**Practical 4**

**Hosting Multiple Websites on a Single Apache Server using Virtual Hosts**

**Aim:**

**To configure and host multiple websites on a single Apache server inside a Virtual Machine using Virtual Hosts, and access them from a host machine by mapping domain names to the VM's IP address.**

Write up:

**Requirements:**

* **Ubuntu VM with internet access and Apache installed**
* **Apache2 service running**
* **Host machine (Windows/Linux/macOS) with a browser**
* **Bridged or Host-only Networking enabled**
* **Sudo/root access in VM**

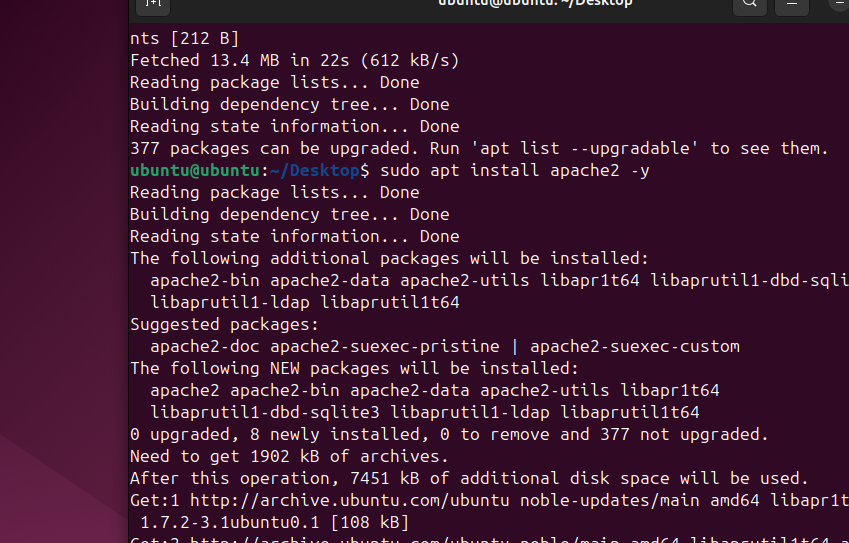
Apache supports **Virtual Hosting**, which allows a single web server to serve **multiple domain names** using different configuration files. Each website is stored in a different directory and mapped using a unique domain like example.com, test.com.

This simulates **shared hosting** on production servers.

**Step 1: Install Apache (If Not Installed)**

sudo apt update

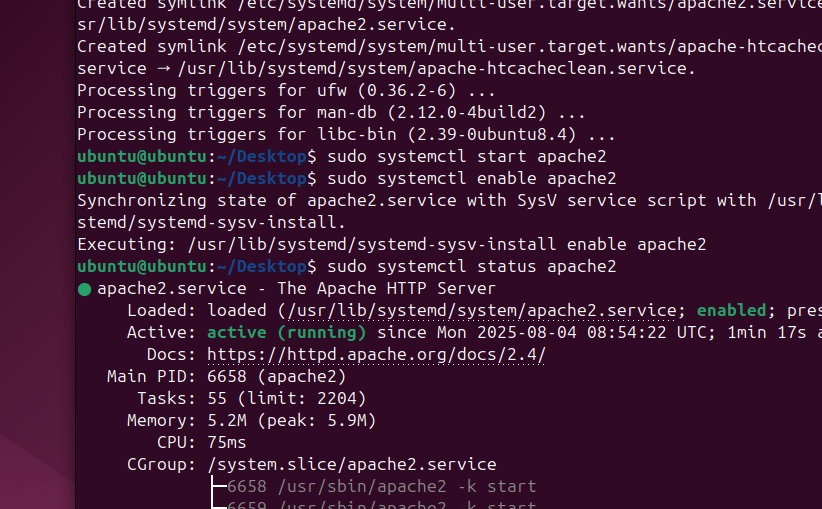
sudo apt install apache2 –y



**Start and enable Apache:**

sudo systemctl start apache2

sudo systemctl enable apache2



**Step 2: Create Website Folders and HTML Pages**

sudo mkdir -p /var/www/example.com

sudo mkdir -p /var/www/test.com

**Add simple HTML files:**

echo "<h1>Hello from example.com</h1>" | sudo tee /var/www/example.com/index.html

echo "<h1>Hello from test.com</h1>" | sudo tee /var/www/test.com/index.html

**Step 3: Set Permissions**

sudo chown -R www-data:www-data /var/www/example.com

sudo chown -R www-data:www-data /var/www/test.com

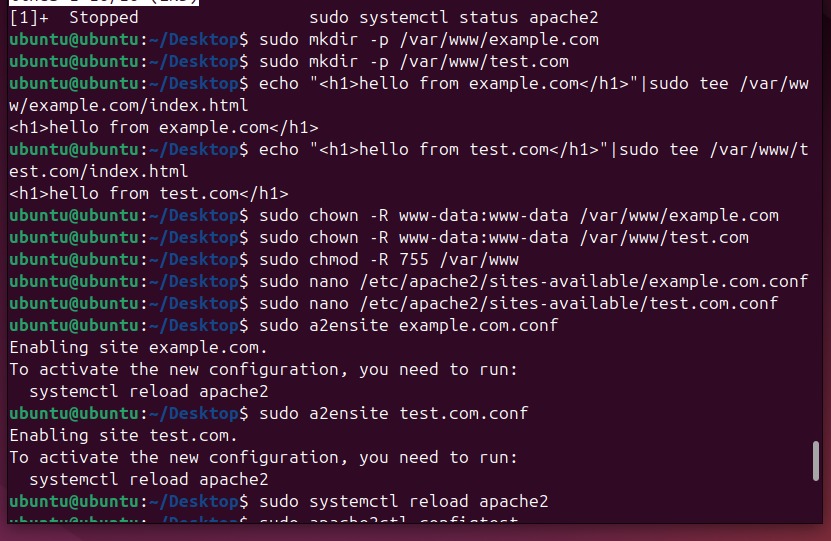
sudo chmod -R 755 /var/www

chown = **change ownership**

-R = apply changes **recursively** (to all files and folders inside)

www-data:www-data = set the **owner and group** to Apache's default user (www-data)

/var/www/example.com = the folder you are changing



**Step 4: Create Apache Virtual Host Config Files (ctrl + O SAVE ,Enter,Ctrl +X/Z)**

**1. example.com Virtual Host**

sudo nano /etc/apache2/sites-available/example.com.conf

<VirtualHost \*:80>

ServerName example.com

DocumentRoot /var/www/example.com

<Directory /var/www/example.com>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/example.com-error.log

CustomLog ${APACHE\_LOG\_DIR}/example.com-access.log combined

</VirtualHost>

**2. test.com Virtual Host**

sudo nano /etc/apache2/sites-available/test.com.conf

<VirtualHost \*:80>

ServerName test.com

DocumentRoot /var/www/test.com

<Directory /var/www/test.com>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/test.com-error.log

CustomLog ${APACHE\_LOG\_DIR}/test.com-access.log combined

</VirtualHost>

**🔹 Step 5: Enable the Sites and Reload Apache**

sudo a2ensite example.com.conf

sudo a2ensite test.com.conf

sudo systemctl reload apache2

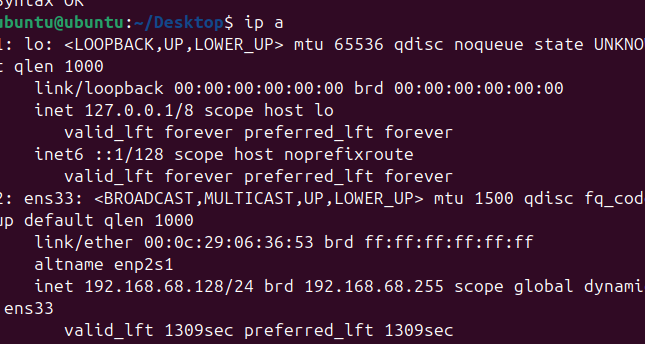
Check config:

sudo apache2ctl configtest

Output : Syntax OK

**🔹 Step 6: Get Your VM's IP Address**

ip a



**Step 7: Map Domains in Host OS (hosts file)**

**On Windows:**

1. Open Notepad **as Administrator**
2. Open file:

C:\Windows\System32\drivers\etc\hosts

1. Add at the bottom:

192.168.68.128 example.com

192. 168.68.128 test.com

1. Save and exit.

**On Linux/macOS:**

sudo nano /etc/hosts

Add:

192.168.56.101 example.com

192.168.56.101 test.com

**Step 8: Test in Browser (from Host OS)**

* Open your browser
* Visit:

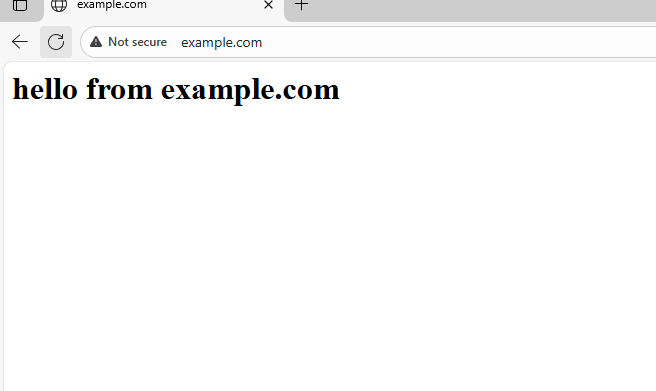
http://example.com

Output:

Hello from example.com

* Visit:

<http://test.com>

 Output:

Hello from test.com

