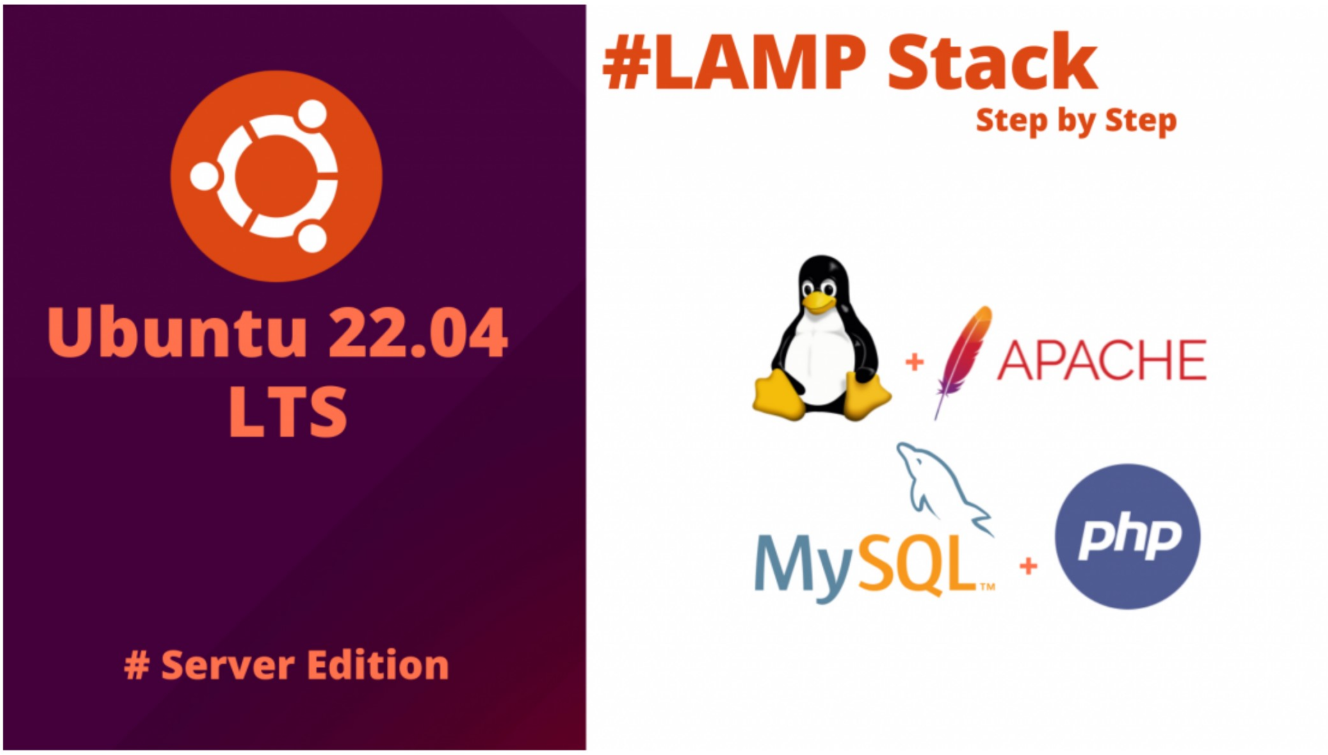
How to Install LAMP Stack (Apache, MySQL, Php) on Ubuntu 22.04

July 26, 2022 / 10,799 views / 6 min read



Overview

LAMP stands for Linux, Apache, MySQL and Php. These software packages, are required for the Php and MySQL based applications. In this guide, I am going to show you How to Install LAMP Stack (Apache, MySQL, Php) on Ubuntu 22.04. Now these days, Most of the Php based websites are CMS based and CMS like WordPress, Zoomla, Magento etc, all required the LAMP Stack to be present on the system. LAMP Stack provides the Php compatibility for the applications on the system by installing and enabling most of the required php modules with the web server. After LAMP Stack installation we will verify it by Installing the WordPress on the system.

In this tutorial, I am going to demonstrate How to Install LAMP Stack (Apache, MySQL, Php) on Ubuntu 22.04, with the following steps.

1. Operating System Upgrade
2. Install Apache Server
3. Install MySQL Server
4. install Php modules

5. Verify the Php Extensions are enabled
6. Install Phpmyadmin
7. Install WordPress on the Server

Prerequisite

Open a SSH terminal and access your Ubuntu 22.04 system with sudo (or root) privileged user.

Step1: Operating System Update

This is a best practice, before provisioning any server for production use, we must update the package repo and upgrade from it.

```
apt update
apt upgrade
```

Step2: Install Apache Server

Apache is the web server, it parses all the php files and delivers to the users browser, apache uses php processors to parse php files.

1. install apache server

```
apt install apache2
```

2. verify the version

```
apachectl -v
```

3. Check Status of apache Server

```
systemctl status apache2
```

Step3: Install MySQL Server

MySQL is the database server, it is mandatory for the web applications which are database based like CMS's, all CMS needs database to be present.

1. Install MySQL Server

```
apt install mysql-server
```

2. Check MySQL version

```
mysql -V
```

3. Check MySQL service Status

```
service mysql status
```

4. Login to MySQL

```
mysql
```

5. Provide MySQL root password

This is an example password, you should provide your own complex password.

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by
'BlueSky123!#';
```

6. Secure MySQL Installation

```
# mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root: [...] provide the MySQL root Password here

VALIDATE PASSWORD COMPONENT 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 component?

Press y|Y for Yes, any other key for No: [Hit Enter]
Using existing password for root.
Change the password for root ? ((Press y|Y for Yes, any other key for No) : [Hit Enter]

 ... skipping.
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
 - Dropping test database...
Success.
```

```
 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!
```

7. Finally, Restart the MySQL once again.

```
service mysql restart
```

Step4: Install Php with Extensions

Ubuntu 22.04 comes with the latest PHP 8.1 repository added. So we can install PHP8.1 with the following commands. These are the common modules required to provide better compatibility and performance to the php applications.

1. Install Php8.1 with modules

```
apt install php8.1 libapache2-mod-php8.1 php8.1-common php8.1-mysql php8.1-xml \
php8.1-xmlrpc php8.1-curl php8.1-gd php8.1-imagick php8.1-cli php8.1-imap \
php8.1-mbstring php8.1-opcache php8.1-apcu php8.1-soap php8.1-zip php8.1-intl \
php8.1-bcmath php8.1-bz2 php8.1-gmp zip unzip wget
```

2. Modify PHP Configurations for better performance.

```
vi /etc/php/8.1/apache2/php.ini

upload_max_filesize = 96M
post_max_size = 64M
memory_limit = 512M
```

```
max_execution_time = 600
max_input_vars = 3000
max_input_time = 1000

:x
```

3. Now, Restart the Apache Server

```
systemctl restart apache2
```

Step5: Verify the Php Extensions are Enabled

we will verify it by installing a php info page, and we will look through the page for the extensions.

1. deploy the php info page on apache's document root.

```
cd /var/www/html

vi info.php

   <?php phpinfo();

:x
```

2. Access the web page

Access the web page from the browser with the URL: *http://server_ip/info.php,* and search the page for modules and extensions like opcache, apcu etc..

Step6: install phpMyadmin for Database access

phpMyadmin is a web based MySQL client, we need this before WordPress installation we need to login to MySQL server and create the database for the WordPress

1. install phpMyadmin

```
apt install phpmyadmin
```

2. check the phpMyadmin configuration files.

```
ls -la /etc/apache2/conf-available/phpmyadmin.conf
// it should be softlink of /etc/phpmyadmin/apache.conf
```

3. Enable phpMyadmin Configuration to apache

```
a2enconf phpmyadmin.conf  // it should be already enabled
```

4. Restart Apache Server

```
systemctl restart apache2
```

5. Access phpMyadmin

Go to Bowser and hit *http://server_ip/phpmyadmin*, now login to phpMyadmin with MySQL root Credentials and create a database for the WordPress installation.

Step7: Install WordPress.

We will install WordPress to see the full compatibility of the LAMP Stack, if the LAMP Stack setups properly wordPress should be installed and accessed successfully. Please follow the steps mentioned below.

1. WordPress Installation Preparation

```
cd /var/www/htm                             // go to apaches document root
wget https://wordpress.org/latest.tar.gz   //download the latest wordpress
tar zxvf latest.tar.gz                     //it will extract on the wordpress
directory
cd wordpress                               // go to wordpress directory
mv * /var/www/html/                        // move all content to document root
chown -R www-data.www-data /var/www/html   // provide apache users permission on
```

```
contents
```

2. goto Browser and hit *http://server_ip*, wordPress Installation page should appear. Please follow the steps to install WordPress.

If you cannot cop up with the installation please follow the video guide provided in this tutorial.

Thanks !!