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: SUBASHINI T

Department: IT

Introduction and Overview

Deploying a web application on the cloud provides scalability, accessibility, and reliability, making it an essential skill for developers and IT professionals. This guide walks through deploying a Python Flask application on a cloud Virtual Machine (VM), configuring the firewall for HTTP traffic, and ensuring smooth deployment.

Objective

The goal of this guide is to:

- Develop a simple Python Flask web application.
- Deploy the application on a cloud-based VM.
- Configure the firewall to allow HTTP traffic.
- Ensure the web application is accessible publicly over the internet.

Importance of Local Hosting Before Deployment

Before deploying on the cloud, testing the application locally ensures:

- The application functions correctly without errors.
- Dependencies and configurations are properly set up.

- Debugging is easier before moving to a live environment.

Step-by-Step Overview

STEP 1:

Set Up a Cloud Virtual Machine

- Choose a cloud provider (AWS, Google Cloud, Azure, DigitalOcean, etc.).
- Create a new VM instance with an appropriate OS (Ubuntu is recommended).
- Configure SSH access to connect to the VM securely.

STEP 2:

Install Required Dependencies

- Update the system:

  sudo apt update && sudo apt upgrade -y

- Install Python, pip, and virtual environment:

sudo apt install python3 python3-pip python3-venv -y

STEP 3:

Develop a Simple Flask Application

- Create a project directory and navigate to it:

mkdir flask_app && cd flask_app

- Set up a virtual environment:
- python3 -m venv venv source venv/bin/activate

- Install Flask:

pip install flask

- Create app.py:
- from flask import Flask
- app = Flask(__name__)
- @app.route('/')
- def home():
- return "Hello, Flask on Cloud!"
- if __name__ == '__main__':

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

STEP 4:

Run and Test Locally

- Start the Flask application:

python app.py

- Open http://127.0.0.1:5000 in a browser to verify functionality.

STEP 5:

Deploy the Flask Application on the Cloud VM

- Transfer project files to the VM using SCP or Git.
- Install dependencies on the VM and run the application.
- Run Flask in the background using nohup or a process manager like gunicorn.

nohup python app.py

STEP 6:

Configure the Firewall to Allow HTTP Traffic

- Allow port 5000 (or use Nginx for port 80 redirection):
- sudo ufw allow 5000/tcp

sudo ufw enable

- If using AWS or another cloud provider, modify security group rules to allow HTTP traffic.

STEP 7:

Access the Application Publicly

- Obtain the public IP of the VM and access http://<public-ip>:5000 in a browser.

Outcome

By following this guide, you will:

- Successfully deploy a Flask application on a cloud-based VM.
- Enable HTTP traffic and access the application from any device.
- Gain practical experience in cloud hosting and deployment.