 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

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

Name: MIRDULA C Department: ADS



**Introduction**

### Cloud deployment is a crucial step in modern web application development. By deploying a Flask application on a cloud platform like AWS, developers can make their applications accessible over the internet, ensuring scalability, security, and reliability. This guide will walk you through deploying a simple Python Flask application on an AWS Virtual Machine (EC2), configuring a firewall to allow HTTP traffic, and ensuring your app is accessible online.

### ****Overview****

In this tutorial, you will:

1. Set up an AWS EC2 instance (virtual machine).
2. Install necessary dependencies for Python and Flask.
3. Develop a simple Flask application.
4. Run the Flask app and make it accessible.
5. Configure the firewall to allow HTTP traffic.
6. Deploy using a web server like Gunicorn & Nginx for production use.

This step-by-step approach ensures a smooth deployment process, making your Flask app available for users.

### **Objectives**

* Understand how to set up an AWS EC2 instance.
* Deploy a Flask web application on a cloud environment.
* Configure firewall rules to allow HTTP traffic.
* Use a production-grade web server (Gunicorn + Nginx) for hosting.
* Scalable Solution: Enable the script to efficiently handle large volumes of files.

**Importance**

 Scalability: Cloud deployment allows applications to handle increased traffic.

 Accessibility: Hosting on the cloud ensures global reach.

 Security: AWS provides firewall configurations for enhanced security.

 Cost-Efficiency: Cloud services allow pay-as-you-go pricing.

**Step-by-Step Overview**

**Step 1: Set Up an AWS EC2 Instance**

1. Log in to AWS and navigate to EC2 Dashboard.
2. Click Launch Instance and select Amazon Linux 2 / Ubuntu as your OS.
3. Choose an instance type (e.g., t2.micro for free-tier users).
4. Configure instance details and add storage.
5. Set up security group rules to allow SSH (port 22) and HTTP (port 80).
6. Launch the instance and download the key pair (.pem file) for SSH access.

**Step 2: Connect to Your Instance**

1. Open a terminal and navigate to the downloaded key pair location.
2. Connect using SSH

**Step 3: Install Python and Flask**

1.Update the system package

2.Install Python and pip

3.Install Flask

**Step 4: Create a Simple Flask App**

1. Create a new project directory
2. Create a Python file (app.py) and add the following Flask code
3. Run the Flask app

**Step 5: Configure AWS Firewall (Security Groups) for HTTP Access**

1. Go to EC2 Dashboard → Security Groups.
2. Select the security group attached to your instance.
3. Click Edit Inbound Rules and add the following rules:
   * Rule 1: Allow HTTP (Port 80) from anywhere (0.0.0.0/0).
   * Rule 2: Allow TCP Port 5000 (optional for testing Flask).
4. Save the changes.

**Step 6: Install and Configure Gunicorn & Nginx for Production**

1. Install Gunicorn
2. Run Flask using Gunicorn
3. Install and configure Nginx
4. Configure Nginx to proxy requests to Gunicorn
5. Enable the Nginx configuration

#### **Step 7: Access Your Deployed Flask App**

* Open a browser and go to
* You should see "Hello, Flask on AWS!" displayed.

**Outcome**

After completing this deployment, you will have:

* A running Flask web application on an AWS EC2 instance.
* Configured firewall rules to allow public access.
* Used Gunicorn and Nginx for a production-grade setup.
* Successfully made your app accessible over the internet.