# How To Install Apache2 (LAMP) Ubuntu 18.04

How to install apache2 (LAMP) server

LAMP is a acronym of the names in the applications stack, Linux OS, Apache server, MySQL database and PHP programming language.

Together they build a framework to run web applications and host sites like WordPress , all applications in the stack are open source and released on most Linux distributions.

**Requirements**

1. Ubuntu 18.04 LTS
2. SSH access to the server (Setup SSH)
3. A non root user with sudo privileges (Add sudo user)
4. Enabled firewall (Setup ufw)

5. Configured hostname (Setup hostname)
6. DNS entries

## Step 1: Install Apache2

**1.1** Lets start by updating the repository's and software packages.

```
sudo apt update
sudo apt upgrade -y
```

**1.2** Install the apache2 package.

```
sudo apt install apache2 -y
```

**1.3** Confirm installation.

```
sudo systemctl status apache2
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2019-06-19 19:08:58 UTC; 3min 21s ago
 Main PID: 7685 (apache2)
    Tasks: 55 (limit: 2213)
   CGroup: /system.slice/apache2.service
           ├─7685 /usr/sbin/apache2 -k start
           ├─7878 /usr/sbin/apache2 -k start
           └─7879 /usr/sbin/apache2 -k start
```

```
Jun 19 19:08:48 iphone systemd[1]: Starting The Apache HTTP Server...
Jun 19 19:08:58 iphone apachectl[7660]: AH00558: apache2: Could not reliably det
Jun 19 19:08:58 iphone systemd[1]: Started The Apache HTTP Server.

toor@iphone:~$
```

## Step 2: Configure Firewall

**2.1** Add firewall rules for Apache.

```
sudo ufw allow in "Apache Full"
```

```
Rule added
Rule added (v6)
toor@iphone:~$
```

**2.2** Display firewall rules and confirm that the firewall is configured

```
sudo ufw status
```

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
```

```
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)

toor@iphone:~$
```

**2.3** Confirm that you can browse to the site

```
http://your_server_ip
```

## Step 3: Create the Directory Structure

**3.1** Virtual host enables us to have multiple websites on one server, each website can have its own home folder “document root” and a unique SSL certificate ,we can have different security policies for each site, and much more.

Create the directory structure.

```
/var/www/
├── Domain-1.local
│ └── html
├── Domain-2.local
│ └── html
```

Before we create the directory for the site, make sure to configure hostname and hosts file.

I will create a website for my local lab domain ceh.local, the /etc/hots file should the the fallowing entry in it.

```
YOUR-IP-ADDRESS ceh.local
```

In this example i am creating a virtual hosts directory called “ceh.local” and i am using the -p flag to create parent directories.

```
sudo mkdir -p /var/www/ceh.local/html
```

**3.2** Assign ownership of the directory to current user

```
sudo chown -R $USER:$USER /var/www/ceh.local/html
```

**3.3** Set directory permissions

```
sudo chmod -R 755 /var/www/ceh.local/html
```

**3.4** Create a sample index.html file using your favorite editor and add it to the root directory.

```
sudo nano /var/www/ceh.local/html/index.html
```

Add the html code bellow

```
<!DOCTYPE html>
```

```
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Welcome to ceh.local</title>
  </head>
  <body>
    <h1>Success! ceh.local home page</h1>
  </body>
</html>
```

**Exit & Save**

## Step 4: Configure Virtual Hosts File

Apache virtual hosts configuration files are stored in.

- /etc/apache2/sites-enabled
- /etc/apache2/sites-available

Lets add a Virtual Hosts configuration file for domain1.local

```
sudo nano /etc/apache2/sites-available/ceh.local.conf
```

Add the fallowing lines and modify them to your site.

```
<VirtualHost *:80>
    ServerName ceh.local
    ServerAlias www.ceh.local
    ServerAdmin admin@ceh.local
    DocumentRoot /var/www/ceh.local/html

    <Directory /var/www/ceh.local/html>
        Options -Indexes +FollowSymLinks
```

```
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/ceh.local-error.log
    CustomLog ${APACHE_LOG_DIR}/ceh.local-access.log combined
</VirtualHost>
```

**Exit & Save**

**4.2** Enable the Virtual Hosts configuration file file with a2ensite

```
sudo a2ensite ceh.local.conf
```

```
Enabling site domain1.local.
To activate the new configuration, you need to run:
  systemctl reload apache2
toor@iphone:~$
```

**4.3** Disable the default site defined in 000-default.conf

```
sudo a2dissite 000-default.conf
```

```
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
toor@iphone:~$
```

**4.4** Test the configuration file for any syntax errors

```
sudo apache2ctl configtest
```

```
Syntax OK
toor@iphone:/var/www/domain1.local/html$
```

**4.5** Restart the Apache service for the changes to take effect

```
sudo systemctl restart apache2
```

**4.6** Confirm that the service have started

```
sudo systemctl status apache2
```

**4.7** launch a web browser and start browsing ceh.local

## Step 5: Install MySQL

**5.1** To install MySQL run

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

**5.2** Verify that MySQL service is running

```
sudo systemctl status mysql
```

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Fri 2019-06-21 11:59:29 UTC; 8s ago
 Main PID: 17807 (mysqld)
    Tasks: 27 (limit: 2322)
   CGroup: /system.slice/mysql.service
           └─17807 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi

Jun 21 11:59:28 srv6 systemd[1]: Starting MySQL Community Server...
Jun 21 11:59:29 srv6 systemd[1]: Started MySQL Community Server.
lines 1-10/10 (END)
toor@srv6:~$
```

**5.3** The default MySQL user “root” have a blank password., we need to secure the MySQL server and remove the default database.

```
sudo mysql_secure_installation
```

Then enter the following security questions

- VALIDATE PASSWORD plugin = NO
- Set root password and confirm
- Remove anonymous users? = YES
- Disallow root login remotely? = NO
- Remove test database and access to it? = YES
- Reload privilege tables now? = YES

**5.4** Start from MySQL Server 5.7, if you do not provide a password to root user during the installation, it will use auth_socket plugin for authentication.

If we want to configure a password authentication, we need to run the following commands.

```
sudo mysql
```

**5.5** Display current configuration

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

**5.6** Alter authentication_string for the root user

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'changeme';
```

**5.7** Flush the privileges and update the changes

```
FLUSH PRIVILEGES;
```

**5.8** Display current configuration

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

**5.9** Exit from the mysql prompt:

```
exit
```

## Step 6: Install PHP

**6.1** PHP is a server side scripting language used to generate dynamic content on websites and applications.

Install PHP (default version is PHP 7.2) and some of the basic modules for web deployments.

```
sudo apt install php php-common php-mysql php-gd php-cli -y
```

**6.2** Create info.php file in the Apache root document folder.

Usually, the apache2 root document folder will be /var/www/html/ or /var/www/ in most Debian based Linux distributions.

If you have followed the guide then the the file should be in /var/www/ceh.local/html/

```
sudo nano /var/www/ceh.local/html/info.php
```

Add the following lines

```
<?php
phpinfo();
?>
```

**Exit and save**

**6.3** Restart Apache

```
sudo systemctl restart apache2
```

**6.4** Test PHP page, open a web browser and enter “http://ceh.local/info.php”

## Step 7: Install PhpMyAdmin

**7.1** With phpMyAdmin we can administrating MySQL from a web browser, start by adding the needed repository.

```
sudo add-apt-repository universe
```

**7.2** Install phpmyadmin

```
sudo apt install phpmyadmin -y
```

Go through the package installation process, select Apache2 and configure a password for the phpmyadmin database.

**7.3** Restart Apache

```
sudo systemctl restart apache2
```

## Conclusion

We have installed installing Apache2, MySQL, PHP and Virtual Hosts on a Ubuntu server and secured it.

For administration of the website we have installed phpmyadmin.