Install LAMP Stack on Ubuntu 18.04

LAMP stack is a collection of open source software made up of Linux operating system, Apache, MySQL database and PHP. Let's explore how to install LAMP Stack on Ubuntu 18.04.

The LAMP stack is a collection of open source software that is typically installed together on a server to run web applications or websites. LAMP is the first four letters of the four components that make up the stack: the Linux operating system, the Apache web server, the MySQL database, and finally PHP.

The following article of ***TipsMake*** *will guide you in detail to install* ***LAMP stack Ubuntu 18.04***

Step 1 - Install Apache and Update Firewall

The Apache web server is one of the most popular web servers in the world. It is well documented and has been widely used for a long time, which makes Apache a great default choice for hosting a website.

Install Apache using Ubuntu's package manager, apt:

$ sudo apt update

$ sudo apt install apache2

Since this is a sudo command, these operations are executed with root privileges. It will ask for the user password for verification.

As soon as you enter your password, apt will tell you what packages it plans to install and how much disk space they will use.

Press **Y** and press **ENTER** to continue and the installation will continue.

Adjust Firewall to allow web traffic

Next, assuming you followed the server setup guide and enabled the UFW firewall, make sure your firewall allows traffic from HTTP and HTTPS. You can check if UFW has an application profile for Apache as follows:

$ sudo ufw app list

Output

Available Apache applications:

  Apache

  Apache Full

  Apache Secure

  OpenSSH

If you look at a full Apache profile, you'll see that it allows traffic to ports 80 and 443:

$ sudo ufw app info "Apache Full"

Output

Profile: Apache Full

Title: Web Server (HTTP,HTTPS)

Description: Apache v2 is the next generation of the omnipresent Apache web

server

Ports:

   80,443/tcp

Allow HTTP and HTTPS traffic:

$ sudo ufw allow in "Apache Full"

You can do an immediate test to verify everything is set up correctly by visiting your server's public IP address in your web browser (see the notes in the next section to find out what your public IP address is if you don't have this information):

http://your_server_ip

You will see a default Ubuntu 18.04 Apache website, this page is for informational and testing purposes.

If you see this page, then the web server is installed correctly and is accessible through your firewall.

Step 2 - Install MySQL

Now that you have your web server ready to run, the next step is to install MySQL.

Continue using the apt command to download and install MySQL.

$ sudo apt install mysql-server

Note: In this case, you do not have to run sudo apt update before the command because it was already run in the previous Apache installation command. The packages installed on your machine are already up to date.

Press **Y** to continue.

Once the installation is complete run a security script that is pre-installed in MySQL to remove some of the harmful bugs and lock down access to the database. Start the interactive script by running:

$ sudo mysql_secure_installation

The command will ask if you want to configure **the PASSWORD VALIDATE PLUGIN** .

Note: If enabled, passwords that do not match the specified criteria will be rejected by MySQL with an error. This can be a problem if you use a weak password in conjunction with software that automatically configures MySQL user credentials, such as the Ubuntu packages for phpMyAdmin. It is safe to disable validation, but you should always use strong, unique passwords for database logins.

Answer Y for "yes" to continue

VALIDATE PASSWORD PLUGIN can be used to check passwords and improve security. It checks the strength of passwords and allows users to set only sufficiently secure passwords.

"Would you like to setup VALIDATE PASSWORD plugin?" - Would you like to setup VALIDATE PASSWORD plugin?

Press y | **Y** for Yes, and press any key for No:

If you answer 'Yes,' you will be asked to select a password strength level. Keep in mind that if you enter 2 to select the strongest level, you will get an error when trying to set any password that does not contain numbers, lowercase letters, uppercase letters, special characters, or common dictionary words.

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

There are levels of password strength:

LOW Length >= 8

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

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

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

Regardless of whether you choose to set up the PASSWORD VALIDATE PLUGIN or not, your server will ask you to choose and confirm a password for the MySQL root user. This is an administrative account in MySQL that has increased privileges. Think of it as the root account for the server itself (although what you are configuring now is a specific MySQL account). Make sure it is a strong, unique password and not blank.

If you have password validation enabled, you will be shown the password strength for the root password you just entered and your server will ask if you want to change that password. If you are happy with your current password, enter N for "no" at the prompt:

Using existing password for root.

Estimated strength of the password: 100

Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

For the rest, press Y and hit ENTER at each prompt. This will remove some anonymous users and test databases, disable remote root logins, and load new rules so MySQL immediately takes note of the changes you made.

If you want to use a password when connecting to MySQL as root, you will need to switch its authentication method from auth_socket to mysql_native_password. To do this, open a MySQL prompt from the terminal:

$ sudo mysql

Next, check the authentication method each of your MySQL user accounts uses with the following command:

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

Output

---------- ---------- ---------- ----------

| users | authentication_string | plugins | host |

---------- ---------- ---------- ----------

| root | | auth_socket | localhost |

| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |

| mysql.sys | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |

| debian-sys-maint | *CC744277A401A7D25BE1CA89AFF17BF607F876FF | mysql_native_password | localhost |

---------- ---------- ---------- ----------

4 rows in set (0.00 sec)

In this example, you can see that the root user actually authenticates using the auth_socket plugin. To configure the root account to authenticate with a password, run the following ALTER USER command. Be sure to change the password to a strong password of your choice:

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password' ;

Then, run FLUSH PRIVILEGES to tell the server to reload the grant tables and put your new changes into effect:

mysql> FLUSH PRIVILEGES;

Check the authentication methods used by each of your users again to confirm that root is no longer authenticating using the auth_socket plugin:

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

Output

---------- ---------- ---------- ----------

| users | authentication_string | plugins | host |

---------- ---------- ---------- ----------

| root | *3636DACC8616D997782ADD0839F92C1571D6D78F | mysql_native_password | localhost |

| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |

| mysql.sys | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |

| debian-sys-maint | *CC744277A401A7D25BE1CA89AFF17BF607F876FF | mysql_native_password | localhost |

---------- ---------- ---------- ----------

4 rows in set (0.00 sec)

You can see in this example output that the MySQL root user has authenticated using a password. Once you confirm this on your own server, you can exit the MySQL shell:

mysql> exit

At this point, your database system is now set up and you can move on to installing PHP, the final component of the LAMPstack.

Step 3 - Install PHP

Again, use apt to install PHP. Additionally, with some helper packages PHP code can run on an Apache server and communicate with your MySQL database:

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

So, you can install PHP without any problem.

In most cases, you will want to modify the way 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. When requesting a webserver, you want to prioritize PHP files over other files, so let Apache look for the index.php file first.

To do this, type the following command to open the dir.conf file in a text editor with root privileges:

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

The following command line will appear:

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

DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm

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

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

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

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

Then, restart the Apache web server for your changes to be recognized. Do this by typing the following command:

$ sudo systemctl restart apache2

You can also check the status of the apache2 service using systemctl:

$ sudo systemctl status apache2

Sample Output

? apache2.service - LSB: Apache2 web server

Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)

Drop-In: /lib/systemd/system/apache2.service.d

??apache2-systemd.conf

Active: active (running) since Tue 2018-04-23 14:28:43 EDT; 45s ago

Docs: man:systemd-sysv-generator(8)

Process: 13581 ExecStop=/etc/init.d/apache2 stop (code=exited, status=0/SUCCESS)

Process: 13605 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)

Tasks: 6 (limit: 512)

CGroup: /system.slice/apache2.service

??13623 /usr/sbin/apache2 -k start

??13626 /usr/sbin/apache2 -k start

??13627 /usr/sbin/apache2 -k start

??13628 /usr/sbin/apache2 -k start

??13629 /usr/sbin/apache2 -k start

??13630 /usr/sbin/apache2 -k start

To enhance PHP's functionality, you have the option to install a number of additional modules. To see the available options for PHP modules and libraries, pipe the results of apt search to less, a pager that lets you scroll through the output of other commands:

$ apt search php- | less

Use the arrow keys to move up and down and press Q to exit.

The result will show all the optional components you can install. It will give you a short description for each component like so:

bandwidthd-pgsql/bionic 2.0.1 cvs20090917-10ubuntu1 amd64

Tracks usage of TCP/IP and builds html files with graphs

bluefish/bionic 2.2.10-1 amd64

advanced Gtk text editor for web and software development

cacti/bionic 1.1.38 ds1-1 all

web interface for graphing of monitoring systems

ganglia-webfrontend/bionic 3.6.1-3 all

cluster monitoring toolkit - web front-end

golang-github-unknwon-cae-dev/bionic 0.0~git20160715.0.c6aac99-4 all

PHP-like Compression and Archive Extensions in Go

haserl/bionic 0.9.35-2 amd64

CGI scripting program for embedded environments

kdevelop-php-docs/bionic 5.2.1-1ubuntu2 all

transitional package for kdevelop-php

kdevelop-php-docs-l10n/bionic 5.2.1-1ubuntu2 all

transitional package for kdevelop-php-l10n

…

:

To learn more about what each module does, you can search the Internet for more information about them. Alternatively, you can see the package's long description by typing:

$ apt show package_name

Along with a wealth of other information, you will find the following:

Output

…

Description: command-line interpreter for the PHP scripting language (default)

This package provides the /usr/bin/php command interpreter, useful for

testing PHP scripts from a shell or performing general shell scripting tasks.

.

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used

open source general-purpose scripting language that is especially suited

for web development and can be embedded into HTML.

.

This package is a dependency package, which depends on Ubuntu's default

PHP version (currently 7.2).

…

If after researching, you decide to install a package, you can do so using the apt install command as you would for other software.

If you choose php-cli, you can type:

sudo apt install php-cli

If you want to install more than one module, list each module, separated by spaces, in the apt install command, like this:

$ sudo apt install package1 package2 .

At this point, your LAMP stack is installed and configured. However, before making any further changes or deploying an application, it is useful to proactively test your PHP configuration in case there are any issues that need to be addressed.

Now that you have the LAMP stack fully installed, you have a lot of options for what to do next. Essentially, you have installed a platform that allows you to run most types of websites and web software on your server.

Good luck!