A Sample Project for Linux Administrators

Project Name: - Configure the LAMP Stack

LAMP stands for Linux Apache MySQL and PHP. It's a very common architecture for web applications. If you think about the LAMP stack in broad terms you have an operating system (Linux), a web server (Apache), a database tier (MySQL), and a programming language (PHP)

1. Linux
2. Apache
3. MySQL
4. PHP

Step 1 — Installing Apache and Updating the Firewall

The Apache web server is among the most popular web servers in the world. It's well documented, has an active community of users, and has been in wide use for much of the history of the web, which makes it a great choice for hosting a website.

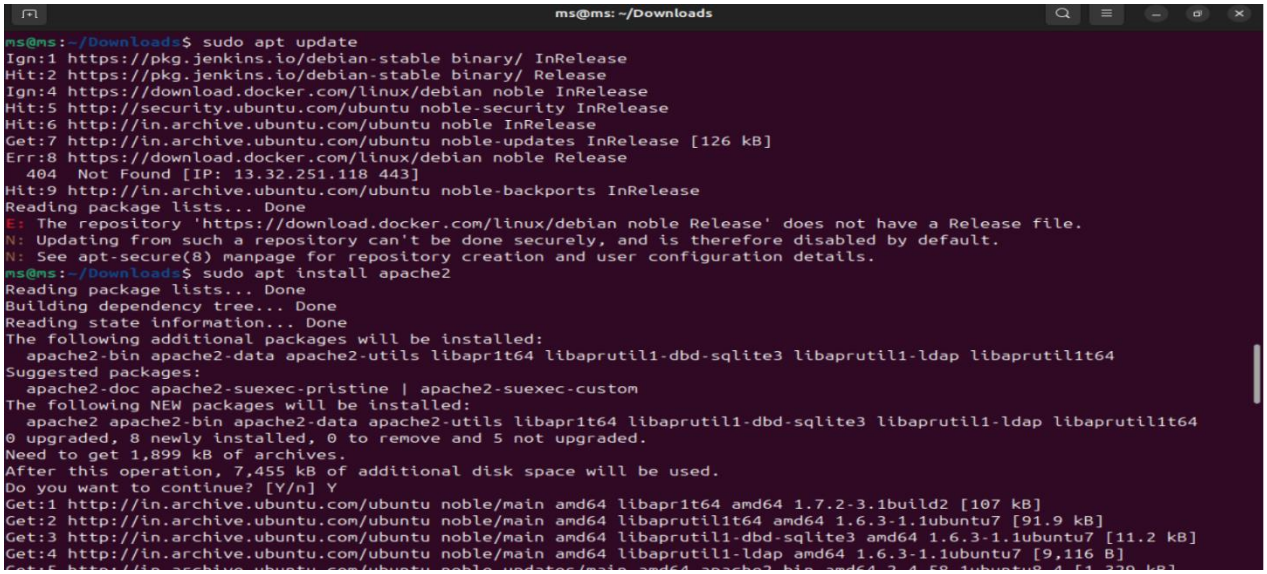
Start by updating the package manager cache. If this is the first time, you're using sudo within this session, you'll be prompted to provide your user's password to confirm you have the right privileges to manage system packages with apt

```
sudo apt update
```

Now I am installing Apache

```
sudo apt install apache2
```

You'll be prompted to confirm Apache's installation. Confirm by pressing Y, then ENTER.



```
sudo ufw app list
```

See the Output

```
ms@ms:~/Downloads$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
```

```
sudo ufw allow in "Apache"

sudo ufw status: - If showing Inactive Then

sudo ufw allow in “Apache”
```

```
ms@ms:~$ sudo ufw allow in "Apache"
[sudo] password for ms:
Rules updated
Rules updated (v6)
ms@ms:~$ sudo ufw status
Status: inactive
ms@ms:~$ sudo ufw status
Status: inactive
ms@ms:~$ sudo ufw allow in "Apache"
Skipping adding existing rule
Skipping adding existing rule (v6)
ms@ms:~$ sudo ufw status
Status: inactive
ms@ms:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ms@ms:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
Apache                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

Install MySQL

```
sudo apt install mysql-server -y
```

```
sudo mysql_secure_installation
```

Follow the prompts to set a root password and remove insecure defaults.

```
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No:
```

```
There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
```

```
Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for
```

Install PHP

```
sudo apt install php libapache2-mod-php php-mysql -y
```

Step 5: Test PHP

Create a PHP info file to test if PHP is working correctly:

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

```
ms@ms:~$ echo "<?php echo 'Hello, LAMP\!'; ?>" | sudo tee /var/www/html/test.php
<?php echo 'Hello, LAMP\!'; ?>
```

Changing Apache's Directory Index (Optional)

In some cases, you'll want to modify the way that Apache serves files when a directory is requested. Currently, if a user requests a directory from the server, Apache will first look for a file called index.html. We want to tell the web server to prefer PHP files over others, to make Apache look for an index.php file first. If you don't do that, an index.html file placed in the document root of the application will always take precedence over an index.php file.

To make this change, open the dir.conf configuration file in a text editor of your choice. Here, we'll use nano:

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

It will look like this:

/etc/apache2/mods-enabled/dir.conf

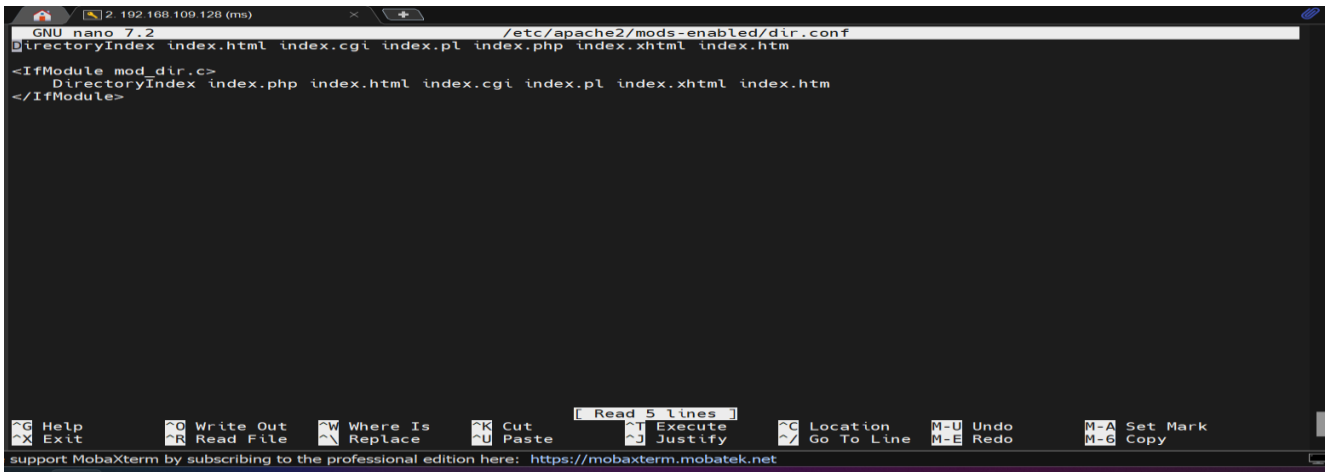
```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

Move the PHP index file (highlighted above) to the first position after the DirectoryIndex specification, like this:

/etc/apache2/mods-enabled/dir.conf

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

When you are finished, save and close the file by pressing CTRL+X. Confirm the save by typing Y and then hit ENTER to verify the file save location.



Install Additional PHP Modules (Optional)

Depending on your needs, you might want to install additional PHP modules. Here are some

```
sudo apt install php-cli php-curl php-gd php-mbstring php-xml php-zip -y
```

```
ms@ms:~$ sudo apt install php-cli php-curl php-gd php-xml php-mbstring
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  php8.3-curl php8.3-gd php8.3-mbstring php8.3-xml
The following NEW packages will be installed:
  php-cli php-curl php-gd php-mbstring php-xml php8.3-curl php8.3-gd php8.3-mbstring php8.3-xml
0 upgraded, 9 newly installed, 0 to remove and 5 not upgraded.
Need to get 721 kB of archives.
After this operation, 2,115 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu noble/main amd64 php-cli all 2:8.3+93ubuntu2 [4,584 B]
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-curl amd64 8.3.6-0ubuntu0.24.04.1 [40.3 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu noble/main amd64 php-curl all 2:8.3+93ubuntu2 [1,836 B]
Get:4 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-gd amd64 8.3.6-0ubuntu0.24.04.1 [31.2 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu noble/main amd64 php-gd all 2:8.3+93ubuntu2 [1,830 B]
Get:6 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-mbstring amd64 8.3.6-0ubuntu0.24.04.1 [512 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 php-mbstring all 2:8.3+93ubuntu2 [1,848 B]
Get:8 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-xml amd64 8.3.6-0ubuntu0.24.04.1 [126 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu noble/main amd64 php-xml all 2:8.3+93ubuntu2 [1,856 B]
Fetched 721 kB in 1s (540 kB/s)
Selecting previously unselected package php-cli.
(Reading database ... 156934 files and directories currently installed.)
Preparing to unpack .../0-php-cli_2%3a8.3+93ubuntu2_all.deb ...
Unpacking php-cli (2:8.3+93ubuntu2) ...
```

Restart Apache

After making changes, restart the Apache server:

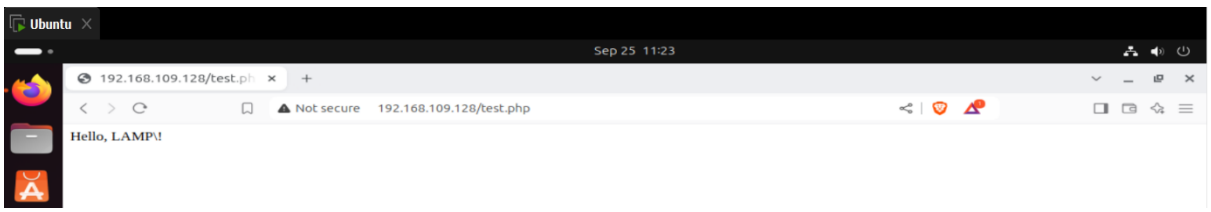
```
sudo systemctl restart apache2
Now You will check your server ip by hostname -I
```

```
ms@ms:~$ hostname -I
192.168.109.128 172.17.0.1
```

Test Everything

1. Check Apache by visiting http://your_server_ip.

   http://192.168.109.128/test.php



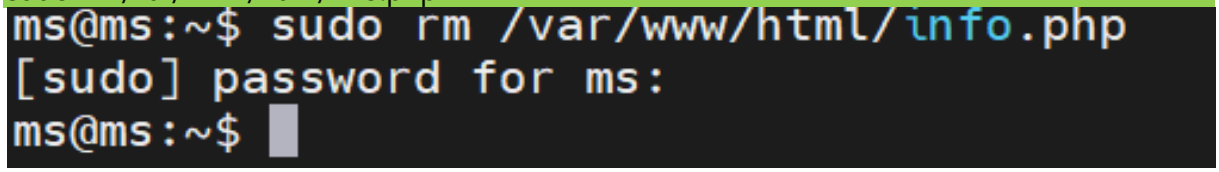
2. Check PHP by visiting http://your_server_ip/info.php.
3. To interact with MySQL, log in using:

```
sudo mysql -u root -p
```

Cleanup

Remove the info.php file for security reasons:

```
sudo rm /var/www/html/info.php
```



Thanks