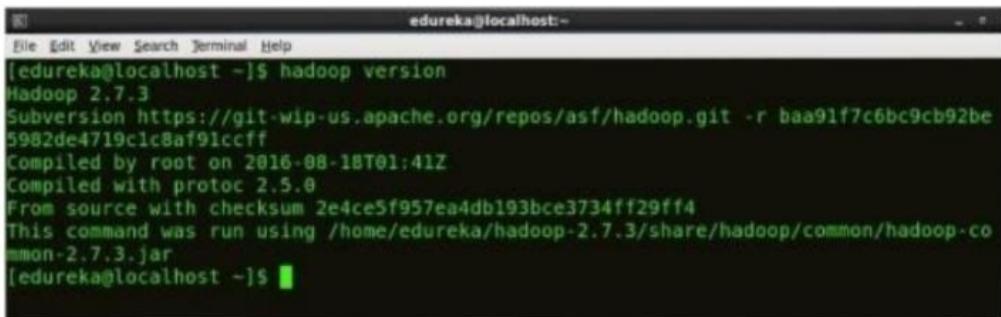
Command: java-version

Fig:Hadoop Installation– Checking Java Version



```
[edureka@localhost ~]$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
[edureka@localhost ~]$
```

Command: hadoop version



```
[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-common-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:



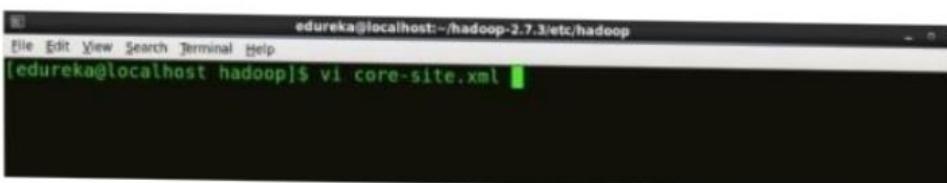
```
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh          mapred-env.sh
configuration.xml           httpfs-log4j.properties   mapred-queues.xml.template
container-executor.cfg       httpfs-signature.secret  mapred-site.xml.template
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
```

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: vi*core-site.xml*



```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

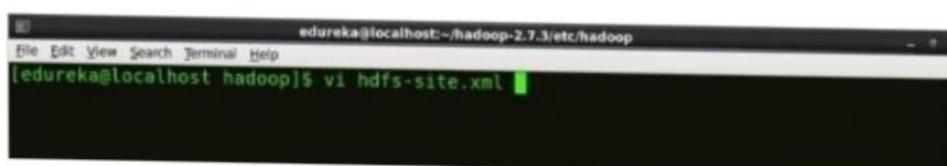
Fig:Hadoop Installation– Configuring core-site.xml

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

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

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: vi*hdfs-site.xml*



```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
```

Fig:Hadoop Installation– Configuring hdfs-site.xml

```
1      <?xmlversion="1.0"encoding="UTF-8"?>
2  <?xmlstylesheet type="text/xsl" href="configuration.xsl"?>
3      <configuration>
4          <property>
5              <name>dfs.replication</name>
6              <value>1</value>
7          </property>
8          <property>
9              <name>dfs.permission</name>
10             <value>false</value>
11         </property>
12     </configuration>
```

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

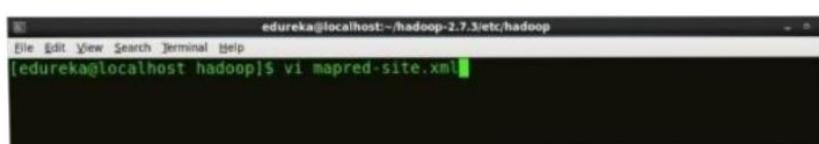
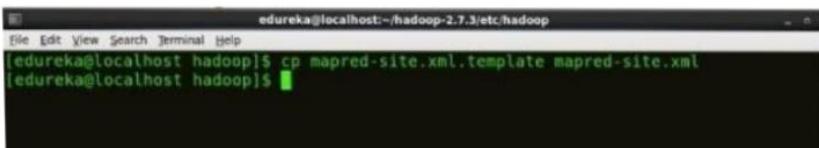
Inside configuration tag:

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: cp*mapred-site.xml.template* *mapred-site.xml*

Command:vim*mapred-site.xml*.



```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

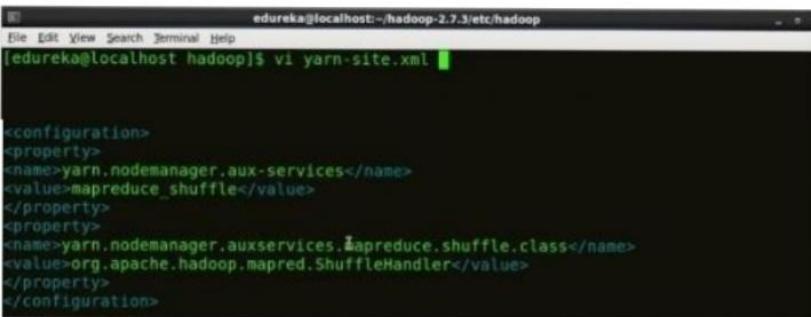
Fig:Hadoop Installation– Configuring mapred-site.xml

```
1           <?xmlversion="1.0"encoding="UTF-8"?>
2             <xmlestsheettype="text/xsl" href="configuration.xsl"?>
3               <configuration>
4                 <property>
5                   <name>mapreduce.framework.name</name>
6                     <value>yarn</value>
7                   </property>
8               </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: vi *yarn-site.xml*



```
<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>
</configuration>
```

Fig:HadoopInstallation– Configuring *yarn-site.xml*

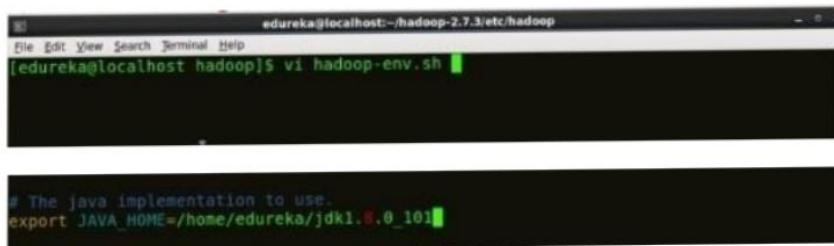
Step11: Edit *hadoop-env.sh* and add the JavaPath as mentioned below:

```
1
2           <?xmlversion="1.0">
3             <configuration>
4               <property>
5                 <name>yarn.nodemanager.aux-services</name>
6                   <value>mapreduce_shuffle</value>
7                 </property>
8               <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</
9                 name>
10                 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
11               </property>
12
13
```

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



```
# The java implementation to use.  
export JAVA_HOME=/home/edureka/jdk1.8.0_101
```

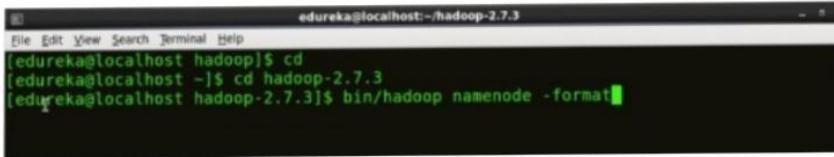
Fig:HadoopInstallation–Configuringhadoop-env.sh**Step 12:**

Go to Hadoop home directory and format the Name Node.

Command: cd

Command: cd hadoop-2.7.3

Command: bin/hadoop namenode -format



```
[edureka@localhost hadoop]$ cd  
[edureka@localhost ~]$ cd hadoop-2.7.3  
[edureka@localhost hadoop-2.7.3]$ bin/hadoop namenode -format
```

Fig:HadoopInstallation– Formatting NameNode

This formats the HDFS via Name Node. This command is only executed for the first time. 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

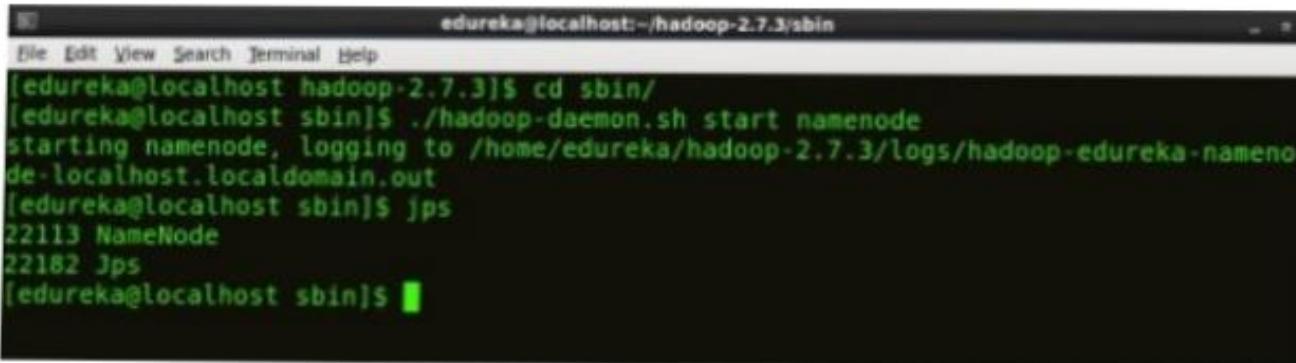
The above command is a combination of *start-dfs.sh*, *start-yarn.sh* & *mr-jobhistory-daemon.sh*

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



A screenshot of a terminal window titled "edureka@localhost:~/hadoop-2.7.3/sbin". The window shows the following command sequence:

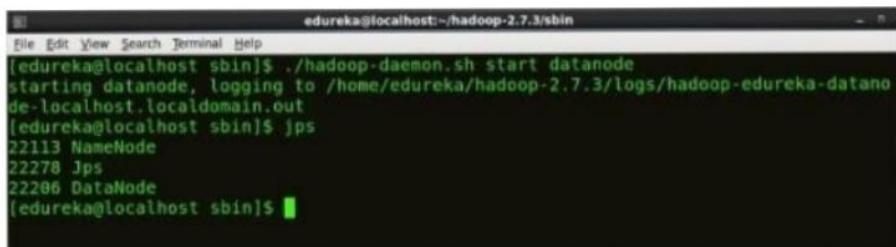
```
[edureka@localhost hadoop-2.7.3]$ cd sbin/  
[edureka@localhost sbin]$ ./hadoop-daemon.sh start namenode  
starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-namenode-localhost.localdomain.out  
[edureka@localhost sbin]$ jps  
22113 NameNode  
22182 Jps  
[edureka@localhost sbin]$ █
```

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



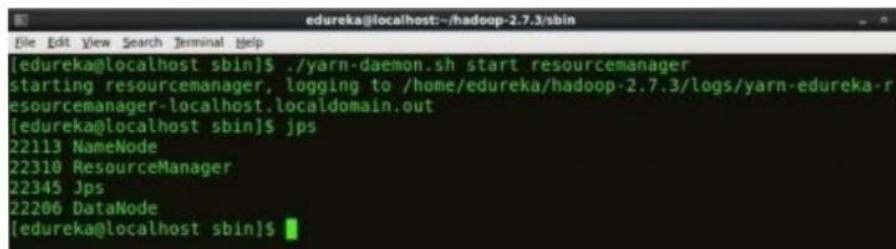
```
edureka@localhost sbin$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datanode-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278 Jps
22206 DataNode
[edureka@localhost sbin]$
```

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.sh start resourcemanager



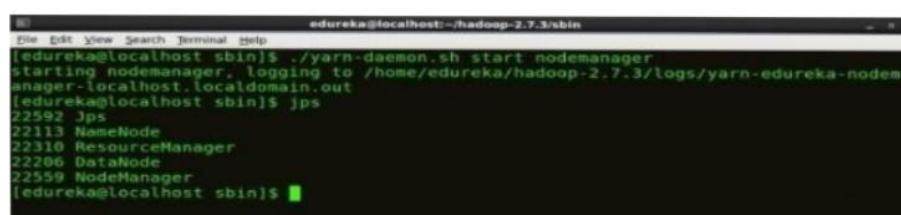
```
edureka@localhost sbin$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-resourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22345 Jps
22206 DataNode
[edureka@localhost sbin]$
```

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



```
edureka@localhost sbin$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-nodemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22592 Jps
22113 NameNode
22310 ResourceManager
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```



[See BatchDetails](#)

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 Hadoop services are up and running, run the below command.

Command:jps

```
File Edit View Search Terminal Help edureka@localhost:~/hadoop-2.7.3/sbin
[edureka@localhost sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/edureka/hadoop-2.7.3/logs/mapred-edureka-h
istoryserver-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22694 JobHistoryServer
22727 Jps
22296 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```

Fig:Hadoop Installation– Checking Daemons

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

The screenshot shows a Mozilla Firefox window titled "Namenode Information - Mozilla Firefox". The address bar displays the URL "http://localhost:50070/dfshealth.html#tab-overview". The main content area is titled "Overview 'localhost:9000' (active)". Below this, there is a table with the following data:

Started:	Wed Nov 02 08:32:45 CET 2016
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-617e6b4f-a7e8-45ee-abae-e59744b38d66
Block Pool ID:	BP-1874109370-127.0.0.1-1477077288629

Fig: Hadoop Installation–Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster

Result:

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

locs_headercta_contactus)

topic_url=https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html



Get started
(#)

Service guides
(#)

Developer tools
(#)



Documentation
(<https://docs.aws.amazon.com/index.html>)

> Amazon RDS
(<https://docs.aws.amazon.com/rds/index.html>)

> User Guide

Create an Amazon RDS DB instance

[PDF \(/pdfs/AmazonRDS/latest/UserGuide/rds-ug.pdf#CHAP_Tutorials.WebServerDB.CreateDBInstance\)](#)

[RSS \(rdsupdates.rss\)](#) Focus mode

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



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**.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE®

Microsoft SQL Server



IBM Db2

IBM Db2



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

Templates

Choose a sample template to meet your use case.

Production

Use defaults for high availability and fast, consistent performance.

Dev/Test

This instance is intended for development use outside of a production environment.

Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)



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.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password [Info](#)



10. In the **Instance configuration** section, set these values:

- **Burstable classes (includes t classes)**
- **db.t3.micro**

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps



Include previous generation classes



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**.

Connectivity Info



Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-1234567890abcdef0

tutorial-ec2-instance-web-server



Some VPC settings can't be changed when a compute resource is added

Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.



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.

tutorial-db-instance

Summary

DB identifier
tutorial-db-instance

CPU
 3.10%

Role
Instance

Current activity
 0 Connections

Connectivity & security

Monitoring

Logs & events

Configuration

Maintenan

Connectivity & security

Endpoint & port

Endpoint
tutorial-db-instance.
west-2.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-west-2a

VPC
tutorial-vpc (vpc-04badc20a546242e6)

Subnet group



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) (<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

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-started.html>)

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.RegionsAndAvailabilityZones.html>)

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

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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) (<https://console.cloud.google.com/iam-admin/iam>)

[Learn more](/sql/docs/mysql/roles-and-permissions) (/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](#) [gcloud](#) (#gcloud) [Terraform](#) (#terraform) More ▾
(#console)

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	<p>Adds a private IP address for your instance. To enable connecting to the instance, additional configuration is required.</p> <p>Optionally, you can specify an allocated IP range for your instances to use for connections.</p> <ol style="list-style-type: none">Expand Show allocated IP range option.Select an IP range from the drop-down menu. <p>Your instance can have both a public and a private IP address.</p>
<p>★ Note: Cloud SQL generates a write endpoint automatically for your Cloud SQL Enterprise Plus edition instance if you do the following:</p> <ol style="list-style-type: none">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.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).Add a private IP address to the instance.Specify an associated network for the instance.Optionally, specify an allocated IP range for the instance. <ul style="list-style-type: none">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	<p>Adds a public IP address for your instance. You can then add authorized networks to connect to the instance.</p> <p>Your instance can have both a public and a private IP address.</p> <p>Learn more about using public IP (/sql/docs/mysql/connect-overview#public_ip).</p>
Authorized networks	<p>Add the name for the new network and the Network address.</p> <p>Learn more (/sql/docs/postgres/authorize-networks#authorized-networks)</p> <p>.</p>
Private path for Google Cloud services	<p>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.</p> <p>★ 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.</p>
Enable Managed Connection Pooling	<p>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).</p>
Security	
Server certificate authority mode	<p>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)</p> <p>.</p> <p>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.</p>

Setting	Notes
Data protection	
Backup tier	<p>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)</p> <p>.</p>
Automate backups	<p>The window of time when you would like backups to start. Learn more (/sql/docs/mysql/instance-settings#backups-and-binary-logging-2ndgen)</p> <p>.</p>
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	<p>Enables point-in-time recovery and write-ahead logging. Learn more (/sql/docs/mysql/instance-settings#backups-and-binary-logging-2ndgen)</p> <p>.</p>
<p>★ Note: The following default behavior applies:</p> <ul style="list-style-type: none"> 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)
<hr/>	
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)
<hr/>	
Flags	
ADD FLAG	You can use database flags to control settings and parameters for your instance. Learn more (/sql/docs/mysql/flags).
<hr/>	
Labels	
<hr/>	

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/mysqlreplication/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.	<p>There are no more available address blocks in the allocated IP range. This can happen for several possible scenarios:</p> <ul style="list-style-type: none">The size of the allocated IP range is too large, for example, larger than /24.The size of the allocated IP range is too small for the number of Cloud SQL instances.

Issue	Troubleshooting
The requirement on the size of IP ranges can limit the number of instances you can create in multiple regions. See Allocate an IP range .	<ul style="list-style-type: none"> The requirement on the size of IP ranges can limit the number of instances you can create in multiple regions. See Allocate an IP range. <p>To resolve this issue, you can either allocate an additional IP range to your instance group or expand the existing allocation.</p> <p>If you used the <code>--allocated-ip-ranges</code> flag when creating the instance, you may only expand the range by 1 IP address at a time.</p> <p>If you're allocating a new range, take care not to overlap with existing allocations.</p> <p>After creating a new IP range, update the configuration of the instance group to include the new range.</p> <pre>gcloud services vpc-peer --service=servicenetwork --ranges=OLD_RESERVED_RANGE \ --network=VPC_NETWORK \ --project=PROJECT_ID \ --force</pre> <p>If you're expanding an existing allocation, make sure the new range does not overlap with the existing range and not decrease it. For example, if trying to create 10 more instances in an existing range going from /24 to /23 is enough.</p> <p>In general, if starting from a /24 range, you can only increase the range by 1 IP address at a time. For example, if trying to create 10 more instances in an existing range going from /24 to /23 is enough.</p> <p>After expanding an existing IP range, update the configuration of the instance group to include the new range.</p> <pre>gcloud services vpc-peer --service=servicenetwork</pre>

Issue	Troubleshooting
	<pre>--ranges=RESERVED_RANGE_ --network=VPC_NETWORK --project=PROJECT_ID</pre>
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 instance again. If the issue persists, contact Google Support.
Error message: HTTPError 400: Invalid request: Incorrect Service Networking config for instance: PROJECT_ID:INSTANCE_NAME:SERVICE_NETWORKING_NOT_ENABLED.	Enable the Service Networking API (/service-infrastructure/docs/service-networking). Run the following command and try to create the instance again:
	<pre>gcloud services enable serviceNetworking --project=PROJECT_ID</pre>
Error message: Failed to create subnetwork. Required 'compute.projects.get' permission for PROJECT_ID.	When you create an instance using the Cloud SQL API, it's created just-in-time using the Service Networking API. If the Service Networking API is not enabled in the project, the instance creation fails. This issue can propagate throughout the system.
Error message: More than 3 subject alternative names are not allowed.	You're trying to use a custom Subject Alternative Name (SAN) certificate of a Cloud SQL instance.
Error message: Subject alternative names %s is too long. The maximum length is 253 characters.	Make sure that any DNS names that you want to use for your Cloud SQL instance don't have more than 253 characters.
Error message: Subject alternative name %s is invalid.	<p>Verify that the DNS names that you want to use for your Cloud SQL instance meet the following requirements:</p> <ul style="list-style-type: none"> They don't have wildcard characters. They don't have trailing dots. They meet RFC 1034 (https://datatracker.ietf.org/doc/html/rfc1034) 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).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](#) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-11-04 UTC.