Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up a Cloud-Based Monitoring Service Enable basic cloud monitoring (e.g., CloudWatch on AWS). View metrics like CPU usage and disk I/O for your cloud VM.

Name: Vishali V **Department: CSE**

Introduction

Deploying a web application to the cloud involves configuring a virtual machine (VM), setting up a web server, and ensuring proper firewall settings for public access. This process enables developers to make their applications accessible to users over the internet with minimal infrastructure management. Cloud platforms like AWS, Azure, and Google Cloud provide tools to simplify the deployment process. In this guide, we will focus on deploying a Python Flask application to a cloud VM.

Objectives

1. Set up a virtual machine in the cloud.
2. Deploy a Python Flask web application on the VM.
3. Configure firewall settings to allow HTTP traffic.
4. Test the deployed application for accessibility over the internet.

Steps and Detailed Procedure

1. Prepare the Flask Application:
    - Step 1.1: Create a simple Flask application on your local machine.

from flask import Flask app = Flask(__name__) @app.route('/') def home():

return "Hello, Cloud!" if __name__ == '__main__':

app.run(host='0.0.0.0', port=5000)

    - Step 1.2: Save the file as app.py and test it locally:

      python app.py

      Access the application at http://localhost:5000 to ensure it works.

2. Set Up the Cloud VM:
    - Step 2.1: Log in to your cloud provider and create a new virtual machine:
        - AWS: Launch an EC2 instance.
        - Azure: Create a Virtual Machine resource.
        - Google Cloud: Create a Compute Engine VM.

- Step 2.2: Select an image with Python pre-installed (e.g., Ubuntu or Amazon Linux).
- Step 2.3: Configure the VM settings (e.g., instance type, storage, and region) and launch the instance.

3. Transfer the Application to the VM:
   - Step 3.1: Securely copy the Flask application to the VM:

     scp -i your_key.pem app.py user@vm_ip_address:/home/user/
   - Step 3.2: Log in to the VM using SSH:

     ssh -i your_key.pem user@vm_ip_address

4. Install Dependencies on the VM:
   - Step 4.1: Update the package manager and install Python: sudo apt update sudo apt install python3 python3-pip -y o Step 4.2: Install Flask:

     pip3 install flask

5. Run the Flask Application on the VM:
   - Step 5.1: Start the Flask application:

     python3 app.py
   - Step 5.2: Verify the application is running by accessing http://vm_ip_address:5000 in your browser.

6. Configure Firewall Settings:
   - Step 6.1: Open the cloud console and navigate to the firewall settings:
     - AWS: Modify the Security Group to allow inbound traffic on port 5000.
     - Azure: Add an inbound security rule for port 5000.
     - Google Cloud: Update the firewall rules to allow TCP traffic on port 5000.
   - Step 6.2: Save the changes and ensure HTTP traffic is allowed.

7. Automate Application Start-Up (Optional):

- Step 7.1: Use a process manager like systemd to start the application on boot:

  sudo nano /etc/systemd/system/flaskapp.service

Add the following content:

```
[Unit]
Description=Flask Application
After=network.target

[Service]
User=user
WorkingDirectory=/home/user
ExecStart=/usr/bin/python3 /home/user/app.py

[Install]
WantedBy=multi-user.target
```

o Step 7.2: Enable and start the service:

```
sudo systemctl enable flaskapp
sudo systemctl start flaskapp
```

Key Learnings

- Basics of Flask application development.
- Steps to set up a virtual machine in the cloud.
- Configuring firewall rules to allow HTTP traffic.
- Automating application deployment using systems.