



LAMP and Wordpress

Date Created

Reviewed

Version

Description

Typographic Conventions

Style	Description	Example
Text Body	Normal body text	This is normal text
Liberation Serif 10		
Key (Light Teal 4)	Keyboard shortcut	Press Ctrl + c to copy object
<i>sudo su</i>	Commands that should be run by theuser	<i>sudo apt update</i>
menu	Menu Item	File > save to access the save feature

Set up an 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. With additional modules it can serve dynamic web pages using scripting languages such as PHP.

Install Apache

- Open a terminal window by selecting **Accessories** > **Terminal** from the menu.
- Install the `apache2` package by typing the following command into the terminal and pressing Enter :

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

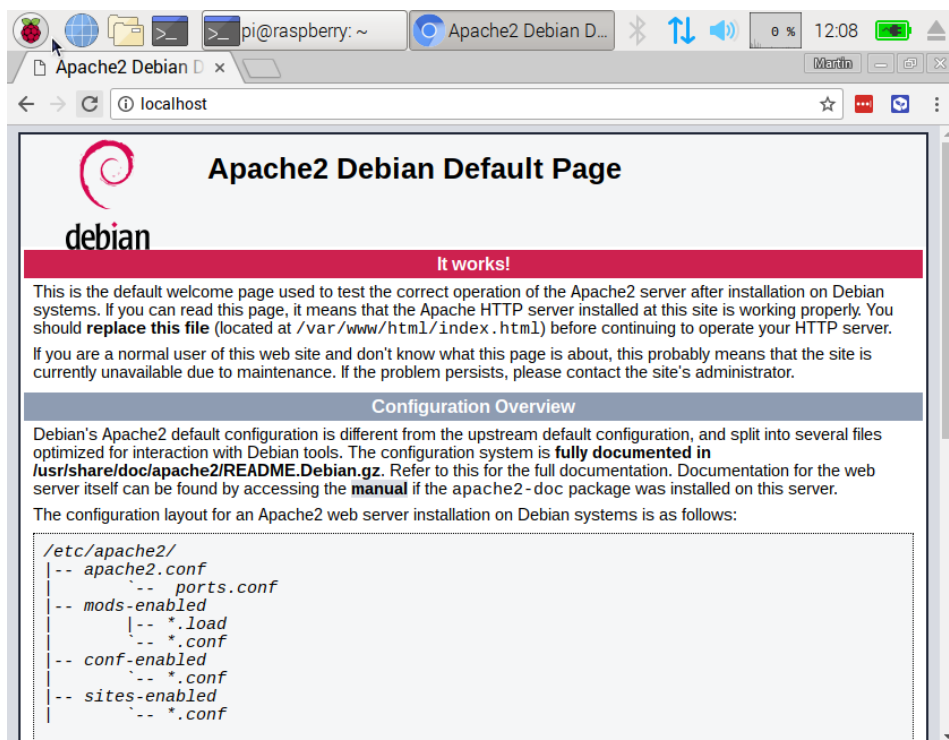
Test the web server

By default, Apache puts a test HTML file in the web folder that you will be able to view from your Pi or another computer on your network.

Open the Apache default web page on your Raspberry Pi:

- Open Chromium by selecting **Internet** > **Chromium Web Browser** from the menu.
- Enter the address `http://localhost`.

You should see this in your browser window:



This means you have Apache working!

You will also be able to open this web page from any other computer on your network using the IP address of your Raspberry Pi, e.g. `http://192.168.1.10`.

To find out your Raspberry Pi's IP address, type `hostname -I` into the terminal window. Your Raspberry Pi's [IP address](#) is a really useful and will allow you to remotely access it.

Changing the default web page

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

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

```
cd /var/www/html
```

```
ls -al
```

you will see:

```
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`. `.` refers to the directory itself `/var/www/html`, and `..` refers to the parent directory `/var/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 that owns the file or directory
4. The group that owns the file or directory
5. The size of the file or directory
6. The date and time of the last modification

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

You can edit this file using nano:

```
sudo nano index.html
```

If you make a change to the file, save it, and refresh the browser, you will see your change appear.

Install PHP

PHP is a preprocessor: it's code that runs when the server receives a request for a web page via a web browser. It works out what needs to be shown on the page, and then sends that page to the browser. Unlike static HTML, PHP can show different content under different circumstances. Other languages are also capable of doing 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: huge projects like Facebook and Wikipedia are written in PHP.

Install the PHP package with the following command:

```
sudo apt-get install php
```

Test PHP

Create the file index.php:

```
sudo nano index.php
```

Put some PHP content in it:

```
<?php echo "hello world"; ?>
```

Save the file. (Ctrl+o, <ENTER>,Ctrl+x)

Delete index.html, because it takes precedence over index.php:

sudo rm index.html

Refresh your browser. You should see “hello world”. This page is not dynamic, but it is still served by PHP.

hello world

If you see the raw PHP above instead of “hello world”, reload and restart Apache like so:

sudo service apache2 restart

Now edit index.php to include some dynamic content, for example:

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

Or show your PHP info:

<?php phpinfo(); ?>

Install MariaDB

MariaDB is a popular database engine. Like PHP, it's widely used on web servers, which is why projects like WordPress use it, and why those projects are so popular.

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

```
sudo apt-get install mariadb-server php-mysql
```

Now restart Apache:

```
sudo service apache2 restart
```

Download WordPress

You can download WordPress from wordpress.org using the `wget` command. Helpfully, a copy of the latest version of WordPress is always available at wordpress.org/latest.tar.gz, 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.

Change directory to `/var/www/html/` and delete all the files in the folder.

```
cd /var/www/html/
```

```
sudo rm *
```

Download WordPress using `wget`.

```
sudo wget http://wordpress.org/latest.tar.gz
```

Extract the WordPress tarball to get at the WordPress files.

```
sudo tar xzf latest.tar.gz
```

Move the contents of the extracted `wordpress` directory to the current directory.

```
sudo mv wordpress/* .
```

Tidy up by removing the tarball and the now empty `wordpress` directory.


```
sudo rm -rf wordpress latest.tar.gz
```

Running the `ls` or `tree -L 1` command now 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
```

3 directories, 16 files

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 all these files to the Apache user:

```
sudo chown -R www-data: .
```

Set up your WordPress Database

Set up MySQL/MariaDB

To get your WordPress site set up, you need a database. This is where MySQL and MariaDB come in!

Run the MySQL secure installation command in the terminal window.

sudo mysql_secure_installation

You will be asked Enter current password for root (enter for none): —
press Enter.

Type in Y and press Enter to Set root password?.

Type in a password at the New password: prompt, and press Enter.
Important: remember this root password, as you will need it later to set up WordPress.

Type in Y to Remove anonymous users.

Type in Y to Disallow root login remotely.

Type in Y to Remove test database and access to it.

Type in Y to Reload privilege tables now.

When complete, you will see the message All done! and Thanks for using MariaDB!.

Create the WordPress database

Run mysql in the terminal window:

sudo mysql -uroot -p

Enter the root password you created.

You will be greeted by the message Welcome to the MariaDB monitor.

Create the database for your WordPress installation at the MariaDB [(none)]> prompt using:

create database wordpress;

Note the semi-colon ending the statement.

If this has been successful, you should see this:

Query OK, 1 row affected (0.00 sec)

create database

Now grant database privileges to the root user. Note: you will need to enter your own password after IDENTIFIED BY.

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'root'@'localhost'  
IDENTIFIED BY 'YOURPASSWORD';
```

For the changes to take effect, you will need to flush the database privileges:

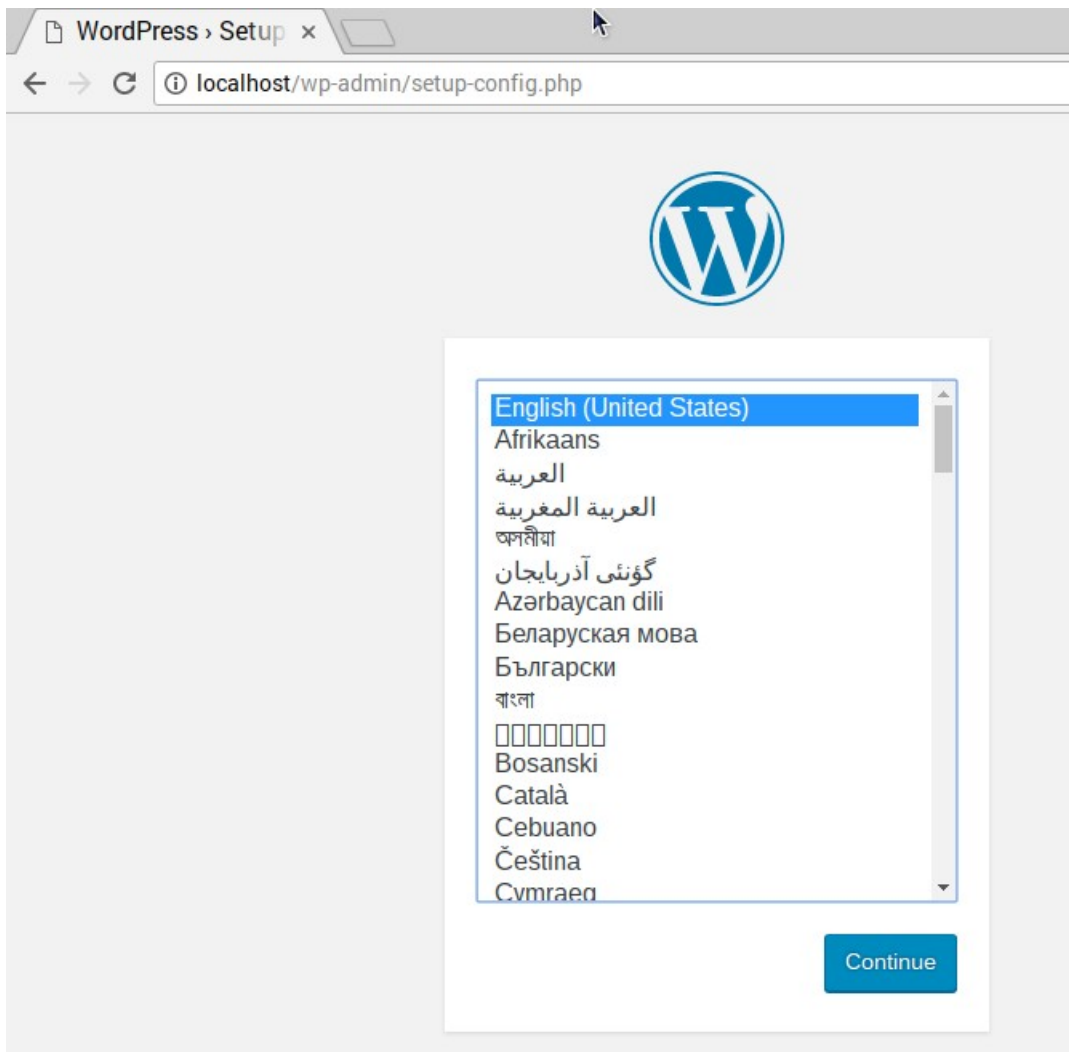
FLUSH PRIVILEGES;

Exit the MariaDB prompt with Ctrl + D.

Restart your Raspberry Pi:

```
sudo reboot
```

Open the web browser on your Pi and goto `http://localhost`, you should see a WordPress page asking to pick your language.



Select your language and click **Continue**.

You will be presented with the WordPress welcome screen.



Welcome to WordPress. Before getting started, we need some information on the database. We need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. **If for any reason this creation doesn't work, don't worry. All this does is fill in the database information to a file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? [We got it.](#)

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready...

Let's go!



Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`. Need more help? [We got it.](#)

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready...

Let's go!

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_

- Click **Submit** to proceed.
- Click the **Run the install** button.

Now you're getting close!



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Use only alphanumeric characters, spaces, underscores, hyphens, and the @ symbol.

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 <http://localhost/wp-admin>.

Log in to WordPress from another computer

To log in from another computer, open a browser and go to `http://PI-IP-ADDRESS/wp-admin`, using your Pi's IP address.

You can find your Pi's IP address using this command:

```
hostname -I
```

```
pi@raspberrypi:~ $ hostname -I
192.168.1.100 fd00::1:e7b3:46aa:a60:6771 fdf4:9ff3:e432:9500:bd4c:caf0:cf81:2e66
pi@raspberrypi:~ $ █
```

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 **Setting**, 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 mousepad /etc/apache2/sites-available/000-default.conf
```

- Add the following lines after line 1.

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

- Ensure it's within the `<VirtualHost *:80>` like so:

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

- Save the file and exit.
- Restart Apache.

```
sudo service apache2 restart
```

Customisation

WordPress is very customisable. By clicking your site name in the WordPress banner at the top of the page (when logged you're in), you'll be taken to the Dashboard. From there, 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.