# **Build a LAMP Web Server with WordPress**

<https://www.raspberrypi.org/learning/lamp-web-server-with-wordpress/worksheet/>

Learn to set up a LAMP (Linux, Apache, MySQL, PHP) stack on your Raspberry Pi and configure it to work as a web server. You'll download and install WordPress and set up a basic website which you can access on any device on the same network as your Pi.

## **Set up Apache Web Server**

Apache is a popular web server application you can install on the Raspberry Pi to allow it to serve web pages.

On its own, Apache can serve HTML files over HTTP, and with additional modules can serve dynamic web pages using scripting languages such as PHP.

### **Install Apache**

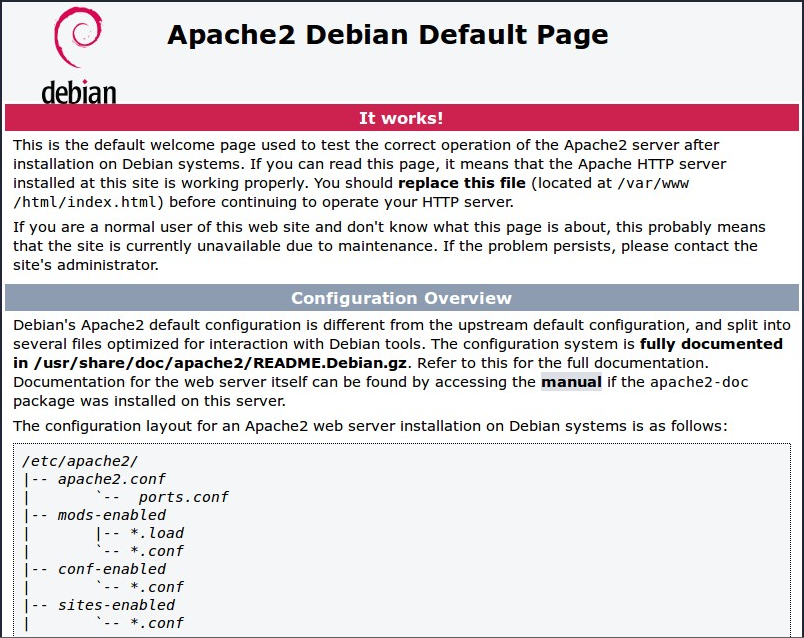
First install the apache2 package by typing the following command into the terminal:

sudo apt-get install apache2 -y

### **Test the web server**

By default, Apache puts a test HTML file in the web folder. This default web page is served when you browse to http://localhost/ on the Pi itself, or http://192.168.1.10 (whatever the Pi's IP address is) from another computer on the network. To find out the Pi's IP address, type hostname -I at the command line (or read more about finding your [IP address](http://www.raspberrypi.org/documentation/troubleshooting/hardware/networking/ip-address.md)) in our documentation.

Browse to the default web page, either on the Pi or from another computer on the network, and you should see the following:



This means you have Apache working!

### **Changing the default web page**

This default web page is just a HTML file on the filesystem. It is located at /var/www/html/index.html.

**Note: The directory was /var/www in Raspbian Wheezy but is now /var/www/html in Raspbian Jessie**

Navigate to this directory in the Terminal and have a look at what's inside:

cd /var/www/html  
ls -al

This will show you:

total 12  
drwxr-xr-x 2 root root 4096 Jan 8 01:29 .  
drwxr-xr-x 3 root root 4096 Jan 8 01:28 ..  
-rw-r--r-- 1 root root 177 Jan 8 01:29 index.html

This shows that there is one file in /var/www/html/ called index.html. The . refers to the directory itself /var/www/html and the .. refers to the parent directory /www/.

### **What the columns mean**

1. The permissions of the file or directory
2. The number of files in the directory (or 1 if it's a file).
3. The user which owns the file or directory
4. The group which owns the file or directory
5. The file size
6. The last modification date & time

As you can see, by default the html directory and index.html file are both owned by the root user, so you'll need to use sudo to edit them.

Try editing this file and refreshing the browser to see the web page change. Press Ctrl + X and hit Enter to save and exit.

## **Install PHP**

PHP is a preprocessor; it's code that runs when the server receives a request for a web page. It runs, works out what needs to be shown on the page, then sends that page to the browser. Unlike static HTML, PHP can show different content under different circumstances. Other languages are capable of this, but since WordPress is written in PHP, that's what we need to use this time. PHP is a very popular language on the web; large projects like Facebook and Wikipedia are written in PHP.

Install the PHP and Apache packages with the following command:

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

### **Test PHP**

Create the file index.php:

sudo leafpad index.php

(or use nano)

Put some PHP content in it:

<?php echo "hello world"; ?>

Now save the file. Next delete index.html because it takes precendence over index.php:

sudo rm index.html

Refresh your browser. You should see "hello world". This is not dynamic but it is still served by PHP. If you see the raw PHP above instead of "hello world", reload and restart Apache like so:

sudo service apache2 restart

Otherwise try something dynamic, for example:

<?php echo date('Y-m-d H:i:s'); ?>

Or show your PHP info:

<?php phpinfo(); ?>

## **Install MySQL**

MySQL (pronounced *My Sequel* or *My S-Q-L*) is a popular database engine. Like PHP, its overwhelming presence on web servers enhanced its popularity. This is why projects like WordPress use it, and why those projects are so popular.

Install the MySQL Server and PHP-MySQL packages by entering the following command into the terminal:

sudo apt-get install mysql-server php5-mysql -y

When installing MySQL you will be asked for a root password. You'll need to remember this to allow your website to access the database.

Now restart Apache:

sudo service apache2 restart

## **Download WordPress**

You can download WordPress from [wordpress.org](http://wordpress.org/) using the wget command. Helpfully, a copy of the latest version of WordPress is always available at [wordpress.org/latest.tar.gz](https://wordpress.org/latest.tar.gz) and [wordpress.org/latest.zip](https://wordpress.org/latest.zip), so you can grab the latest version without having to look it up on the website. At the time of writing, this is version 4.5.

Navigate to /var/www/html/, and download WordPress to this location. You'll need to empty the folder first (be sure to check you're not deleting files you need before running rm); change the ownership of this folder to the pi user too.

cd /var/www/html/  
sudo rm \*  
sudo wget http://wordpress.org/latest.tar.gz

Now extract the tarball, move the contents of the folder it extracted (wordpress) to the current directory and remove the (now empty) folder and the tarball to tidy up:

sudo tar xzf latest.tar.gz  
sudo mv wordpress/\* .  
sudo rm -rf wordpress latest.tar.gz

Running the ls or (tree -L 1) command here will show you the contents of a WordPress project:

.  
├── index.php  
├── license.txt  
├── readme.html  
├── wp-activate.php  
├── wp-admin  
├── wp-blog-header.php  
├── wp-comments-post.php  
├── wp-config-sample.php  
├── wp-content  
├── wp-cron.php  
├── wp-includes  
├── wp-links-opml.php  
├── wp-load.php  
├── wp-login.php  
├── wp-mail.php  
├── wp-settings.php  
├── wp-signup.php  
├── wp-trackback.php  
└── xmlrpc.php

This is the source of a default WordPress installation. The files you edit to customise your installation belong in the wp-content folder.

You should now change the ownership of these files to the Apache user:

sudo chown -R www-data: .

## **Set up your WordPress Database**

To get your WordPress site set up, you need a database. Run the mysql command in the terminal and provide your login credentials (e.g. username root, password password):

mysql -uroot -ppassword

Here I have provided my password (the word password) on the command line; there is no space between -p and your password.

Alternatively you can simply supply an empty -p flag and wait to be asked for a password:

mysql -uroot -p

Now you will be prompted to enter the root user password you created earlier.

Once you're connected to MySQL, you can create the database your WordPress installation will use:

mysql> create database wordpress;

Note the semi-colon ending the statement. On success you should see the following message:

Query OK, 1 row affected (0.00 sec)

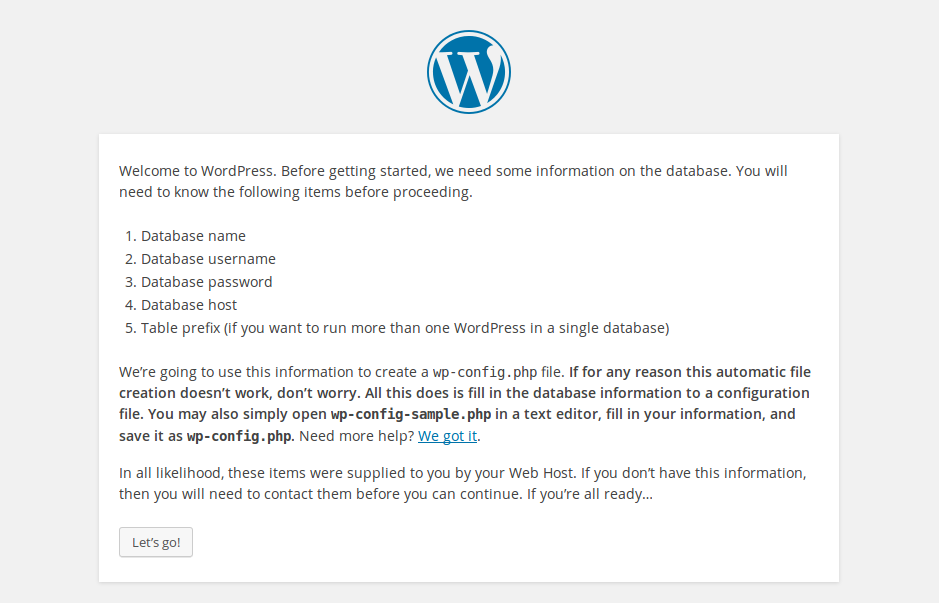
Exit out of the MySQL prompt with Ctrl + D.

## **WordPress Configuration**

You need to find out your Pi's IP address to access it in the browser, so in a terminal type the command hostname -I.

Navigate to http://YOUR-IP-ADDRESS e.g. http://192.168.1.5 in the web browser on your Pi.

You should see a WordPress weclome page.



Click the Let's go! button.

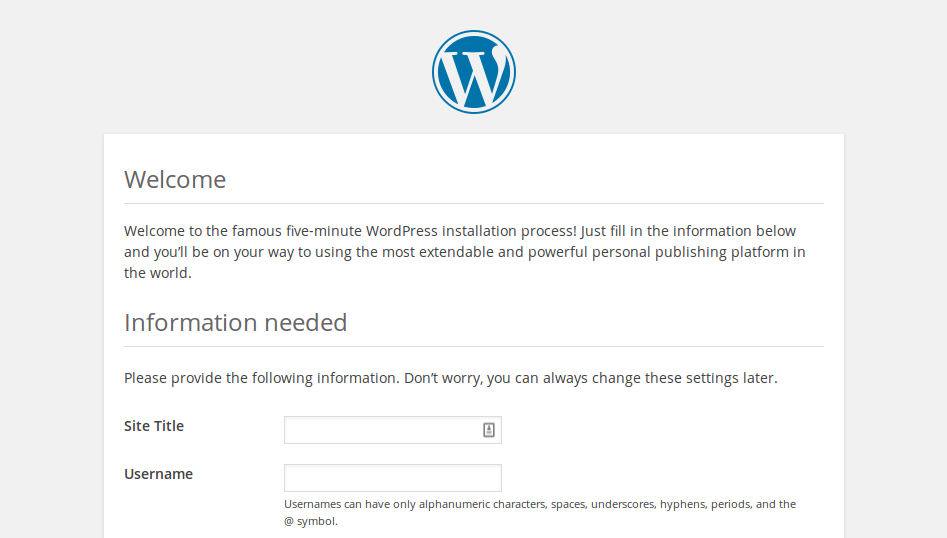
Now fill out the basic site information as follows:

Database Name: wordpress  
User Name: root  
Password: <YOUR PASSWORD>  
Database Host: localhost  
Table Prefix: wp\_

and click Submit to proceed.

Now hit the Run the install button.

Now you're getting close.



Fill out the information: give your site a title, create a username and password and enter your email address. Hit the Install WordPress button, then log in using the account you just created.

Now you're logged in and have your site set up, you can see the website by visiting your IP address in the browser on the Pi or another computer on the network. To log in again (or on another computer), go to http://YOUR-IP-ADDRESS/wp-admin.

### **Friendly permalinks**

It's recommended that you change your permalink settings to make your URLs more friendly.

To do this, log in to WordPress and go to the dashboard.

Go to Settings then Permalinks.

Select the Post name option and click Save Changes.

You'll need to enable Apache's rewrite mod:

sudo a2enmod rewrite

You'll also need to tell the virtual host serving the site to allow requests to be overwritten.

Edit the Apache configuration file for your virtual host:

sudo leafpad /etc/apache2/sites-available/000-default.conf

(or use nano)

Add the following lines after line 1:

<Directory "/var/www/html">  
 AllowOverride All  
</Directory>

ensuring it's within the <VirtualHost \*:80> like so:

<VirtualHost \*:80>  
 <Directory "/var/www/html">  
 AllowOverride All  
 </Directory>  
 ...

And then restart Apache again:

sudo service apache2 restart

### **Customisation**

WordPress is very customisable. By clicking your site name in the WordPress banner along the top of the page (when logged in), you'll be taken to the Dashboard. From here you can change the theme, add pages and posts, edit the menu, add plugins and lots more. This is just a taster for getting something interesting set up on the Raspberry Pi's web server.