**https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04**

**How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 14.04**

[Ubuntu](https://www.digitalocean.com/community/tags/ubuntu)[MySQL](https://www.digitalocean.com/community/tags/mysql)[Apache](https://www.digitalocean.com/community/tags/apache)[PHP](https://www.digitalocean.com/community/tags/php)[Getting Started](https://www.digitalocean.com/community/tags/getting-started)[LAMP Stack](https://www.digitalocean.com/community/tags/lamp-stack)

* By [Justin Ellingwood](https://www.digitalocean.com/community/users/jellingwood)

Published onApril 18, 2014 2mviews



English



**This article uses Ubuntu 14.04**This distro reached end of life (EOL) on Apr 2019.



Ubuntu 14.04



[See More](https://www.digitalocean.com/community/tutorial_collections/how-to-install-lamp)

We recommend upgrading to a more modern version. [Read upgrade instructions](https://www.digitalocean.com/community/tutorials/how-to-upgrade-to-ubuntu-18-04).

Introduction

A “LAMP” stack is a group of open source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is actually an acronym which represents the **L**inux operating system, with the **A**pache web server. The site data is stored in a **M**ySQL database, and dynamic content is processed by **P**HP.

In this guide, we’ll get a LAMP stack installed on an Ubuntu 14.04 Droplet. Ubuntu will fulfill our first requirement: a Linux operating system.

**Note:** The LAMP stack can be installed automatically on your Droplet by adding [this script](http://do.co/1FOx1QV) to its User Data when launching it. Check out [this tutorial](https://www.digitalocean.com/community/tutorials/an-introduction-to-droplet-metadata) to learn more about Droplet User Data.

Prerequisites

Before you begin with this guide, you should have a separate, non-root user account set up on your server. You can learn how to do this by completing steps 1-4 in the [initial server setup for Ubuntu 14.04](https://www.digitalocean.com/community/articles/initial-server-setup-with-ubuntu-14-04).

Step 1: Install Apache

The Apache web server is currently the most popular web server in the world, which makes it a great default choice for hosting a website.

We can install Apache easily using Ubuntu’s package manager, apt. A package manager allows us to install most software pain-free from a repository maintained by Ubuntu. You can learn more about [how to use apt](https://www.digitalocean.com/community/articles/how-to-manage-packages-in-ubuntu-and-debian-with-apt-get-apt-cache) here.

For our purposes, we can get started by typing these commands:

sudo apt-get update

sudo apt-get install apache2

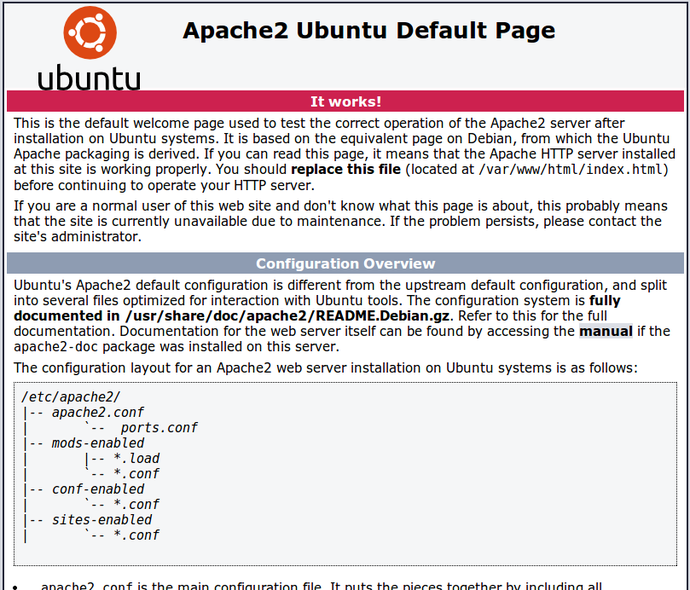
Since we are using a sudo command, these operations get executed with root privileges. It will ask you for your regular user’s password to verify your intentions.

Afterwards, your web server is installed.

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 (see the note under the next heading to find out what your public IP address is if you do not have this information already):

<pre>  
http://<span class=“highlight”>your*server*IP\_address</span>  
</pre>

You will see the default Ubuntu 14.04 Apache web page, which is there for informational and testing purposes. It should look something like this:



If you see this page, then your web server is now correctly installed.

How To Find your Server’s Public IP Address

If you do not know what your server’s public IP address is, there are a number of ways you can find it. Usually, this is the address you use to connect to your server through SSH.

From the command line, you can find this a few ways. First, you can use the iproute2 tools to get your address by typing this:

ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\/.\*$//'

This will give you one or two lines back. They are both correct addresses, but your computer may only be able to use one of them, so feel free to try each one.

An alternative method is to use an outside party to tell you how *it* sees your server. You can do this by asking a specific server what your IP address is:

curl http://icanhazip.com

Regardless of the method you use to get your IP address, you can type it into your web browser’s address bar to get to your server.

Step 2: Install MySQL

Now that we have our web server up and running, it is time to install MySQL. MySQL is a database management system. Basically, it will organize and provide access to databases where our site can store information.

Again, we can use apt to acquire and install our software. This time, we’ll also install some other “helper” packages that will assist us in getting our components to communicate with each other:

sudo apt-get install mysql-server php5-mysql

**Note**: In this case, you do not have to run sudo apt-get update prior to the command. This is because we recently ran it in the commands above to install Apache. The package index on our computer should already be up-to-date.

During the installation, your server will ask you to select and confirm a password for the MySQL “root” user. This is an administrative account in MySQL that has increased privileges. Think of it as being similar to the root account for the server itself (the one you are configuring now is a MySQL-specific account however).

When the installation is complete, we need to run some additional commands to get our MySQL environment set up securely.

First, we need to tell MySQL to create its database directory structure where it will store its information. You can do this by typing:

sudo mysql\_install\_db

Afterwards, we want to run a simple security script that will remove some dangerous defaults and lock down access to our database system a little bit. Start the interactive script by running:

sudo mysql\_secure\_installation

You will be asked to enter the password you set for the MySQL root account. Next, it will ask you if you want to change that password. If you are happy with your current password, type “n” for “no” at the prompt.

For the rest of the questions, you should simply hit the “ENTER” key through each prompt to accept the default values. This will remove some sample users and databases, disable remote root logins, and load these new rules so that MySQL immediately respects the changes we have made.

At this point, your database system is now set up and we can move on.

Step 3: Install PHP

PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display.

We can once again leverage the apt system to install our components. We’re going to include some helper packages as well:

sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt

This should install PHP without any problems. We’ll test this in a moment.

In most cases, we’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 our web server to prefer PHP files, so we’ll make Apache look for an index.php file first.

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

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

It will look like this:

<pre>  
<IfModule mod\_dir.c>  
DirectoryIndex index.html index.cgi index.pl <span class=“highlight”>index.php</span> index.xhtml index.htm </IfModule> </pre>

We want to move the PHP index file highlighted above to the first position after the DirectoryIndex specification, like this:

<pre>  
<IfModule mod\_dir.c>  
DirectoryIndex <span class=“highlight”>index.php</span> index.html index.cgi index.pl index.xhtml index.htm </IfModule> </pre>

When you are finished, save and close the file by pressing “CTRL-X”. You’ll have to confirm the save by typing “Y” and then hit “ENTER” to confirm the file save location.

After this, we need to restart the Apache web server in order for our changes to be recognized. You can do this by typing this:

sudo service apache2 restart

Install PHP Modules

To enhance the functionality of PHP, we can optionally install some additional modules.

To see the available options for PHP modules and libraries, you can type this into your system:

apt-cache search php5-

The results are all optional components that you can install. It will give you a short description for each:

php5-cgi - server-side, HTML-embedded scripting language (CGI binary)

php5-cli - command-line interpreter for the php5 scripting language

php5-common - Common files for packages built from the php5 source

php5-curl - CURL module for php5

php5-dbg - Debug symbols for PHP5

php5-dev - Files for PHP5 module development

php5-gd - GD module for php5

. . .

To get more information about what each module does, you can either search the internet, or you can look at the long description in the package by typing:

<pre>  
apt-cache show <span class=“highlight”>package\_name</span>  
</pre>

There will be a lot of output, with one field called Description-en which will have a longer explanation of the functionality that the module provides.

For example, to find out what the php5-cli module does, we could type this:

apt-cache show php5-cli

Along with a large amount of other information, you’ll find something that looks like this:

. . .

SHA256: 91cfdbda65df65c9a4a5bd3478d6e7d3e92c53efcddf3436bbe9bbe27eca409d

Description-en: command-line interpreter for the php5 scripting language

This package provides the /usr/bin/php5 command interpreter, useful for

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

.

The following extensions are built in: bcmath bz2 calendar Core ctype date

dba dom ereg exif fileinfo filter ftp gettext hash iconv libxml mbstring

mhash openssl pcntl pcre Phar posix Reflection session shmop SimpleXML soap

sockets SPL standard sysvmsg sysvsem sysvshm tokenizer wddx xml xmlreader

xmlwriter zip zlib.

.

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.

Description-md5: f8450d3b28653dcf1a4615f3b1d4e347

Homepage: http://www.php.net/

. . .

If, after researching, you decide you would like to install a package, you can do so by using the apt-get install command like we have been doing for our other software.

If we decided that php5-cli is something that we need, we could type:

sudo apt-get install php5-cli

If you want to install more than one module, you can do that by listing each one, separated by a space, following the apt-get install command, like this:

<pre>  
sudo apt-get install <span class=“highlight”>package1</span> <span class=“highlight”>package2</span> <span class=“highlight”>…</span>  
</pre>

At this point, your LAMP stack is installed and configured. We should still test out our PHP though.

Step 4: Test PHP Processing on your Web Server

In order to test that our system is configured properly for PHP, we can create a very basic PHP script.

We will call this script info.php. In order for Apache to find the file and serve it correctly, it must be saved to a very specific directory, which is called the “web root”.

In Ubuntu 14.04, this directory is located at /var/www/html/. We can create the file at that location by typing:

sudo nano /var/www/html/info.php

This will open a blank file. We want to put the following text, which is valid PHP code, inside the file:

<?php

phpinfo();

?>

When you are finished, save and close the file.

Now we can test whether our web server can correctly display content generated by a PHP script. To try this out, we just have to visit this page in our web browser. You’ll need your server’s public IP address again.

The address you want to visit will be:

<pre>  
http://<span class=“highlight”>your*server*IP\_address</span>/info.php  
</pre>

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



This page basically gives you information about your server from the perspective of PHP. It is useful for debugging and to ensure that your settings are being applied correctly.

If this was successful, then your PHP is working as expected.

You probably want to remove this file after this test because it could actually give information about your server to unauthorized users. To do this, you can type this:

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

You can always recreate this page if you need to access the information again later.

Conclusion

Now that you have a LAMP stack installed, you have many choices for what to do next. Basically, you’ve installed a platform that will allow you to install most kinds of websites and web software on your server.

Some popular options are:

* [Install Wordpress](https://www.digitalocean.com/community/articles/how-to-install-wordpress-on-ubuntu-14-04) the most popular content management system on the internet
* [Set Up PHPMyAdmin](https://www.digitalocean.com/community/articles/how-to-install-and-secure-phpmyadmin-on-ubuntu-12-04) to help manage your MySQL databases from web browser.
* [Learn more about MySQL](https://www.digitalocean.com/community/articles/a-basic-mysql-tutorial) to manage your databases.
* [Learn how to create an SSL Certificate](https://www.digitalocean.com/community/articles/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-12-04) to secure traffic to your web server.
* [Learn how to use SFTP](https://www.digitalocean.com/community/articles/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server) to transfer files to and from your server.

**Note**: We will be updating the links above to our 14.04 documentation as it is written.

<div class=“author”>By Justin Ellingwood</div>