

Placement Empowerment Program Cloud Computing and DevOps Centre

Host a Static Website Locally: Set Up a Local Server
Apache and Host a Simple HTML page with your
name

Name:Madhumitha.H

Department:ADS

Introduction

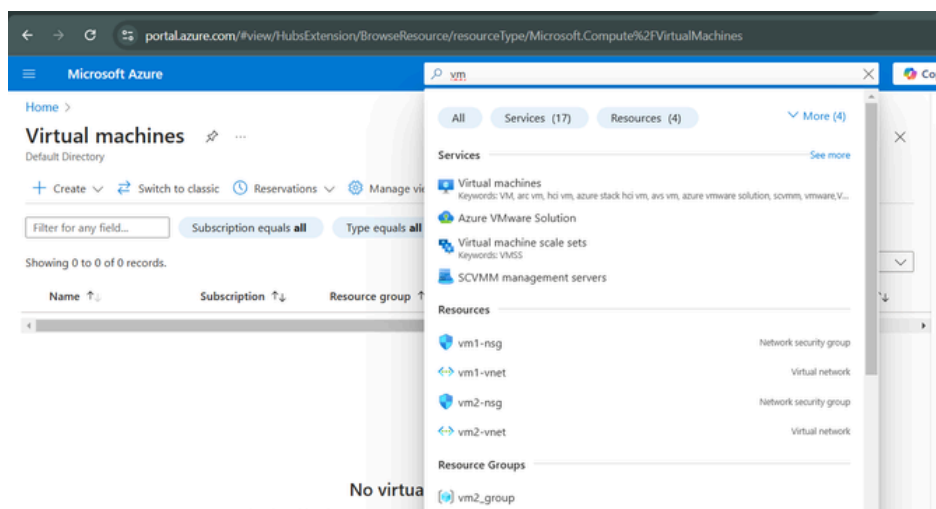
In this proof of concept (POC), Overview: Hosting a Static Website on Azure Using Apache

This task involves deploying a static website on an Azure Virtual Machine (VM) using Apache, a widely used web server. A static website consists of fixed HTML, CSS, and JavaScript files, which do not require backend processing or databases.

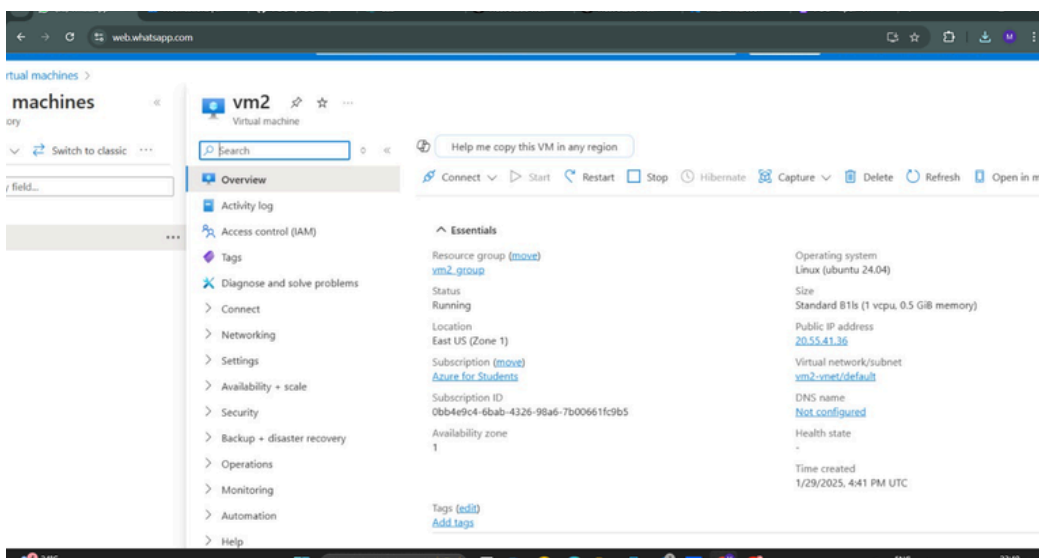
The process includes setting up a cloud-based VM, installing and configuring Apache, and hosting an HTML webpage accessible via a web browser. Once deployed, the website can be accessed using the public IP address of the Azure VM.

This setup demonstrates fundamental cloud computing, web hosting, networking, and security concepts, providing hands-on experience with server deployment, firewall configuration, and basic web hosting in a cloud environment. It is a foundational step for learning web development, DevOps, and cloud infrastructure.

Step 1: Set Up An Azure Vm



Search for virtua machine in the azure portal



Create a VM with the following configuration and give review +create at the end

Step2: connect to the vm via powershell using SSH or

```
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\madhu> ssh madhu@20.55.41.36
The authenticity of host '20.55.41.36 (20.55.41.36)' can't be established.
ED25519 key fingerprint is SHA256:T0qkBUZZWC79p0w7VdjzGTyuoKzmx3ugJCAesEomWLM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.55.41.36' (ED25519) to the list of known hosts.
madhu@20.55.41.36's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1020-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Jan 29 16:43:18 UTC 2025

System load:  0.59               Processes:    110
Usage of /:   5.4% of 28.02GB    Users logged in: 0
```

connection via powershell using this command :

`ssh username@your-vm-ip`

Step 3 : Install apache webserver after logging into your vm

1)update package :

in the powershell after logging in give the update package command :**sudo apt update && sudo apt upgrade -y**

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

madhu@vm2:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301
Get:9 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
```

2)Install apache:

sudo apt install apache2 -y

```
madhu@vm2:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-t64
  liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libap
  aprutil1t64
liblua5.4-0 ssl-cert
```

3)Start apache:

sudo systemctl start apache2

```
madhu@vm2:~$ sudo systemctl start apache2
madhu@vm2:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/s
ll.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
madhu@vm2:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-01-29 16:45:06 UTC; 1min 12s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2778 (apache2)
    Tasks: 55 (limit: 400)
   Memory: 5.5M (peak: 8.2M)
      CPU: 37ms
   CGroup: /system.slice/apache2.service
           └─2778 /usr/sbin/apache2 -k start
             └─2780 /usr/sbin/apache2 -k start
               └─2781 /usr/sbin/apache2 -k start
```

4)Enable apache to start on Boot:

sudo systemctl enable apache2

command is given after giving the start apache command .refer the above image for reference

5)Check apache status:

sudo systemctl status apache

```
madhu@vm2:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-01-29 16:45:06 UTC; 1min 12s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2778 (apache2)
    Tasks: 55 (limit: 400)
   Memory: 5.5M (peak: 8.2M)
      CPU: 37ms
   CGroup: /system.slice/apache2.service
           └─2778 /usr/sbin/apache2 -k start
             └─2780 /usr/sbin/apache2 -k start
               └─2781 /usr/sbin/apache2 -k start

Jan 29 16:45:06 vm2 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 29 16:45:06 vm2 systemd[1]: Started apache2.service - The Apache HTTP Server.
madhu@vm2:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
madhu@vm2:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
```

Step 4: configure firewall to allow HTTP Traffic

1)Allow apache through Firewall : **sudo ufw allow'apache'**

2) enable firewall: **sudo ufw enable**

3)check firewall:**sudo ufw status**

```
└─2780 /usr/sbin/apache2 -k start
  └─2781 /usr/sbin/apache2 -k start

Jan 29 16:45:06 vm2 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 29 16:45:06 vm2 systemd[1]: Started apache2.service - The Apache HTTP Server.
madhu@vm2:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
madhu@vm2:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

Step 5: Deploy a simple html page

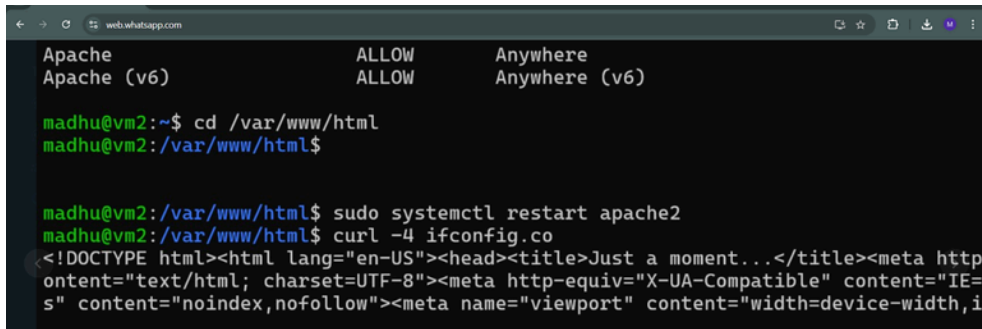
1) Navigate to web root dictionary:

Cd/var/www/html

2) create new Html page:

sudo nano index.html

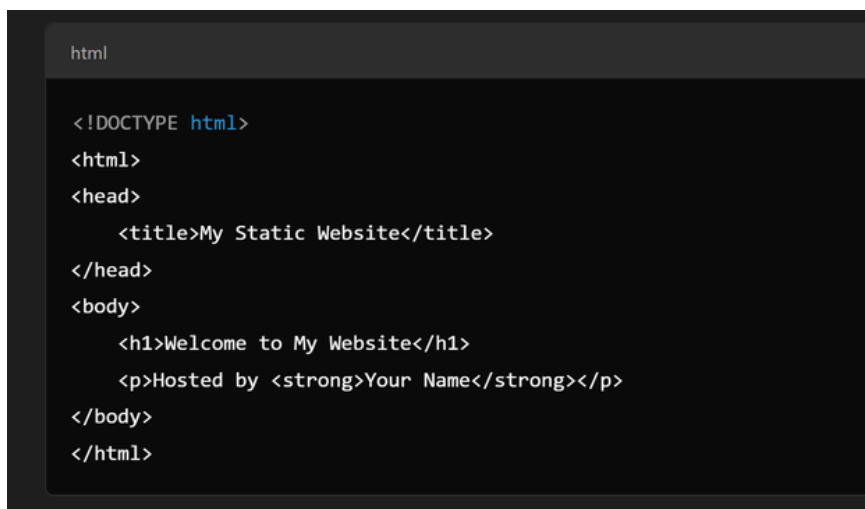
This command directly navigates to the html file so that you can paste or type in your html code there for your static website



```
madhu@vm2:~$ cd /var/www/html
madhu@vm2:/var/www/html$
madhu@vm2:/var/www/html$ sudo systemctl restart apache2
madhu@vm2:/var/www/html$ curl -4 ifconfig.co
<!DOCTYPE html><html lang="en-US"><head><title>Just a moment...</title><meta http-
ontent="text/html; charset=UTF-8"><meta http-equiv="X-UA-Compatible" content="IE=
s" content="noindex,nofollow"><meta name="viewport" content="width=device-width, in
```

once you have finished typing in our content give **Ctrl+X** then **Y** press enter

example content:



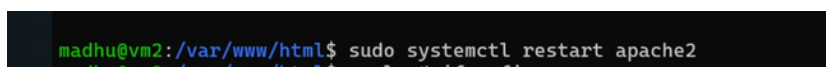
```
html

<!DOCTYPE html>
<html>
<head>
  <title>My Static Website</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>Hosted by <strong>Your Name</strong></p>
</body>
</html>
```

Step 6:

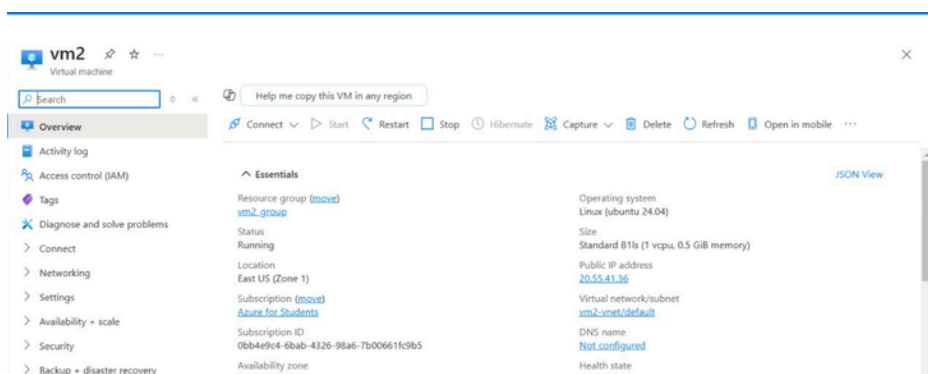
1) restart the apache to apply changes:

sudo systemctl restart apache2



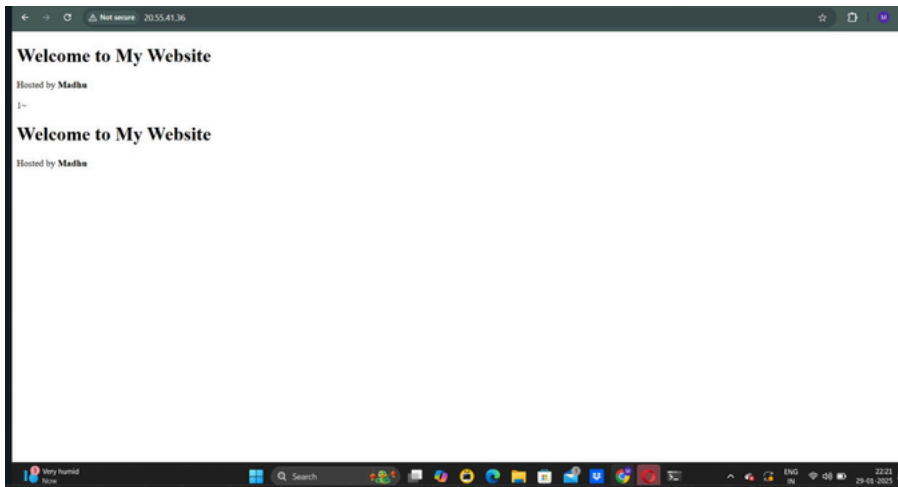
```
madhu@vm2:/var/www/html$ sudo systemctl restart apache2
```

2) Get your ip address: Get it from the over view page of your azure vm



3) access your page by opening a web browser and visit:

http://your-vm-ip



Outcomes of the Task

- 1)Set up a virtual machine on Azure to act as a web server.
- 2)Installed and configured Apache to serve web pages.
- 3)Created and deployed a simple static HTML page.
- 4)Configured firewall settings to allow HTTP traffic.
- 5)Accessed the hosted website using the virtual machine's public IP address.