Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up a Virtual Machine in the CloudCreate a free tier AWS, Azure, or GCP account. Launch a virtual machine and SSH into it.

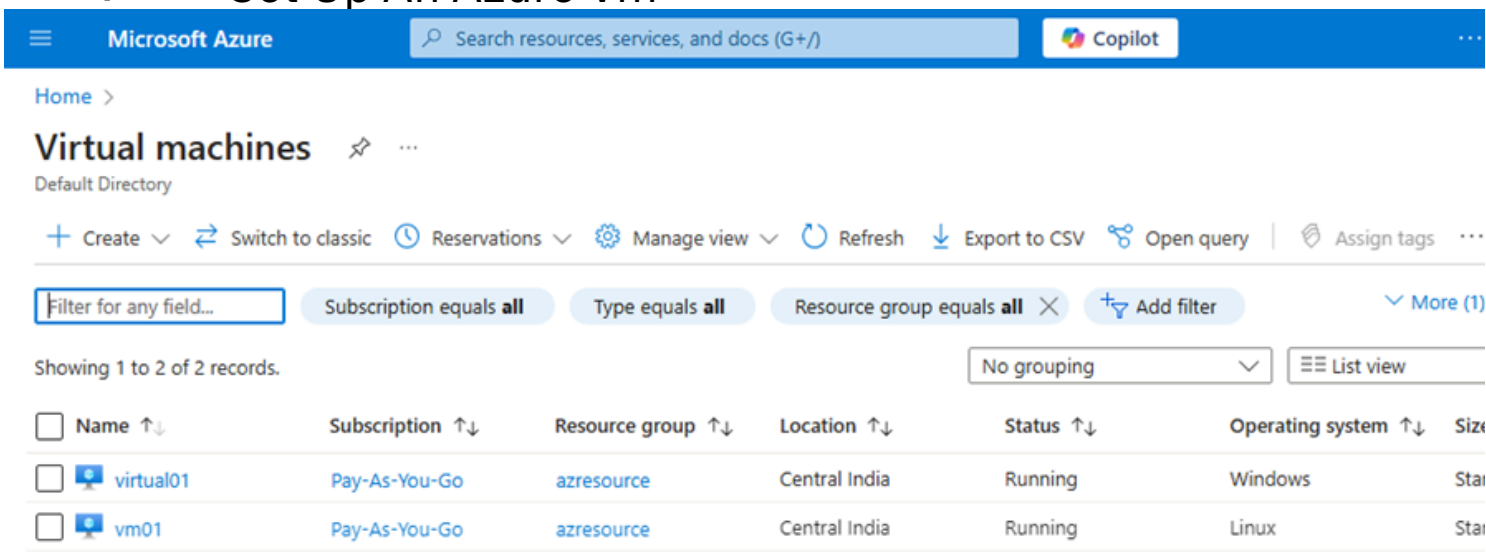
Name: Keshika D Department:ADS

Introduction

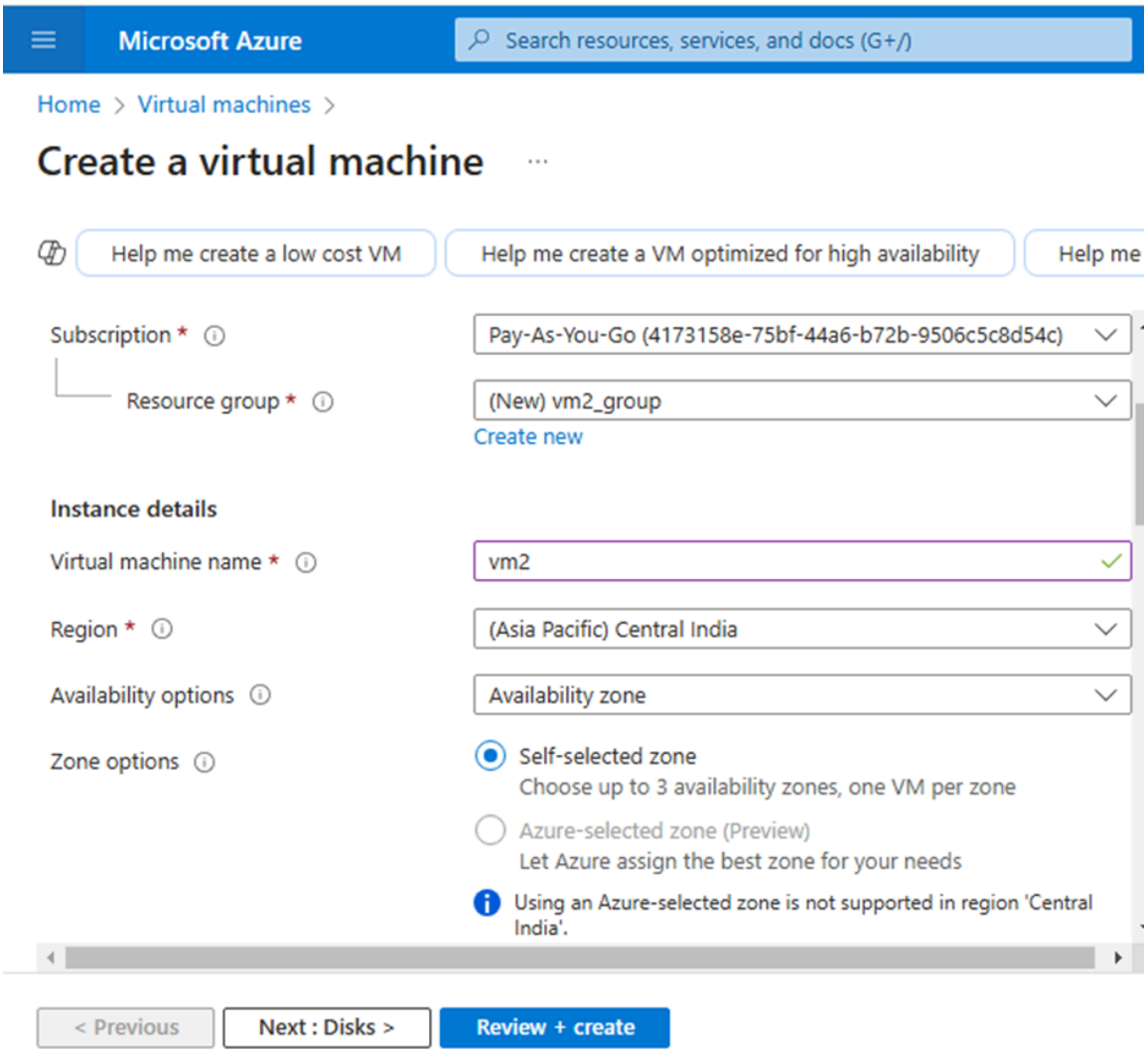
In this proof of concept (POC), Overview: Hosting a Static Website on Azure Using Apache This task involves deploying a static website on an Azure Virtual Machine (VM) using Apache, a widely used web server. A static website consists of fixed HTML, CSS, and JavaScript files, which do not require backend 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 comp uting, 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

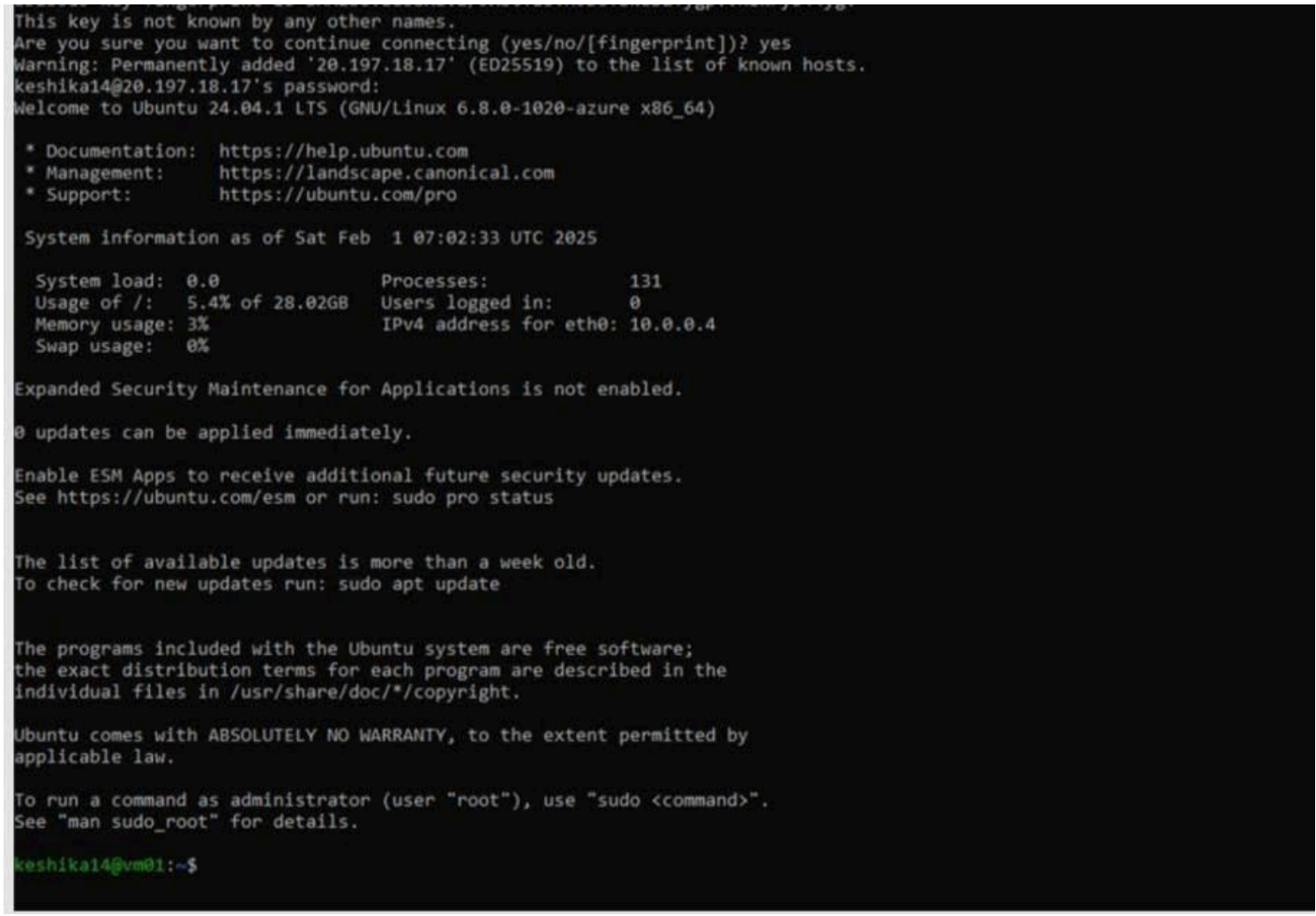


Search for virtual machine in the azure portal



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

Step2: connect to the vm via powershell using SSH or



connection via powershell using this command :

ssh username@your-vm-ip

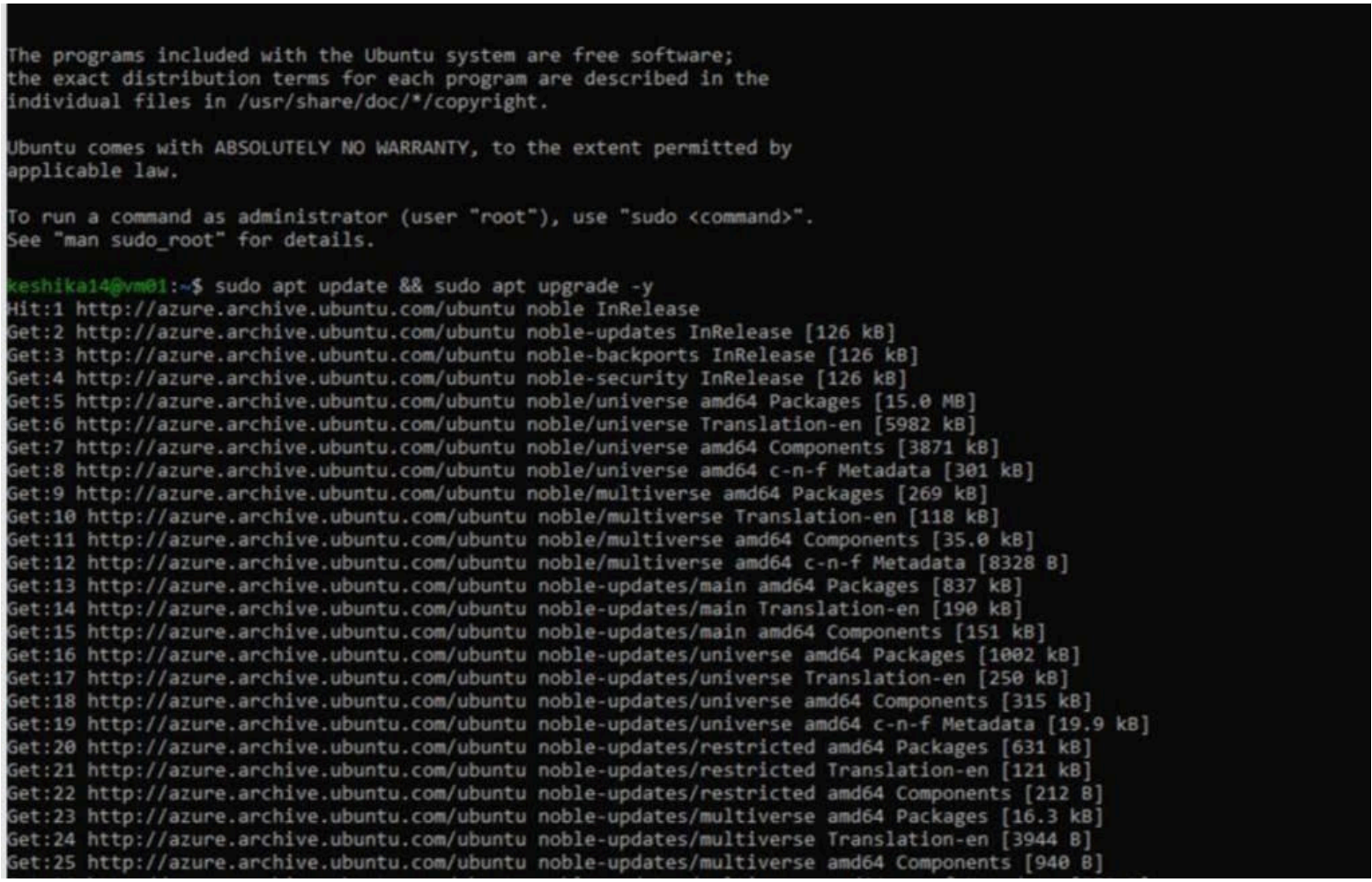
Step 3 :

Install apache webserver after logging into your vm

1)update package :

in the powershell after loggining give the update package command :**sudo apt update && sudo apt**

upgrade -y



2)Install apache:

sudo apt install apache2 -y

3)Sart apache:

sudo systemctl start apache 2

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
keshika14@vm01:~$ 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-ldap libaprutil1t64 liblua5.
  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 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 k
Get:4 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.5 [1329 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.5 [163 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.5 [97.1 kB
Get:9 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.5 [90.2 kB]
```

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:

```
eshika14@vm01:~$ sudo systemctl start apache2
eshika14@vm01:~$ sudo systemctl enable apache2
ynchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
xecuting: /usr/lib/systemd/systemd-sysv-install enable apache2
eshika14@vm01:~$ 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 Sat 2025-02-01 07:14:02 UTC; 5min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 14148 (apache2)
      Tasks: 55 (limit: 9459)
     Memory: 5.5M (peak: 5.9M)
        CPU: 57ms
     CGroup: /system.slice/apache2.service
             ├─14148 /usr/sbin/apache2 -k start
             ├─14151 /usr/sbin/apache2 -k start
             └─14152 /usr/sbin/apache2 -k start

eb 01 07:14:02 vm01 systemd[1]: Starting apache2.service - The Apache HTTP Server...
eb 01 07:14:02 vm01 systemd[1]: Started apache2.service - The Apache HTTP Server.
eshika14@vm01:~$ sudo ufw allow 'Apache'
ules updated
ules updated (v6)
eshika14@vm01:~$ sudo ufw enable
ommand may disrupt existing ssh connections. Proceed with operation (y|n)? y
irewall is active and enabled on system startup
eshika14@vm01:~$
```

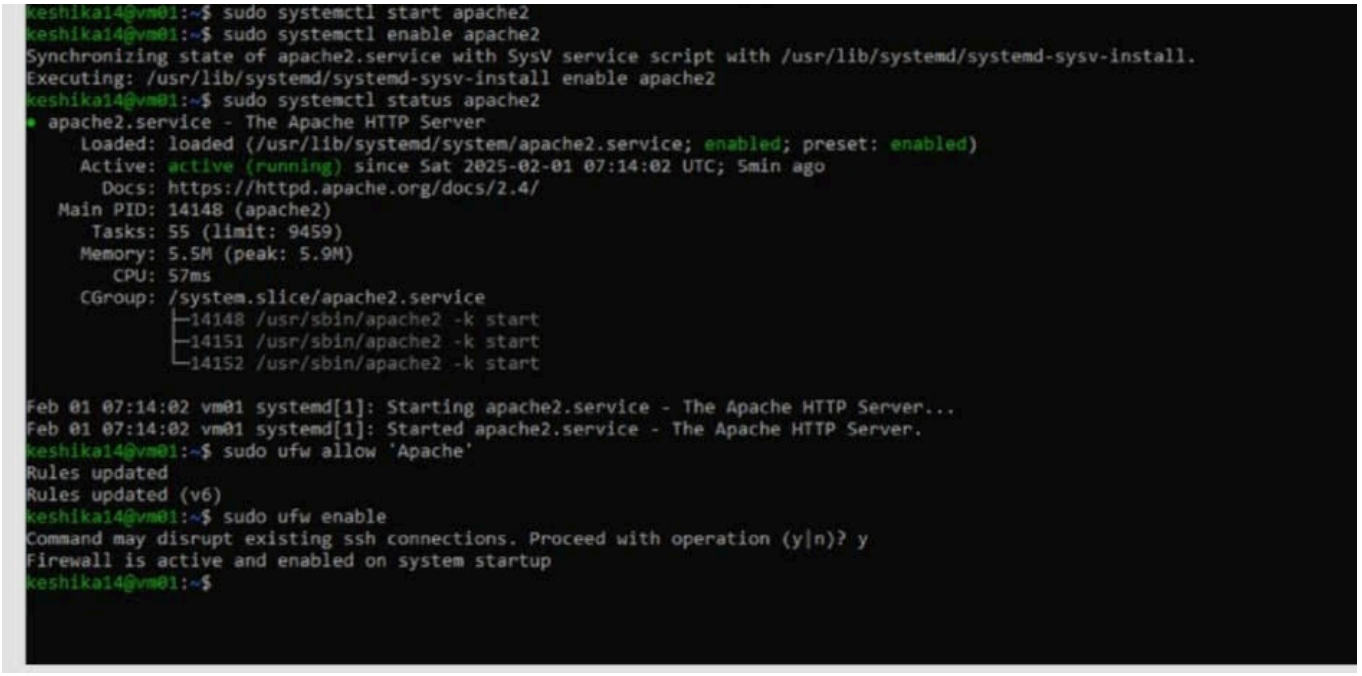
sudo systemctl status apache

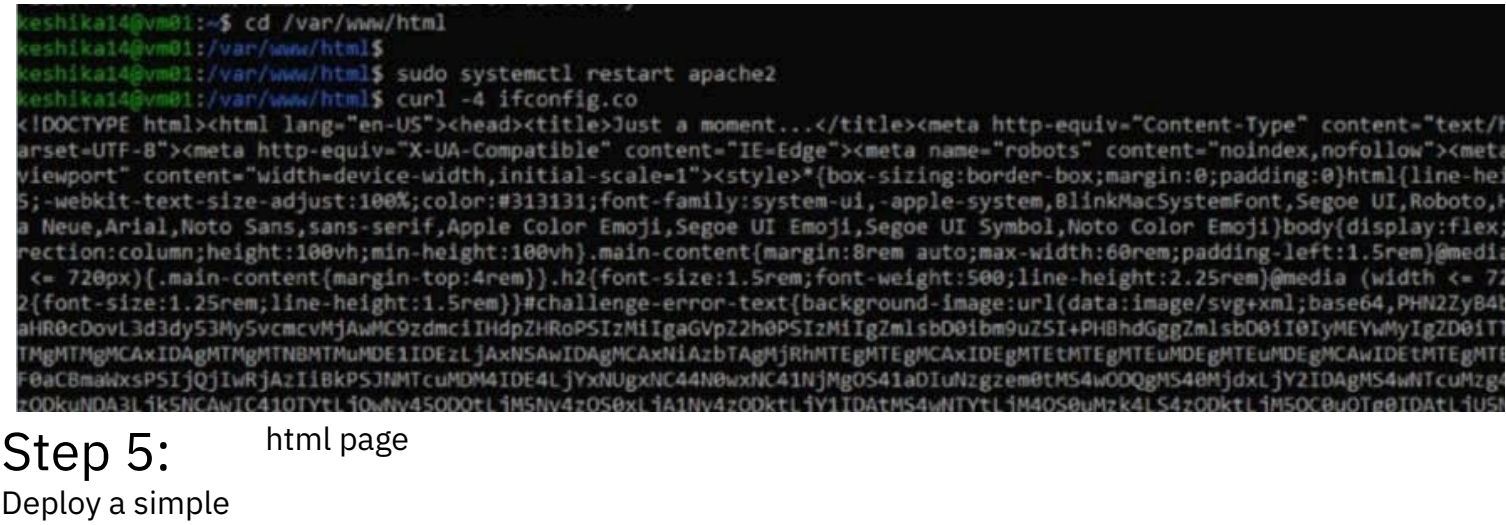
configure firewall to
allow HTTP Traffic

Step 4:

1)Allow apache through Firewall : **sudo ufw allow'apache'** 2) enable firewall: **sudo ufw enable**

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





Step 5:

html page

Deploy a simple

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

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Web Page</title>
</head>
<body>
    <h1>Welcome to My Website!</h1>
    <p>This is a test page hosted on Apache.</p>
</body>
</html>
```

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

press enter example content:

Step 6:

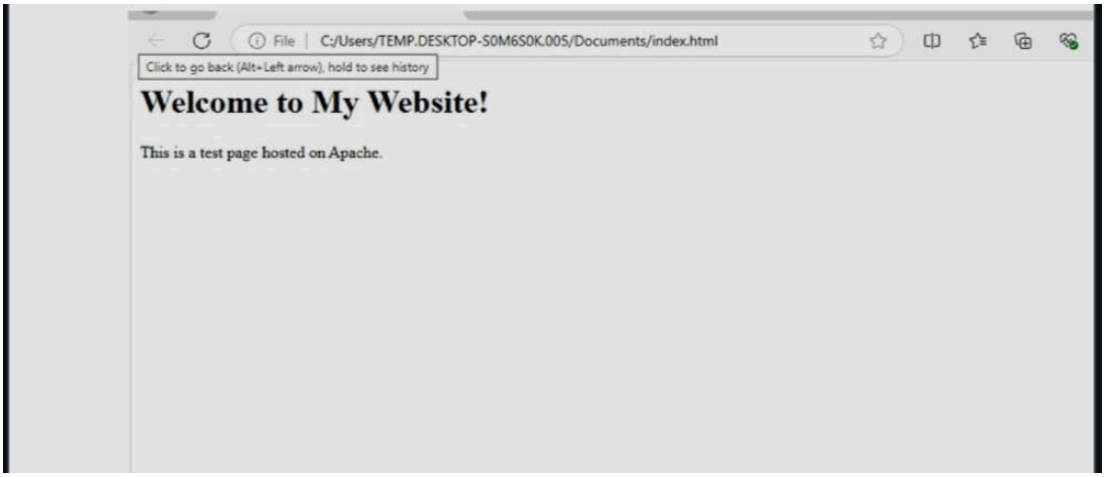
1) restart the apache to apply changes:

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.