

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Deploy a Web Application on the Cloud Write a Python Flask application and deploy it on your cloud VM. Configure the firewall to allow HTTP traffic.

**Name: Kirthika S Department: CSE**

A black and white logo

Description automatically generated

**Introduction**

Cloud computing has revolutionized the way applications are developed and deployed, offering scalability, flexibility, and cost-effectiveness. This PoC focuses on deploying a Python-based Flask web application on an AWS EC2 instance. Flask, a lightweight web framework, is ideal for building simple yet powerful web applications. Through this project, you will learn how to set up a virtual machine in AWS, configure it, and deploy a web application, making it accessible to users globally.

**Objectives**

**1. Understand Flask Framework**: Learn the basics of Flask and how to write a simple web application.

**2. Cloud Deployment**: Gain hands-on experience deploying an application on AWS EC2.

**3. Security Configuration**: Configure inbound rules in AWS to allow HTTP traffic securely.

**4. Application Accessibility**: Ensure the application is accessible globally via a public IP.

**5. Real-World Skills**: Develop skills in cloud computing and web application deployment.

**Importance**

**1. Practical Exposure**: Provides real-world experience in deploying applications to the cloud, an essential skill in modern IT infrastructure.

**2. Skill Development**: Improves your understanding of cloud services, virtual machines, and web development.

**3. Scalability**: Demonstrates how applications can be deployed and scaled easily using cloud infrastructure.

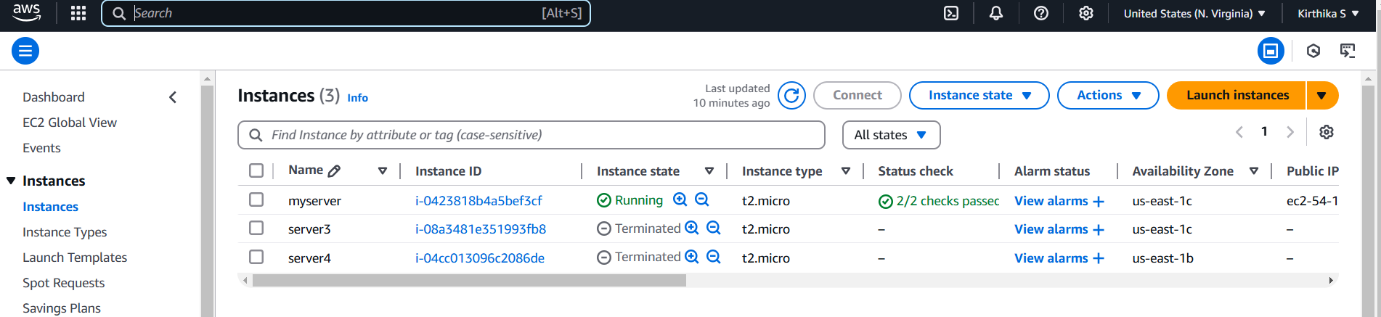
**4. Career Advancement**: Builds foundational knowledge in cloud computing, a highly sought-after skill in the tech industry.

**Step-by-Step Overview**

**Step 1:**

Go to AWS Management Console.

On the EC2 Dashboard, click on **Launch Instances** and enter a name for your instance and select Ubuntu as OS and create a key pair. Leave other settings as default and Click **Launch Instance**.



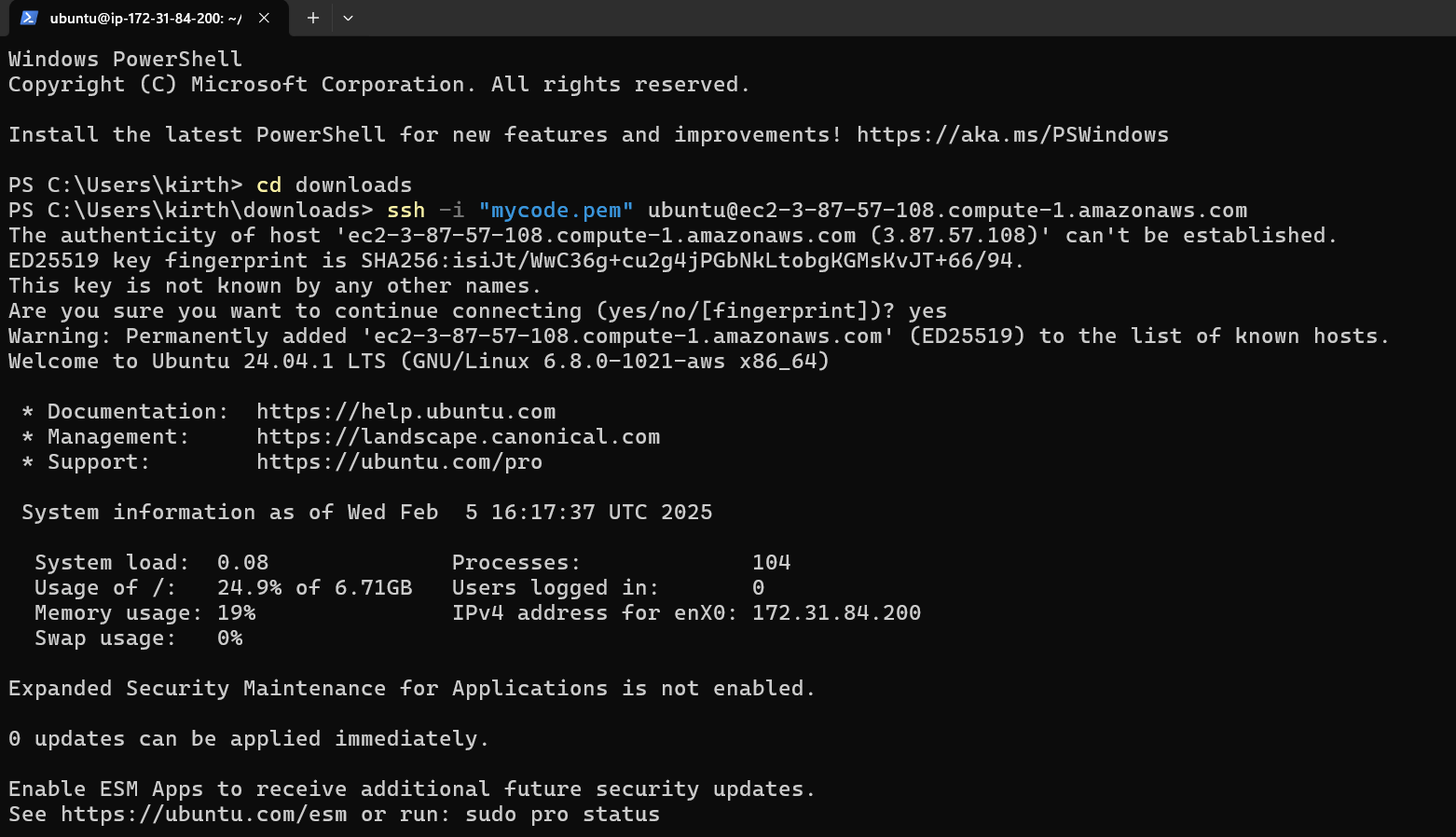
**Step 2:**

Click the 'Connect' option on your launched instance, go to the SSH

client section, and copy the command provided under the 'Example' section.

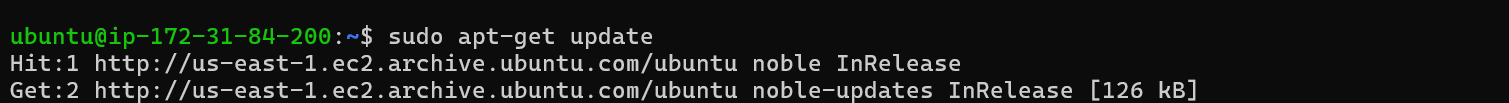
Open PowerShell, navigate to the 'Downloads' directory where the downloaded key pair is located using the **cd Downloads** command

Paste the command copied from the EC2 Connect's SSH client section, replace the key pair name with your downloaded key, press Enter, and type 'yes' when prompted.



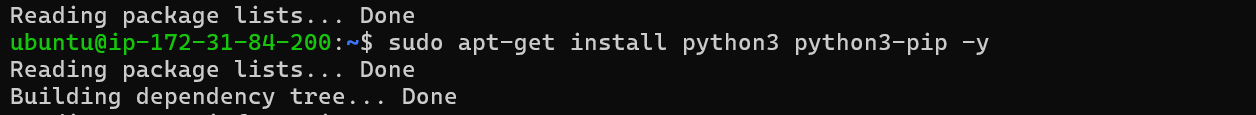
**Step 3:**

Update the Package List :



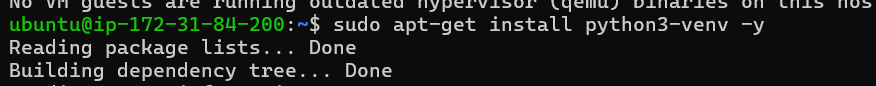
**Step 4:**

Install Python3 and pip



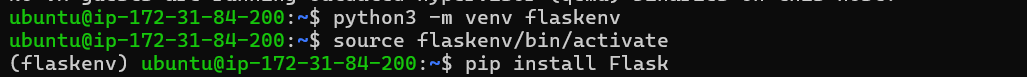
**Step 5:**

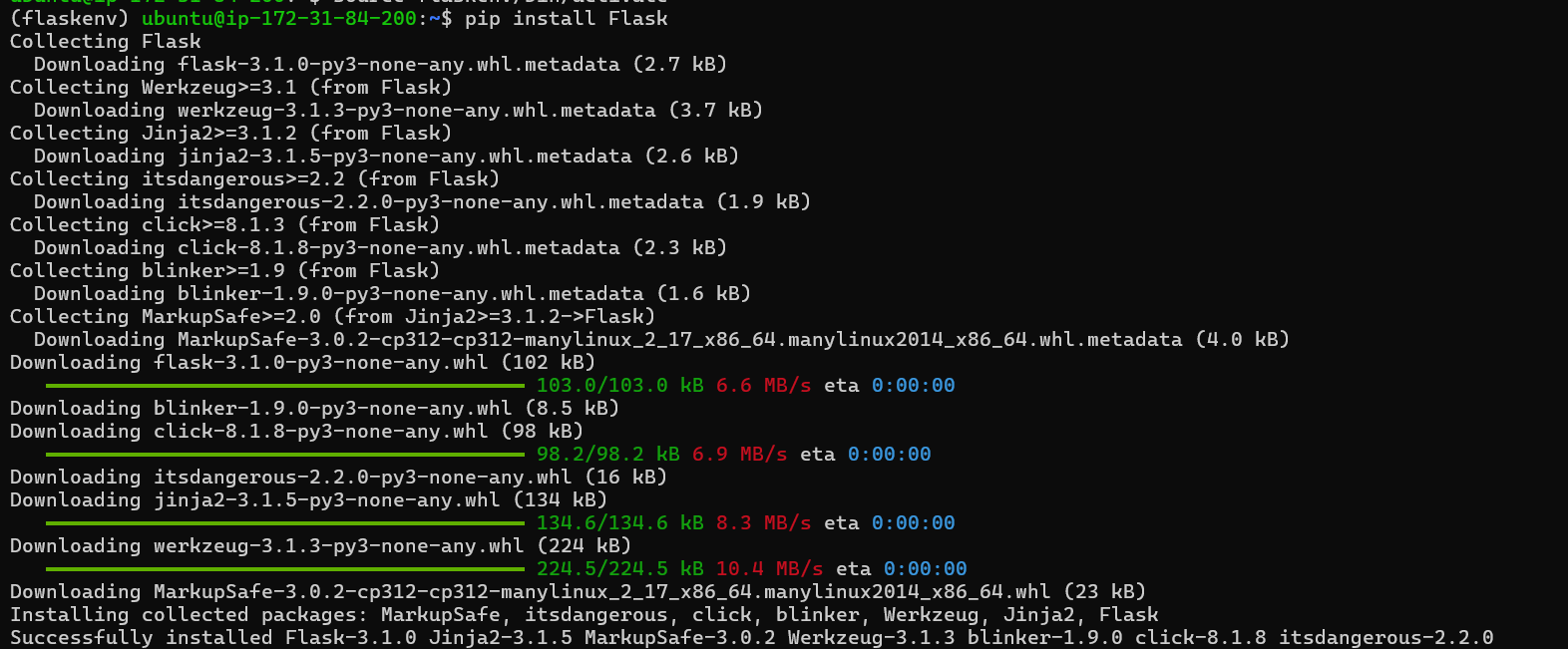
Install Virtual Environment Tools: This helps keep your app’s dependencies separate.

****

**Step 6:**

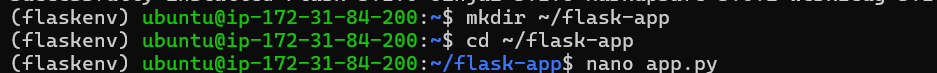
Create and Activate a Virtual Environment and install Flask

****

****

**Step 7:**

Create a Directory for Your App and Create a file called app.py using a text editor (like nano).

****

**Step 8:**

Write this code into the editor and press **Ctrl + O** (to write out) and then **Enter**, then **Ctrl + X** to exit.



**Step 9:**

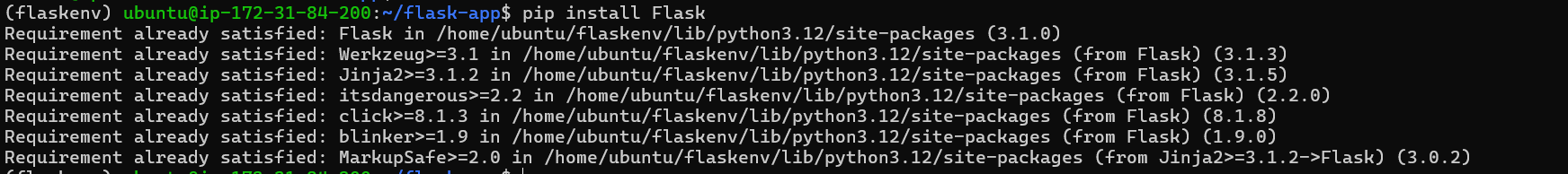
Exit the virtual environment:

****

**Step 10:**

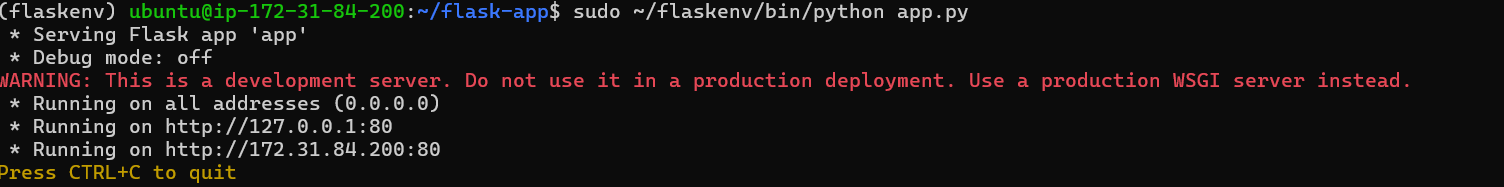
Add your virtual environment’s Python path to the sudo command and Run the application using the virtual environment's Python:

****

****

**Step 11:**

Your Flask app is now running!

****

**Step 12:**

Go to the **EC2 Dashboard** > **Instances**.

Find your instance and note the **Security Group** attached to it.

Navigate to **Security Groups** under the **Network & Security** section.

Select the Security Group associated with your EC2 instance.

Under the **Inbound Rules** tab, ensure there is a rule for **HTTP (port 80)**:

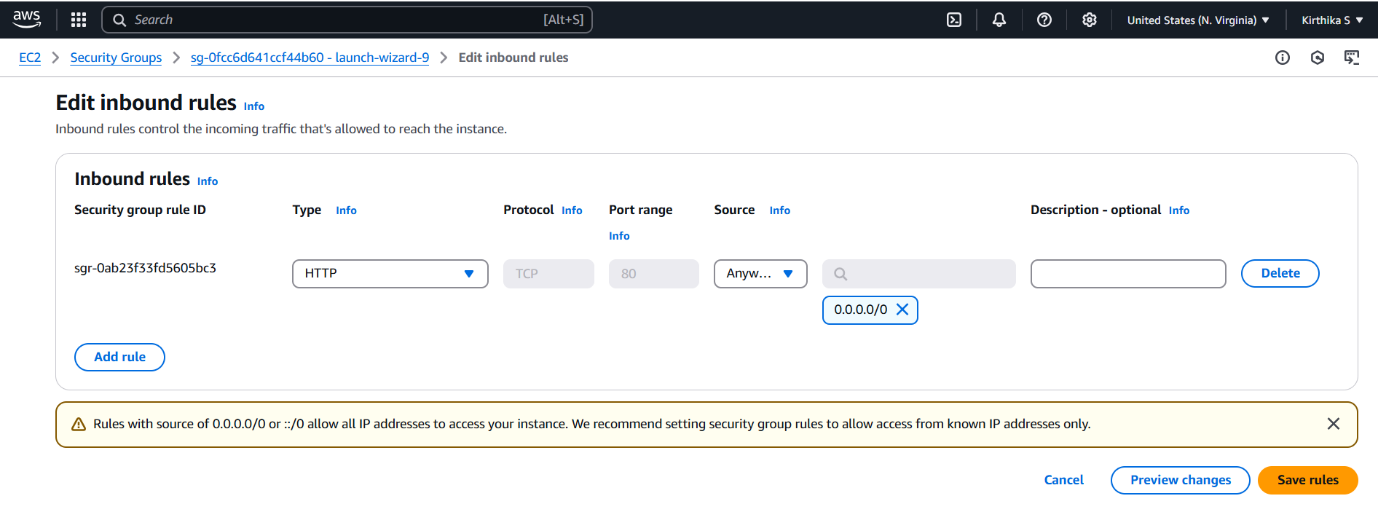
**Type:** HTTP

**Protocol:** TCP

**Port Range: 80**

**Source:** Anywhere (0.0.0.0/0, ::/0)

If there isn't an HTTP rule, click **Edit inbound rules** and add it.

****

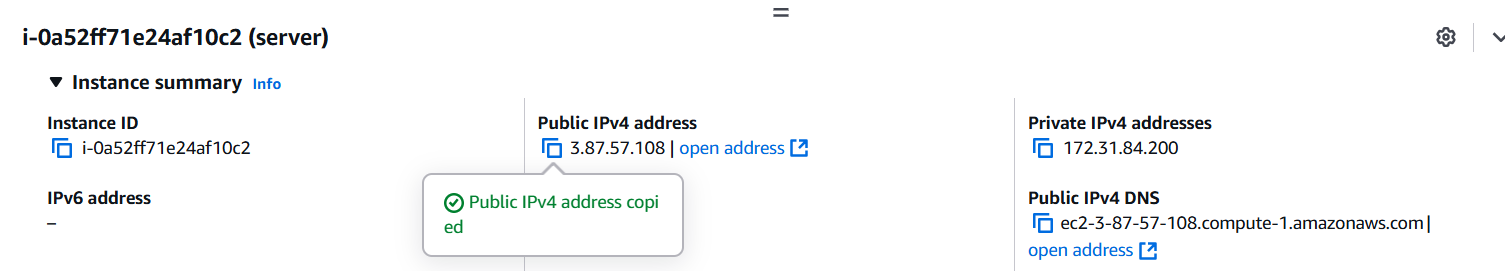
**Step 13:**

Open your browser and navigate to:

http://<Your-Instance-Public-IP>/

Replace <Your-Instance-Public-IP> with the Public IPv4 address of your EC2 instance (e.g: 3.87.57.108).

Public IPv4 address can be found in your Ec2 instance dashboard

****

**Outcome**

By completing this PoC of deploying a Flask web application using an EC2 instance, you will:

1. Launch and configure an EC2 instance with Ubuntu as the operating system.

2. Install and configure the necessary Python environment and dependencies for the Flask framework.

3. Write a simple Flask application (app.py) that displays a message when accessed through a web browser.

4. Host the Flask web application on the EC2 instance and configure it to allow HTTP traffic by updating the security group rules.

5. Access your Flask web application live on the web using the EC2 instance's Public IPv4 DNS or IP address.