 

**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: Rubesh Sk Department: CSE



**Introduction**

Deploying a web application in the cloud allows users to access it from anywhere. Flask, a lightweight Python web framework, is ideal for building and deploying small to medium-scale applications. Hosting a Flask app on an AWS EC2 Windows instance provides flexibility and control over your deployment environment. To make the application accessible, you must configure the firewall settings to allow HTTP traffic.

**Overview**

In this guide, you will:

* Create a Flask web application on a Windows EC2 instance.
* Install Python and necessary dependencies.
* Deploy the Flask app using Gunicorn or the built-in Flask development server.
* Configure Windows Firewall and AWS Security Groups to allow HTTP traffic.

**Objectives**

By the end of this guide, you will be able to:

* Set up a Windows-based EC2 instance for Flask.
* Deploy a Flask web application and access it via a web browser.
* Configure firewall and security group rules to allow HTTP traffic**.**

**Step-by-St** **ep process**

Step 1: Launch a Windows EC2 Instance

1. Log in to the AWS Management Console.
2. Navigate to EC2 and launch a Windows Server instance (e.g., Windows Server 2019).
3. Choose t2.micro (Free Tier eligible).
4. Configure Security Groups:
   * Allow RDP (port 3389) for remote access.
   * Allow HTTP (port 80) for web traffic.
5. Launch the instance and connect via RDP.

Step 2: Install Python and Flask

1. Open PowerShell as Administrator.
2. Download and install Python (if not already installed):

Invoke-WebRequest -Uri https://www.python.org/ftp/python/3.9.10/python-3.9.10-amd64.exe -OutFile python-installer.exe

Start-Process -FilePath python-installer.exe -ArgumentList "/quiet InstallAllUsers=1 PrependPath=1" -Wait

1. Verify installation:

python --version

1. Install Flask:

pip install flask

Step 3: Create a Flask Web Application

1. Navigate to your working directory:

cd C:\Users\Administrator

1. Create a new Python script app.py:

New-Item -Path "app.py" -ItemType File

1. Open app.py in Notepad and add the following Flask code:

python

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return "Hello, Flask is running on Windows EC2!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=80)

1. Save and close the file.

Step 4: Run the Flask Application

1. In PowerShell, start the Flask app:

python app.py

1. If Windows Firewall blocks access, allow Python through the firewall:

New-NetFirewallRule -DisplayName "Allow Python Flask" -Direction Inbound -Action Allow -Protocol TCP -LocalPort 80

Step 5: Configure AWS Security Groups for HTTP Access

1. Go to the AWS EC2 Dashboard.
2. Select your running Windows instance.
3. Click Security Groups > Edit inbound rules.
4. Add a new rule:
   * Type: HTTP
   * Protocol: TCP
   * Port Range: 80
   * Source: Anywhere (0.0.0.0/0)
5. Click Save rules.

Step 6: Test the Deployment

1. Open a browser on your local computer.
2. Enter your EC2 public IP:

http://<your-ec2-public-ip>

1. If successful, you should see:

Hello, Flask is running on Windows EC2!

**Outcome**

By following these steps, you successfully deployed a Flask web application on an AWS EC2 Windows instance. You also configured firewall and security settings to allow HTTP traffic, making your app accessible over the internet. This setup provides a foundation for deploying more complex web applications with databases and APIs.