 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Host a Static Website on a Cloud VMInstall Apache/Nginx on your cloud VM and host a simple HTML website.

Name: MIRDULA C Department: ADS



**Introduction**

### Hosting a static website on a cloud virtual machine (VM) allows users to serve web pages over the internet using cloud infrastructure. A static website consists of HTML, CSS, and JavaScript files without any backend processing. By using a web server like Apache or Nginx, we can deploy a simple website and make it accessible to users globally.

### In this guide, we will install Apache or Nginx on a cloud VM, upload a simple HTML page, configure firewall settings, and access the website via a public IP.

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

### This tutorial will cover:

### Provisioning a Cloud VM (AWS EC2, Google Compute Engine, or Azure VM).

### Installing a web server (Apache or Nginx).

### Uploading a simple static website (HTML, CSS, JavaScript files).

### Configuring firewall rules to allow HTTP/HTTPS traffic.

### Accessing the website via a browser.

### Hosting a website on a cloud VM provides more control compared to shared hosting and allows customization of configurations and security settings.

### **Objectives**

* Set up a Cloud VM to host a website.
* Install and configure Apache or Nginx as a web server.
* Deploy a simple HTML website.
* Configure firewall rules to allow public access.
* Verify that the website is accessible from the internet.

**Importance**

✅ Custom Hosting Environment – Full control over web server settings and configurations.  
✅ Scalability – Can scale up resources based on traffic demand.  
✅ Cost-Effective – Using a cloud VM is often cheaper than traditional hosting for small projects.  
✅ Security – Cloud providers offer security measures such as firewalls and access control.

**Step-by-Step Overview**

**Step 1: Set Up a Cloud VM**

1. Log in to your cloud provider (AWS, Google Cloud, or Azure).
2. Create a Virtual Machine (VM):
   * For AWS: Launch an EC2 instance (Ubuntu or Amazon Linux).
   * For Google Cloud: Create a Compute Engine instance.
   * For Azure: Deploy a Virtual Machine (Linux-based).
3. Choose an OS (Ubuntu, Debian, or CentOS recommended).
4. Configure networking:
   * Enable HTTP (port 80) and HTTPS (port 443) traffic in the firewall settings.
5. Connect to the VM using SSH

**Step 2: Install a Web Server (Apache or Nginx)**

**For Apache (Ubuntu/Debian-based systems)**

1. Update the system
2. Install Apache
3. Start and enable the Apache service
4. Check if Apache is running

**For Nginx (Ubuntu/Debian-based systems)**

1. Install Nginx
2. Start and enable the Nginx service
3. Check if Nginx is running

**Step 3: Upload a Simple Static Website**

1. Navigate to the web server’s root directory:
   * **Apache (Ubuntu/Debian)**:

Nginx (Ubuntu/Debian)

1. Create a simple HTML file
2. Add the following HTML code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>My Cloud Website</title>

</head>

<body>

<h1>Welcome to My Static Website Hosted on a Cloud VM!</h1>

</body>

</html>

1. Save the file (CTRL + X, then Y, then Enter).

**Step 4: Configure Firewall Rules**

1. **Allow HTTP and HTTPS Traffic**:
   * For **Ubuntu/Debian (UFW Firewall)**:
   * For **AWS EC2**:
     + Go to **Security Groups**.
     + Edit **Inbound Rules** → Add **HTTP (80) and HTTPS (443)**.
   * For **Google Cloud**:
     + Navigate to **VPC Network** → **Firewall rules**.
     + Create a rule to allow **TCP traffic on ports 80 and 443**.
   * For **Azure**:
     + Go to **Networking** → Add inbound rule **Allow HTTP/HTTPS**.

**Step 5: Access the Website**

1. Open a web browser.
2. Enter your **VM’s public IP address**
3. If everything is configured correctly, you should see your static website’s homepage.

**Outcome**

After completing this setup, you will have:  
✅ A **fully functional web server** (Apache/Nginx) on a cloud VM.  
✅ A **hosted static website** accessible via the internet.  
✅ **Firewall rules configured** for secure web access.  
✅ **A basic understanding** of deploying websites on cloud infrastructure.