**Name: Vishnu Sadasivan**

**Roll No: 52**

**Batch: MCA-B**

**Date: 02-06-2022**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 26**

**Aim**

To install Linux Apache,MySQL,PHP(LAMP) stack on ubuntu and also install MySql and Wordpress

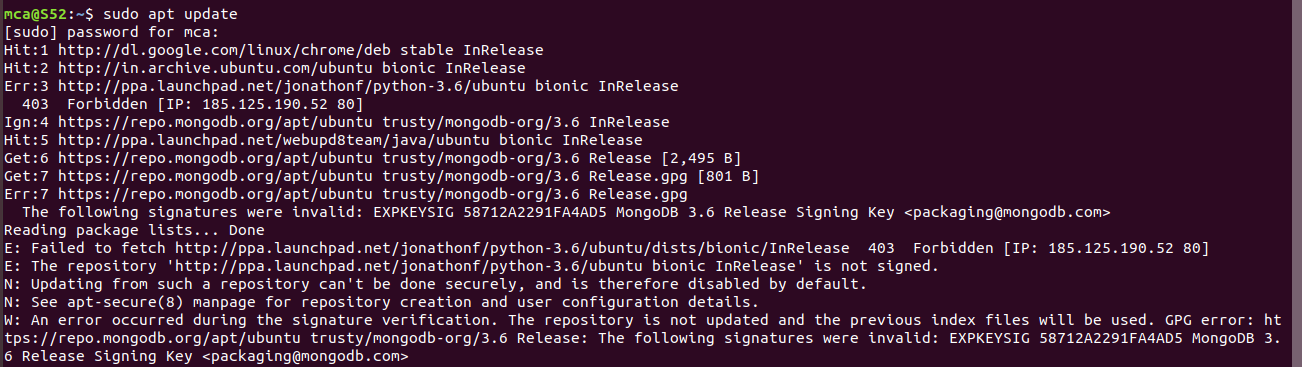
**Procedure :**

**Step 1: Installing Apache and updating firewall**

1.Make sure apt class is updated

**$ sudo apt update**

**Output:-**

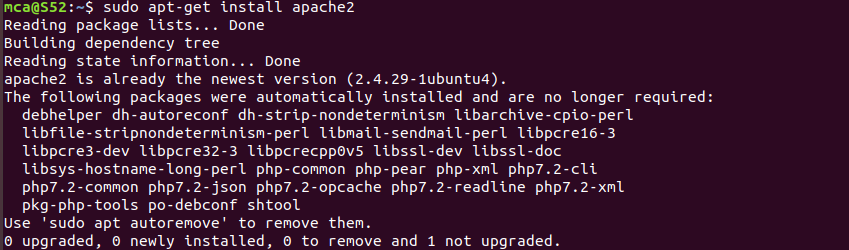


2.Once the apt is updated, install Apache

After entering this command, apt will tell you which packages it plans to install and how much extra disk space they’ll take up. Press Y and hit ENTER to confirm, and the installation will proceed.

**$ sudo apt install apache2**

**Output:-**



3.Adjust the firewall to allow web traffic

Next, assuming that you have followed the initial server setup instructions and enabled the UFW firewall, make sure that your firewall allows HTTP and HTTPS traffic. You can check that UFW has an application profile for Apache like so

**$ sudo ufw app list**

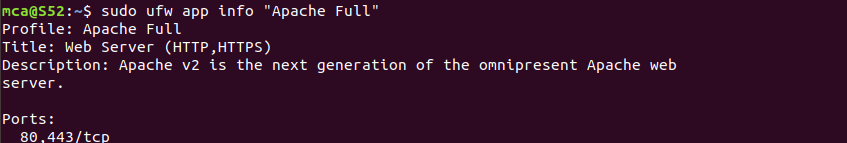
**Output:-**



4. If you look at the Apache Full profile details, you’ll see that it enables traffic to ports 80 and 443:

**$ sudo ufw app info “Apache Full”**

**Output:-**



5. To allow incoming HTTP and HTTPS traffic for this server, run:

**$ sudo ufw allow “Apache full”**

**Output:-**

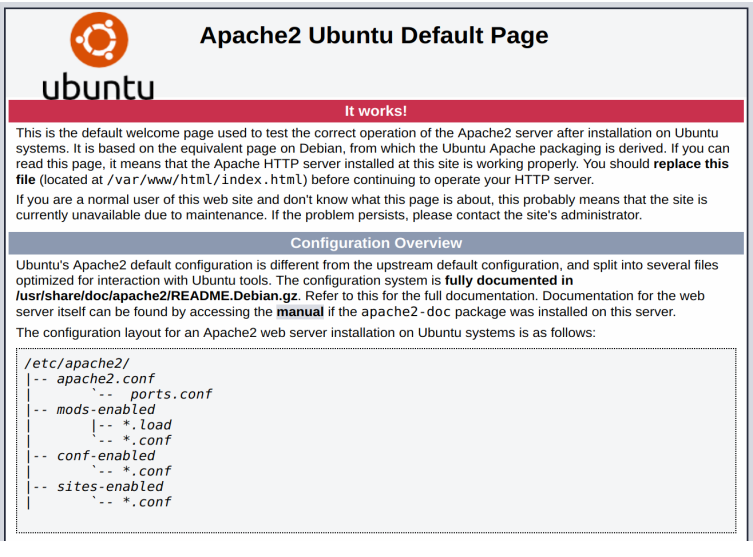


6. You can do a spot check right away to verify that everything went as planned by visiting your server’s public IP address in your web browser:

**http://your\_server\_ip**

**http://localhost**

**Output:-**

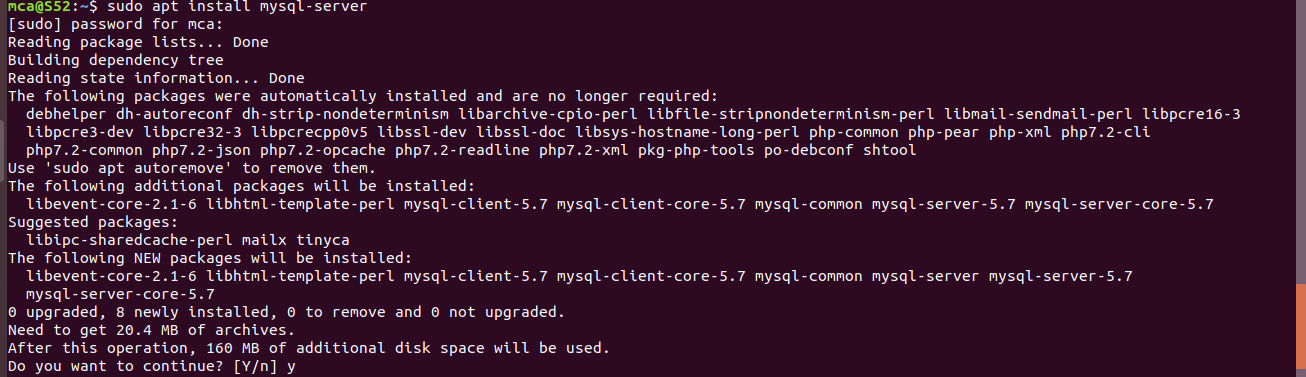


**Step 3: Installing MySQL**

1. Again , apt to acquire and install this software

**$ sudo apt install mysql-server**

**Output:-**



1. When the installation is complete, run a simple security script that comes pre-installed with MySQL which will remove some dangerous defaults and lock down access to your database system

This will ask if you want to configure the VALIDATE PASSWORD PLUGIN. Answer Y for yes, or anything else to continue without enabling.

**$ sudo mysql\_secure\_installation**

**Output:-**

1. When you’re finished, test if you’re able to log in to the MySQL console by typing sudo mysql,This will connect to the MySQL server as the administrative database user root, which is inferred by the use of sudo when running this command:

**$ sudo mysql**

**Output:-**

1. To exit MySQL console,

**$ exit**

**Step 4: Installing PHP**

1. In addition to the php package, you’ll also need libapache2-mod-php to integrate PHP into Apache, and the php-mysql package to allow PHP to connect to MySQL databases. Run the following command to install all three packages and their dependencies

**$ sudo apt install php libapache2-mod-php php-mysql**

**Output:-**

2. Apache web server in order for your changes to be recognized. You can do that with the following command:

**$ sudo systemctl restart apache2**

**Output:-**

**Step 5: Testing The PHP processing on your web server**

1. In order to test that your system is properly configured for PHP, create a PHP script called info.php. In order for Apache to find this file and serve it correctly, it must be saved to your web root directory. Create the file at the web root you created in the previous step by running:

**$ sudo nano /var/www/your\_domain/info.php**

This will open a blank file. Add the following text, which is valid PHP code, inside the file

**Output:-**

2.When you are finished, save and close the file. Now you can test whether your web server is able to correctly display content generated by this PHP script. To try this out, visit this page in your web browser. You’ll need your server’s public IP address or domain name again. The address you will want to visit is:

**http://your\_domain/info.php**

The page that you come to should look something like this: