**Task 2-A : Install Apache2 on Ubuntu Linux**

Please follow the below steps in order to install Apache2 on to the Ubuntu Linux.

**Step 1: Update the system repositories.**

Latest version of the software can be downloaded by first updating the local package index of Ubuntu repositories. Then, open the terminal and enter the following command as shown below in the screenshot.

Command: $ sudo apt update

A screenshot of a computer

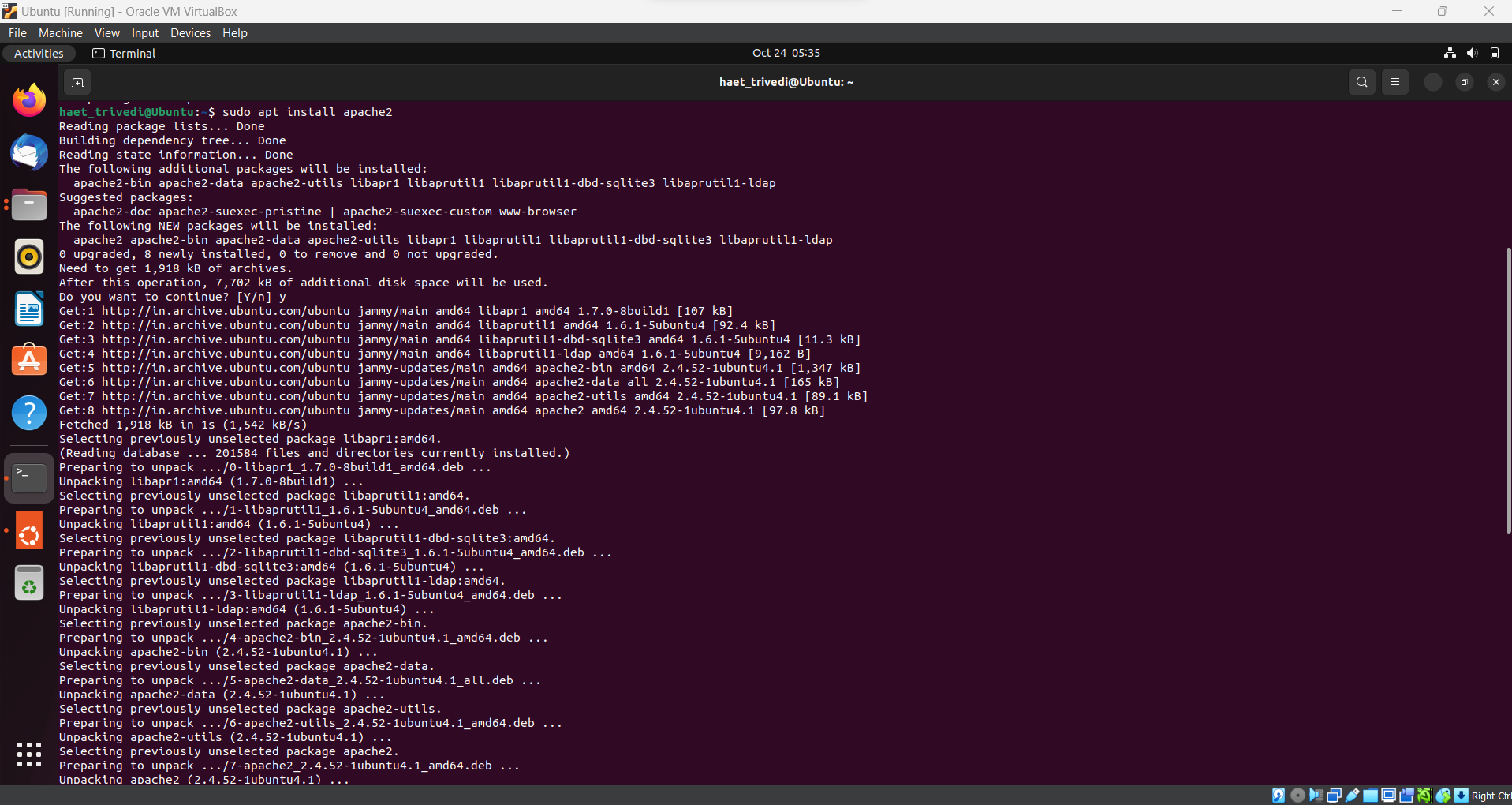
Description automatically generated with medium confidence

**Step 2: Install Apache2 by using the apt command**

Next, enter the following command that starts with the sudo keyword in order to install Apache2 and all its essential dependencies.

While installing the Apache2, you may be prompted with a y/n option to continue the installation. Please enter Y, to continue the installation process.

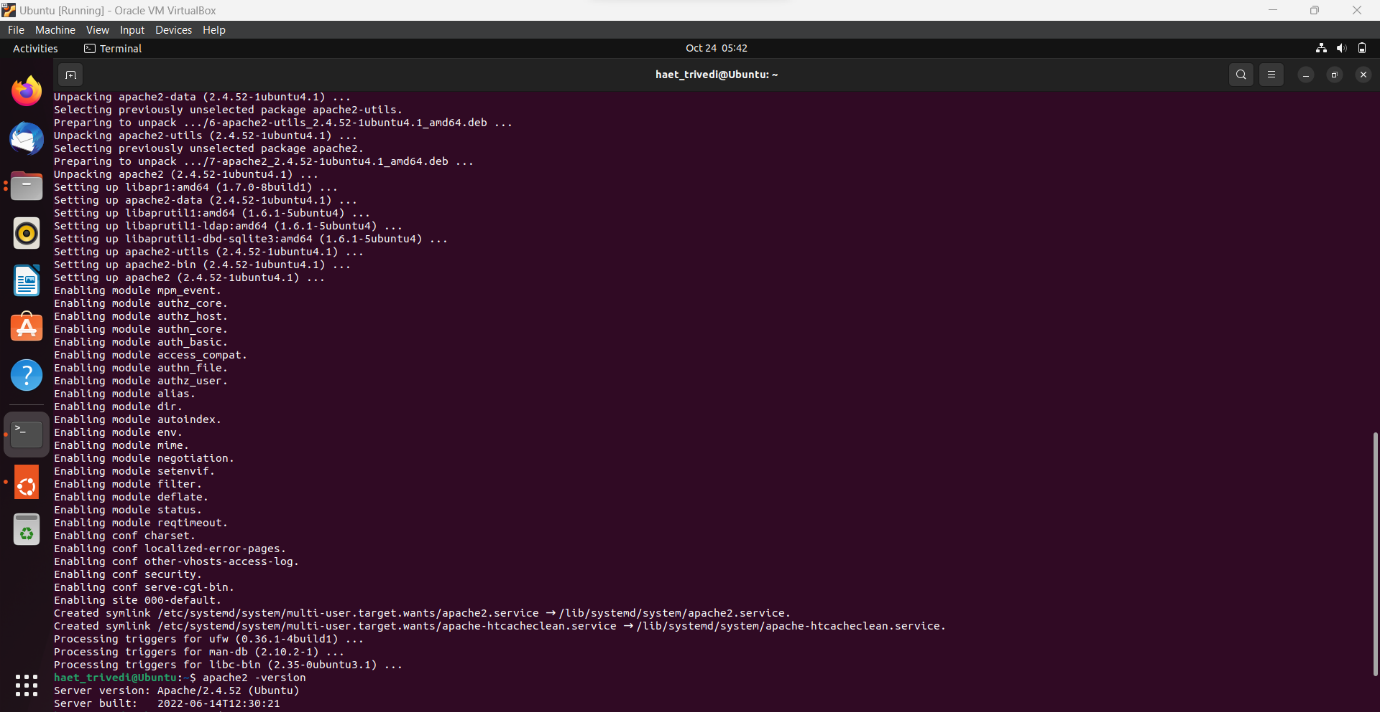
Command: $ sudo apt install apache2



**Step 3: Validate the Apache Installation**

Once, the installation is completed, you can verify the version number of the installed Apache2 software in your system by using the following command as shown below.

Command: $ apache2 -version



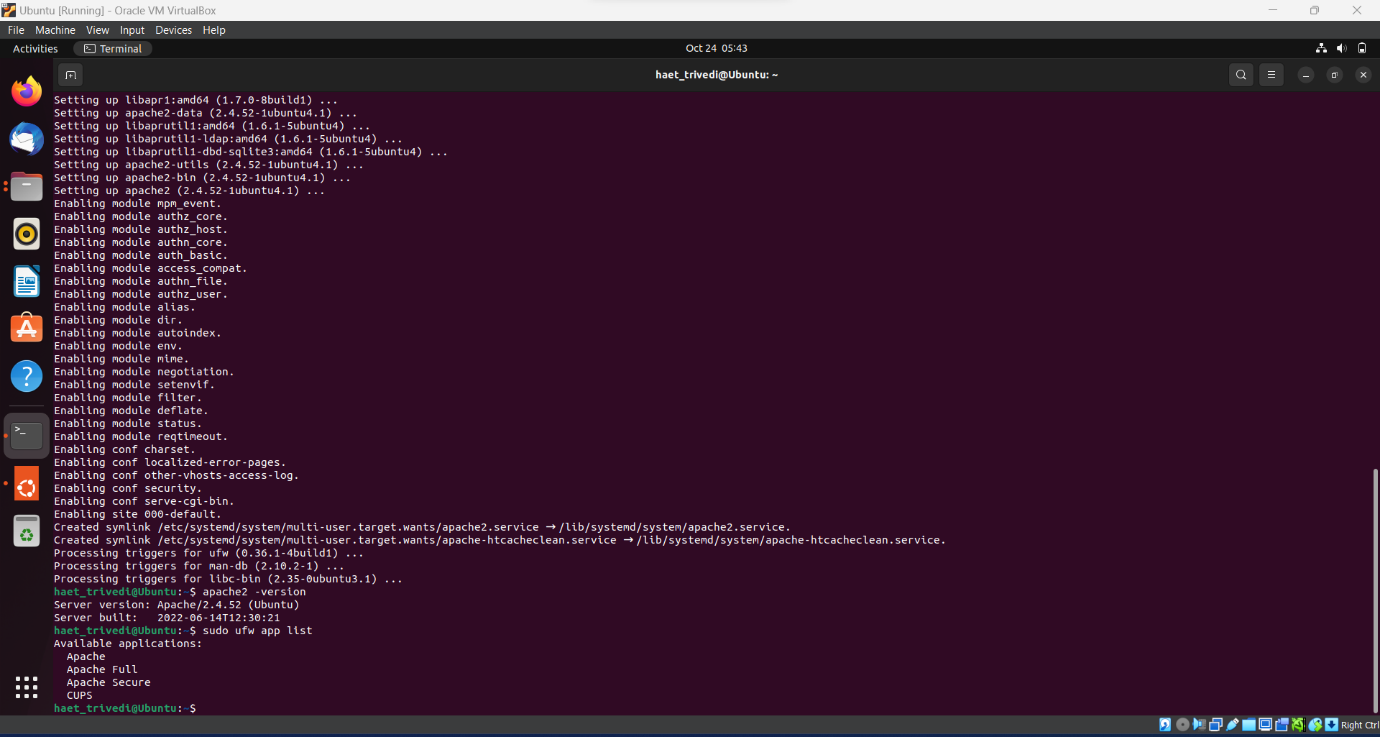
**Configure the Firewall Settings**

To configure Apache, we first need to allow outside access to certain web ports of our system and allow Apache on your UFW firewall.

**Step 1: List the UFW Application Profiles.**

Firewall configuration is done by listing the application profiles. This can be achieved by enabling the access to the Apache using the following command.

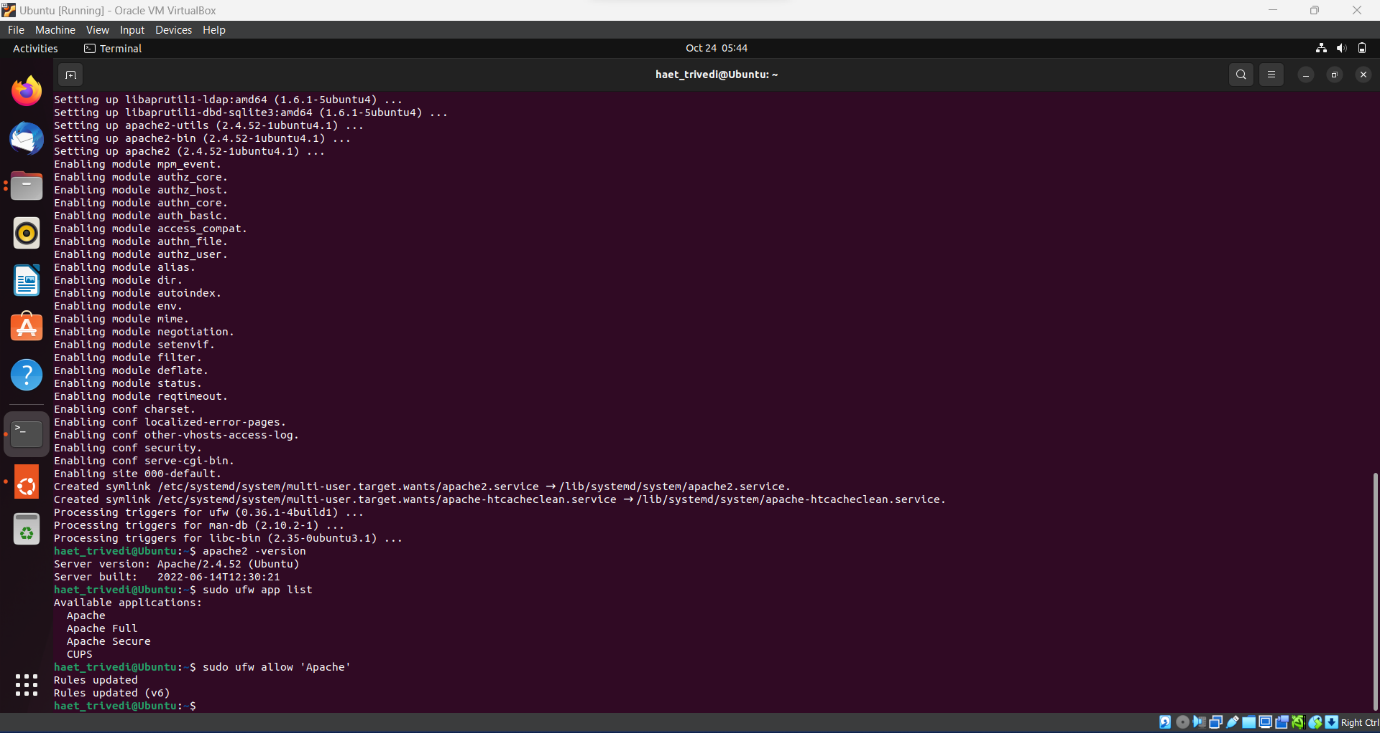
Command: $ sudo ufw app list



**Step 2 : Allow Apache on UFW and verify its status.**

Allowing Apache on UFW will open the port number 80 for the network traffic, while providing the maximum security to the server by using the following command as shown in the screenshot.

Command: $ sudo ufw allow 'Apache'



The status of the UFW will now display Apache enabled on the firewall by using the below command:

Command: $ sudo ufw status

A screenshot of a computer

Description automatically generated with medium confidence

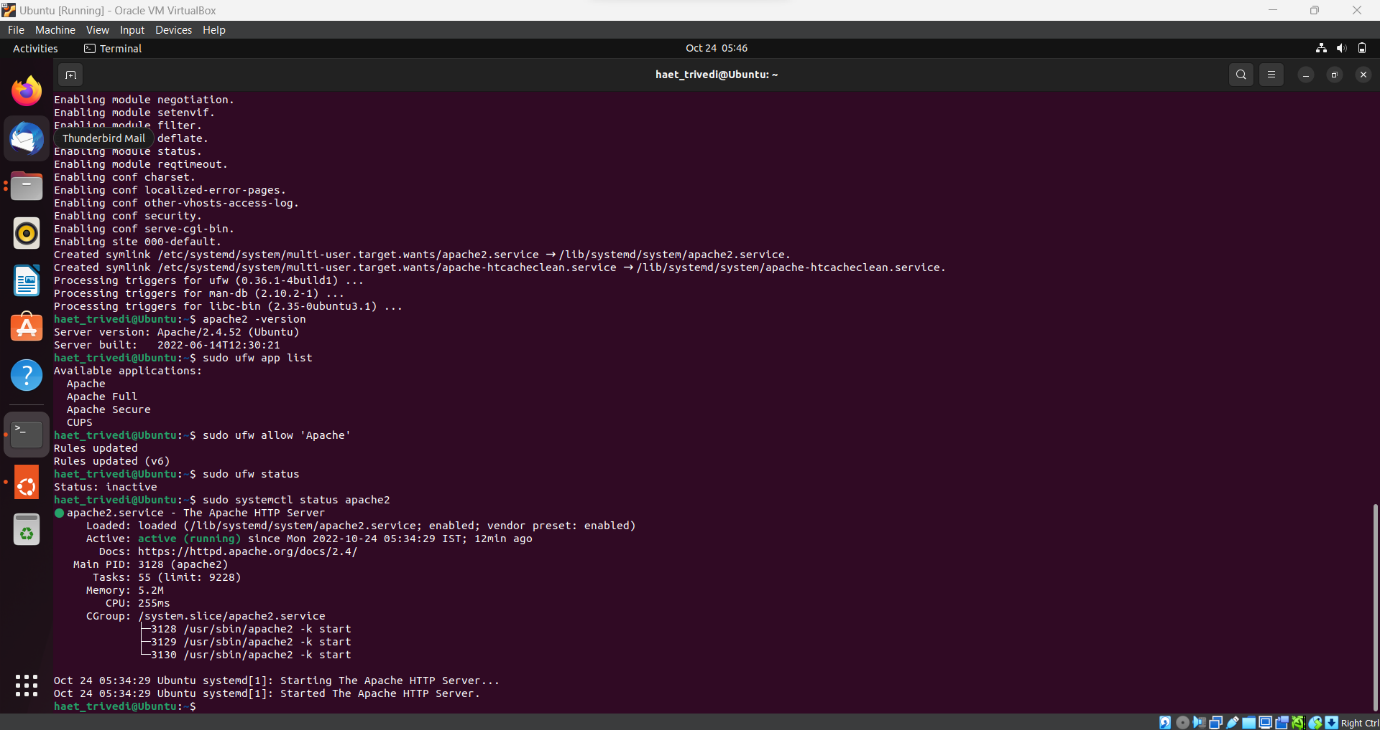
**Configure the Apache Web Server Settings.**

**Step 1: Verify that the Apache Service is running.**

Use the below command to verify that the Apache2 service is up and running on your system.

The status active(running) verifies that the Apache2 service is running.

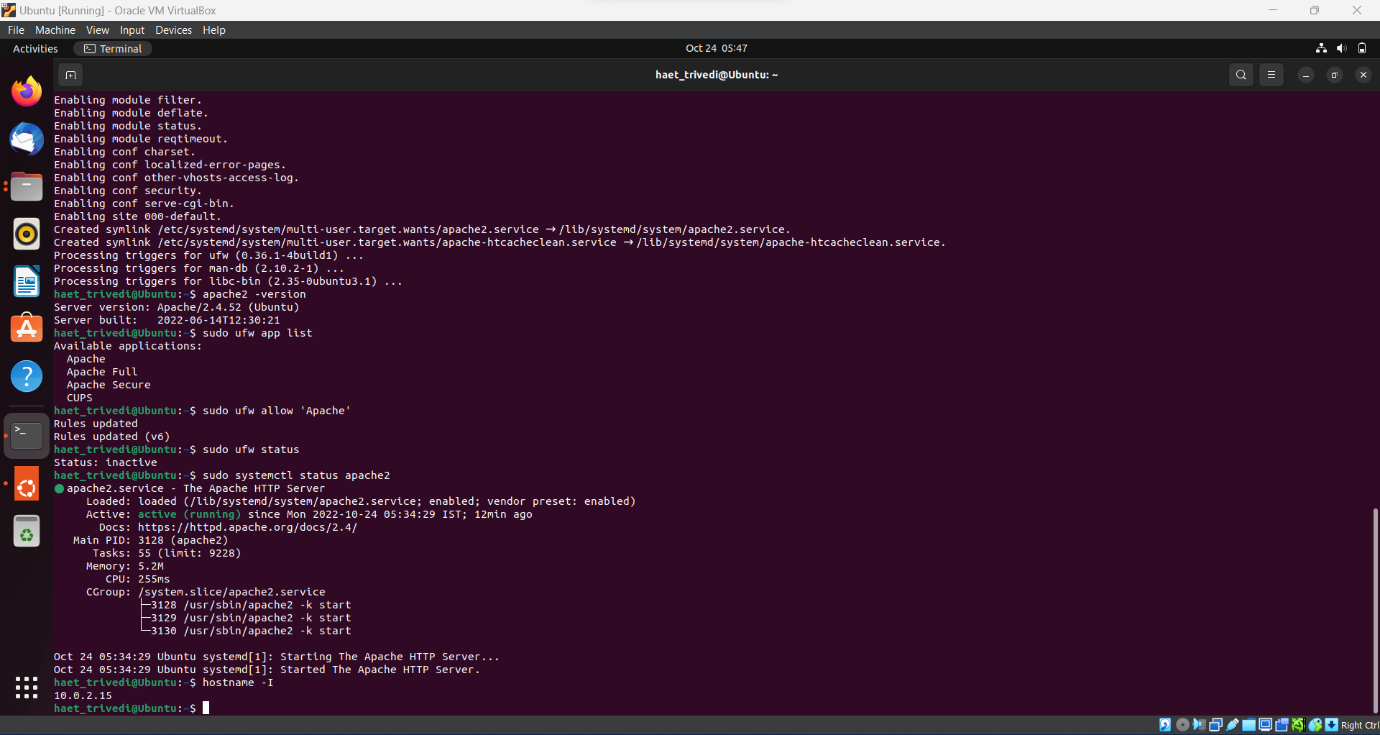
Command: $ sudo systemctl status apache2



**Step 2: Verify that the Apache2 is running properly and listens on your IP Address.**

First, use below command to know your server’s IP Address.

Command: $ hostname -I



Next, you need to verify the IP address in your web browser by inserting the below URL.

http://10.0.2.15

Graphical user interface, text, application, Word

Description automatically generated

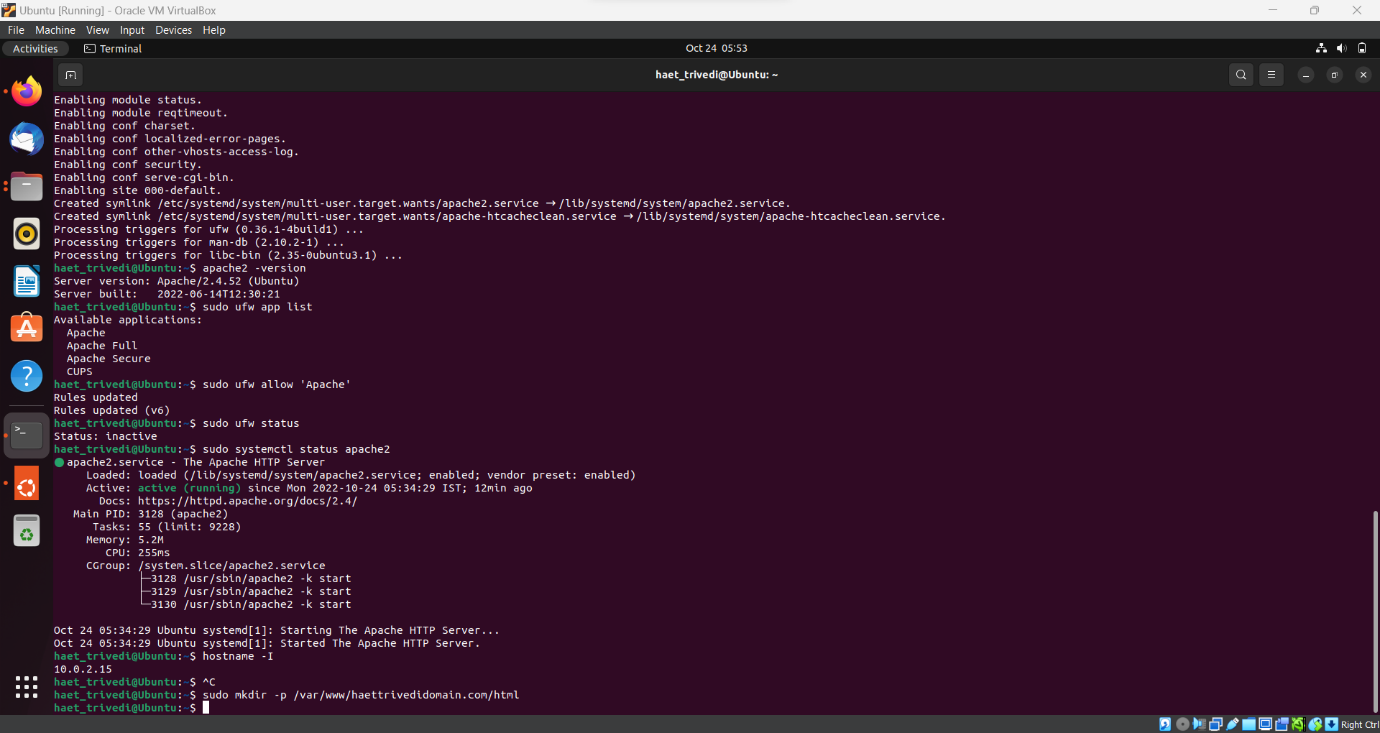
**Set up Virtual Hosts in Apache**

Virtual Host is used to manage configurations for more than one domain from one server. For example, we will setup a website named haettrivedidomain.com by using the server block that is enabled by default in Apache for Ubuntu.

**Step 1: Set up a domain name.**

Create a directory through the following command and use your own domain name.

Command: sudo mkdir -p /var/www/sampledomain.com/html



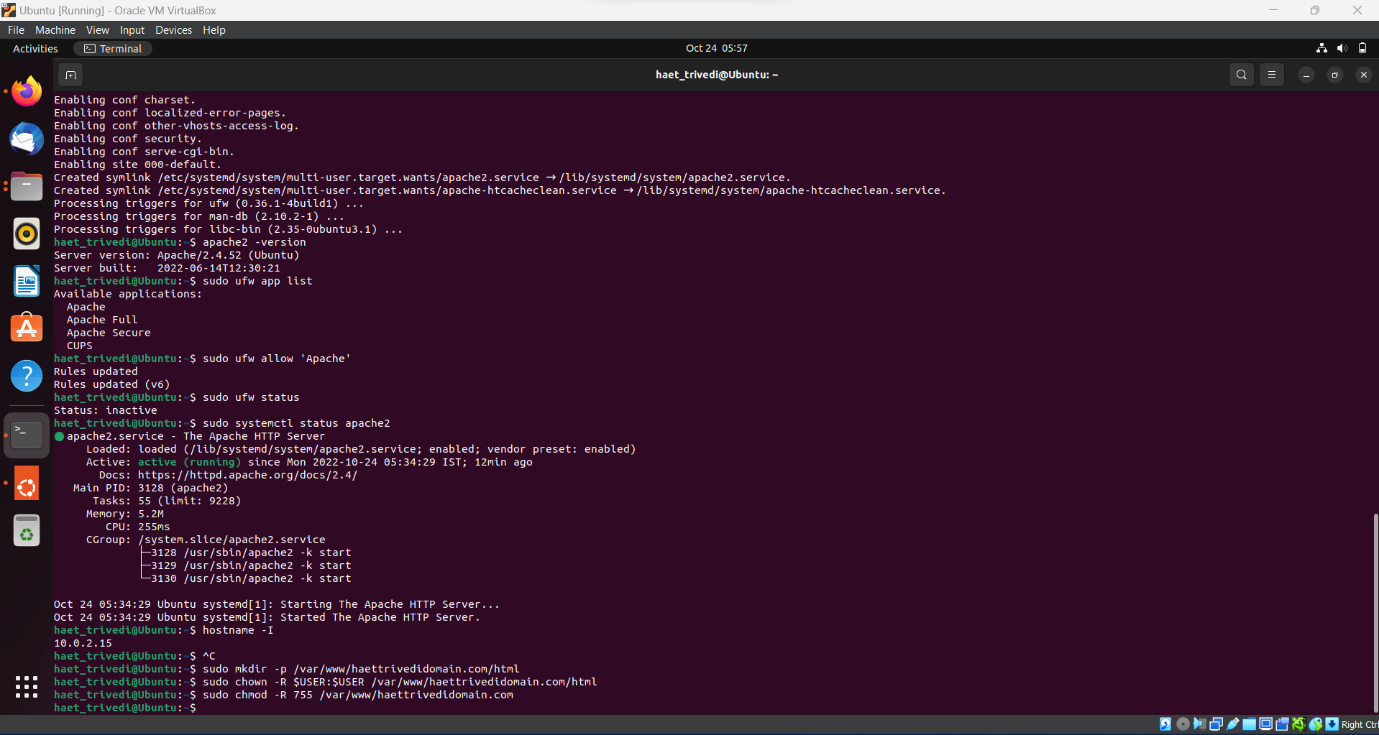
Then, assign the ownership of the directory through the following commands.

Command : sudo chown -R $USER:$USER /var/www/sampledomain.com/html

sudo chmod -R 755 /var/www/sampledomain.com

A screenshot of a computer

Description automatically generated



Now create an index page that we can later access to test if Apache is running our domain name. Create an HTML file by using either Nano editor or any other text editor.

Command: $ nano /var/www/sampledomain.com/html/index.html

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

You can then save the file in nano by using Ctrl + X and then enter Y and hitting enter.

Apache needs the virtual host file to serve the contents of your server. The default configuration file for this purpose is already created but we will make a new one for the custom configurations.

Command: $ sudo nano /etc/apache2/sites-available/sampledomain.com.conf

A screenshot of a computer

Description automatically generated

Text

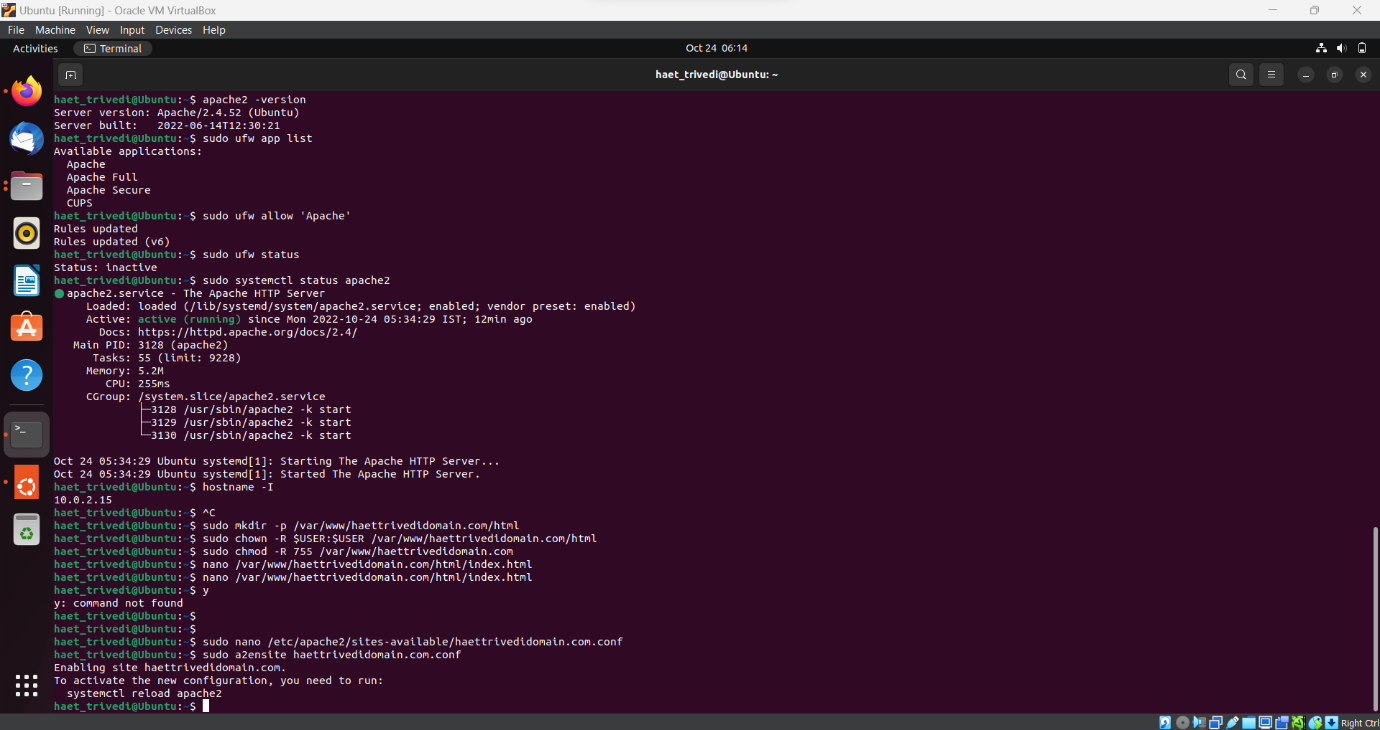
Description automatically generated

You can then save the file in nano by using Ctrl + X and then enter Y and hitting enter.

**Step 2: Enable the domain configuration file.**

Enable the domain configuration file by using the a2ensite tool.

Command: $ sudo a2ensite sampledomain.com.conf



The output will suggest activating the new configuration, but we can do it all collectively after running the following command that is used to disable the original configuration file.

Command: $ sudo a2dissite 000-default.conf

A screenshot of a computer

Description automatically generated

Now, restart the Apache service.

Command: $ sudo systemctl restart apache2

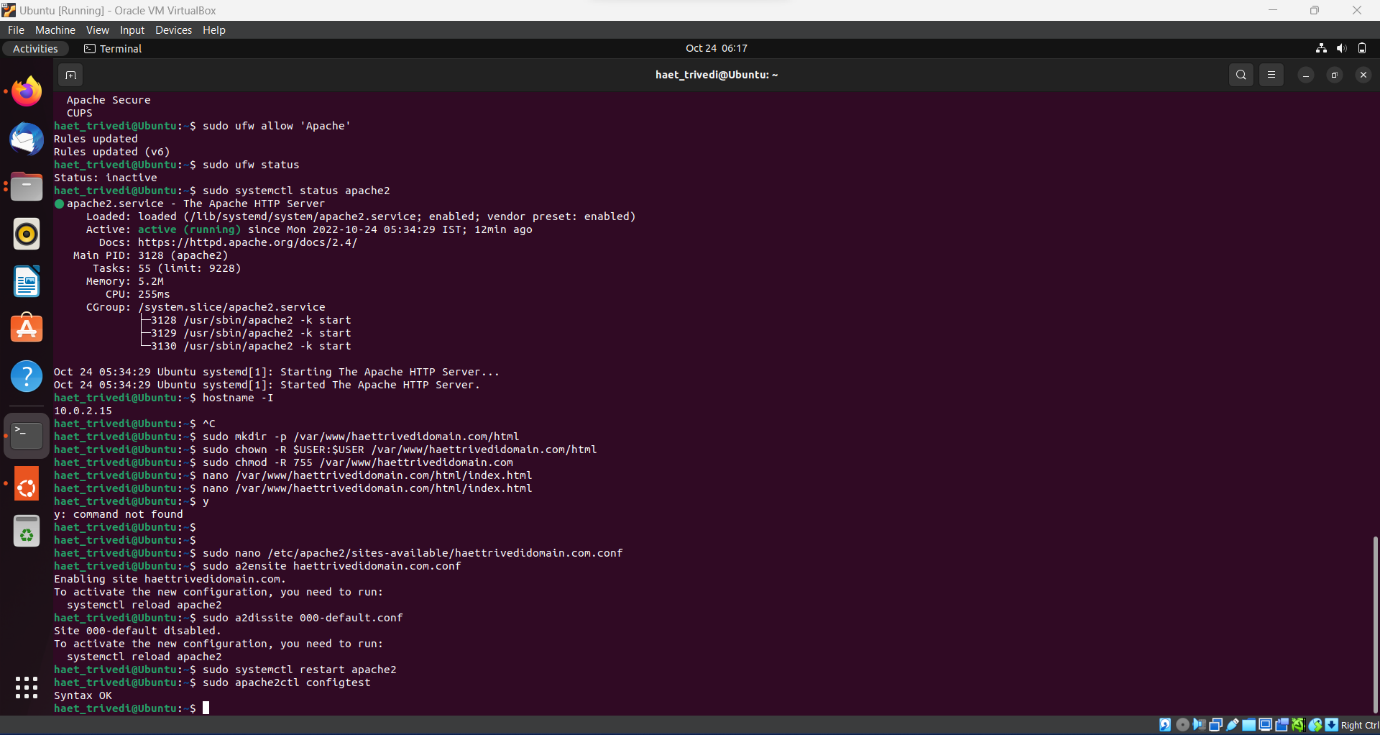
A screenshot of a computer

Description automatically generated

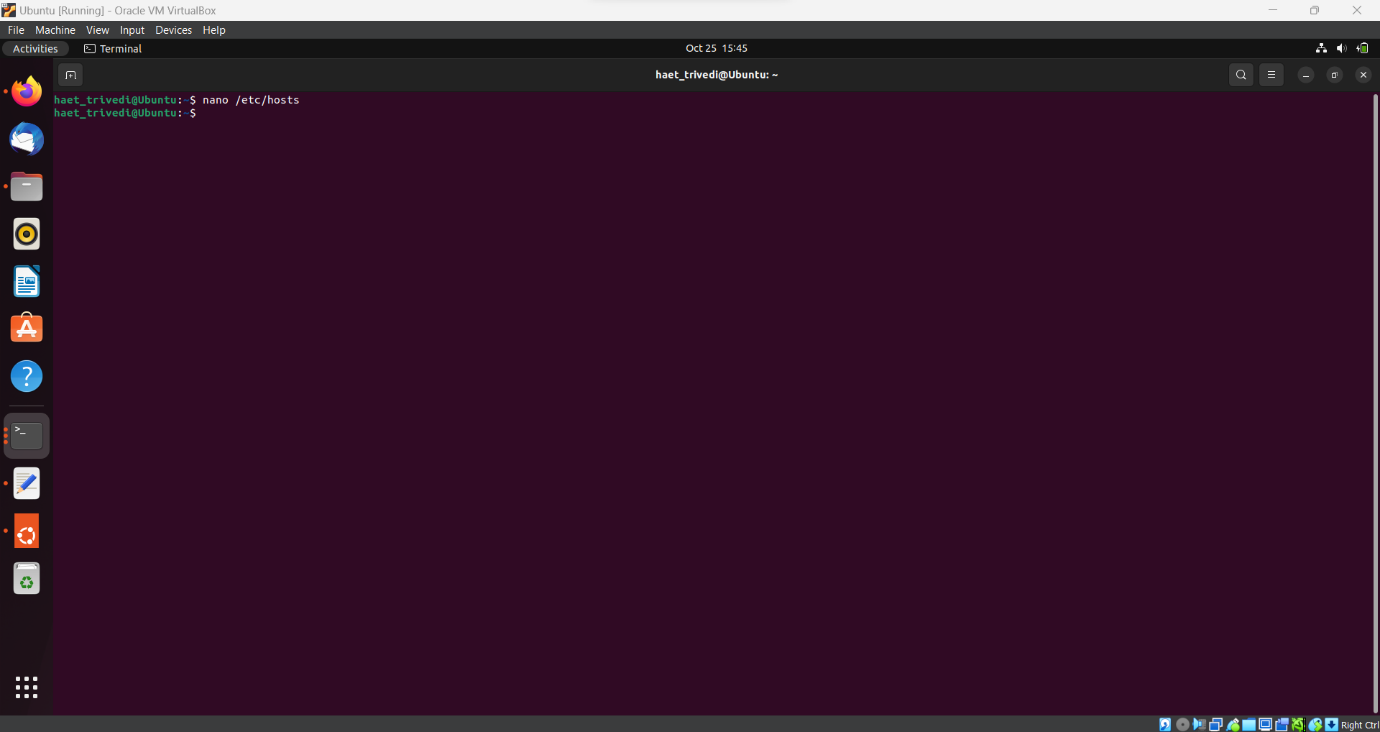
**Step 3 : Test for errors and edit etc/hosts file configuration**

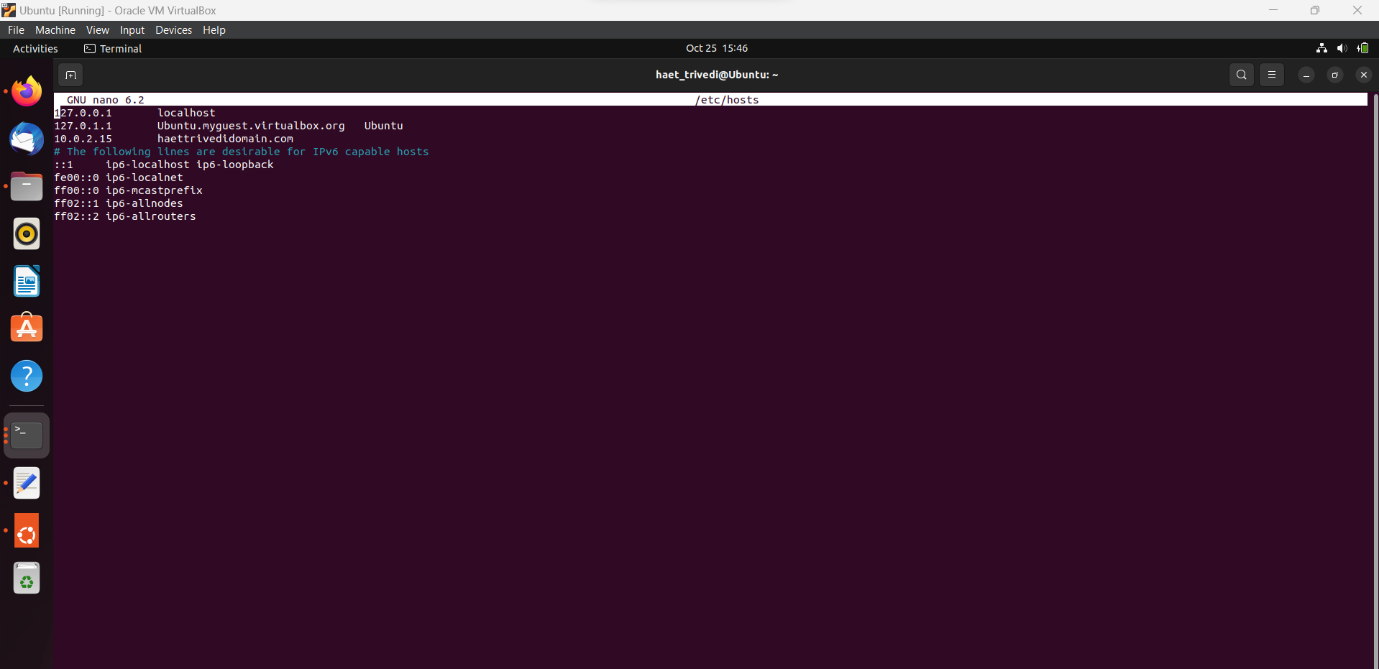
Use the following command to test the configuration errors.

Command: $ sudo apache2ctl configtest



Then, open the /etc/hosts file and edit the file by adding the IP address and the custom domain name.





**Step 4: Test if Apache is serving your domain name.**

Apache server is now configured to serve your domain name. This can be verified by entering your server name in any of the web browsers running on your system.

URL : <http://haettrivedidomain.com>

Here, in the below code the title is “Welcome to the Programming and System development Course!

And, the Body includes the Student Name and Student ID.

Graphical user interface, text

Description automatically generated with medium confidence

**Task 2-B : Enabling SSH on Ubuntu**

By default on Ubuntu desktop systems the SSH server is not installed, but it can be easily installed from the standard Ubuntu repositories.

To install and enable SSH on your Ubuntu system complete the following steps:

**Step 1: Open the terminal and install the openssh-server package by using the following command.**

Command: $ sudo apt update

$ sudo apt install openssh-server

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**Step 2: After the installation is completed, the SSH service will start automatically. To verify that the installation was successful and SSH service is up and running type the following command which will print the SSH server status.**

Command: $ sudo systemctl status ssh

A screenshot of a computer

Description automatically generated with medium confidence

**Step 3: Ubuntu includes the firewall configuration tool called UFW. If the firewall is enabled on your system, make sure to open the SSH port.**

Command: $ sudo ufw allow ssh

A screenshot of a computer

Description automatically generated with medium confidence

**Step 4: Connecting to SSH over LAN by using the following command.**

Command: $ ssh [haet\_trivedi@10.0.2.15](mailto:haet_trivedi@10.0.2.15)

A screenshot of a computer

Description automatically generated with medium confidence

**Conclusion:**

By following the above steps, I have learned to install and configure the Apache Web Server on the Ubuntu System. Additionally, I learned how to install and enable SSH on the Ubuntu System.