Less Techie Web Server on Raspbian

Date Created: 28/04/2019

Reviewed: 28/04/2019

Version 1

Description: A less technical install of a web server on the Raspbian operating system it still is techie though!

Table of Contents

Step by step guide to install WordPress on a Raspberry Pi (or Raspbian for PC)

A guide to WordPress installed and functional on Raspberry Pi or Raspbian for PC

If you have not done a lot with Linux get a new SD card and use that. Keep your original card safe!

WordPress will run on a Raspberry Pi provided you first install LAMP server on it, i.e.: Linux + Apache + MySQL + PHP

Assuming your Raspberry Pi is blank, or that you want to reinstall the operating system just for WordPress.

Note: Don’t forget you could put a different sd card in your Pi so you don’t break your working one. I use a 16Gb card for this. It’s more than ample for the task

Step 1 – Install Raspbian

Download Raspbian

First of all, you will have to download the latest version of Raspbian from the official website

If you only want to install WordPress, the lite version may be enough, if you use the lite version it comes without the GUI so all commands are done either by SSH on a remote computer or using the command line on the Pi. It is not too complicated

Once you have the image you will need to move it to an SD card. Obviously the card will have to be bigger than the image but anything 4Gb or bigger will do fine.

Flash an SD Card with Raspbian

I use Etcher for this.

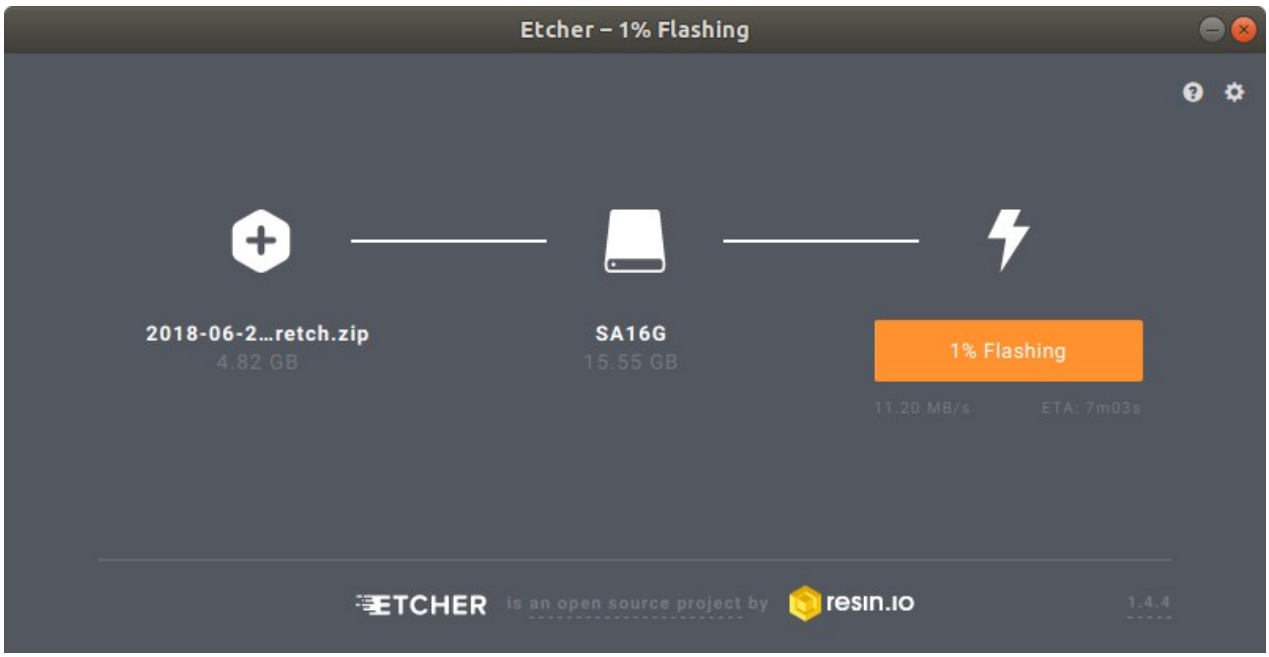
If you have not installed it yet, get it from the official website etcher.io It’s a tool that allows you to flash an SD card very easily on Linux, Mac, or Windows

Start Etcher

Select the location of the Raspbian image usually your downloads folder

Choose your SD card (be careful you don’t choose your hard drive)

Click on **Flash**



Once the SD card is ready, eject it and **insert in your Raspberry Pi**

First Boot

The installation is automatic, you just have to start your Raspberry Pi, and Raspbian will launch. If you use a very big SD card (I have used 128Gb) then things might be slow while the system sorts out the additional free space.

GUI

If you chose the Desktop version, a welcome menu will open:

Choose your language preferences

Change the default password (If you leave this blank the password will remain as raspberry)

Connect to the wifi if necessary

Accept system updates

Reboot the Raspberry Pi

Next, if you want to use SSH it has to be enabled. SSH allows you to access your Pi remotely. SSH is a secure remote connection protocol, which allows you to launch commands from another computer on the network

Note: The use of SSH is often the case with web servers as they are often used without keyboard mouse or monitor a state referred to as headless

By default, the service is not started on Raspberry Pi

Unless it’s not possible for you, I strongly recommend using SSH for the rest of this guide

It will be much simpler to copy commands, test, etc … from your usual computer

GUI

To enable SSH through the GUI, go to the applications menu, then Preferences > Raspberry Pi Configuration > Interfaces
Check Enabled for the SSH line

(see appendix 1)

Terminal

You can also launch the terminal and type this command:

```
sudo service ssh start
```

To Autorun SSH at each boot

The SSH service is not started automatically when the Raspberry Pi is started
If necessary you can start it with each reboot, by typing this command:

```
sudo crontab -e
```

If you are new to And adding this line:

```
@reboot /usr/sbin/service ssh start
```

Connecting

You can now connect to your Raspberry Pi with SSH

I recommend you can access you Pi with PuTTY which can be installed with

```
sudo apt-get install putty
```

Once PuTTY is installed you can access your Pi using SSH. You will need to know the IP address of your Pi

(See appendix 2 for more details)

Step 2 – Install the web server (Apache, PHP)

Introduction

WordPress is a web application, written in PHP
We need a web server to make it available so do that we will **set up all the components of a LAMP server**:
– **L:** Linux (Raspbian)
– **A:** Apache
– **M:** Mysql (MariaDB)
– **P:** PHP

We already have the “L” in place with our Raspbian installation, let’s go to Apache

Install Apache

Apache is the most popular web server on the internet

Its role is to provide visitors with HTML files that will then be interpreted by browsers

Install Apache:

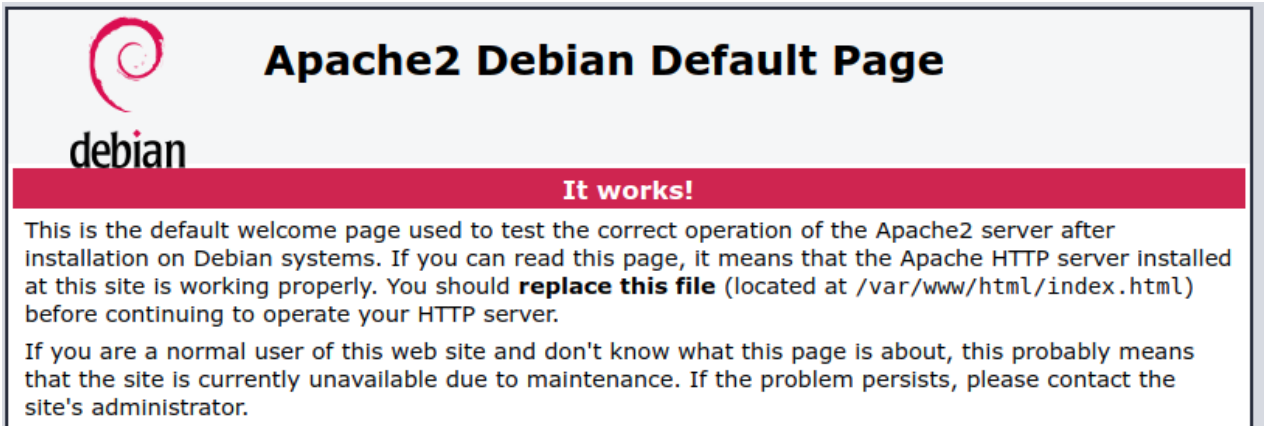
sudo apt-get install apache2

Once completed you have apache installed.

You can now navigate to the default web page by typing the IP address of the Raspberry Pi into a browser (http: //nnn.nnn.nnn.nnn)

Note: If you are actually working from the Pi you could replace nnn.nnn.nnn.nnn with 127.0.0.1

You should see something like this :



If you see this you have installed Apache properly.

Install PHP

PHP is a programming language, which will allow you to create dynamic web pages (e.g. display to add your name dynamically in the page)

We need to **install PHP** and allow Apache to use it

- sudo apt-get install php

Test PHP with Hello world

To make sure that PHP is active, we will do the following test:

Go to the folder /var/www/html

cd /var/www/html

Create and edit the test.php file

sudo nano test.php

Type the following PHP code

```
<?php
        echo "Hello World!";
?>
```

Save and close

Then go to http://nnn.nnn.nnn.nnn/test.php and watch

It should be displayed only “Hello World!”

Step 3 – Install database server (MariaDB)

Introduction to MariaDB

We are now able to create a website in HTML and PHP and you could try doing this with projects from codeclub.org

To continue with WordPress you a little more; it requires a database to store all posts, pages, and configurations

For that, you will install MariaDB, (a free fork of MySQL which exists since the acquisition of MySQL by Oracle. It is the preferred database management solution)

Installation of MariaDB

To install it you use the apt-get command

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

It takes a little longer than the previous installs

Configuration of MariaDB

Access to the database is protected by a password, which may be different from the one of the user

Note: Make a careful not of all the usernames and passwords you use and keep them safe

By default only the root user has access without a password from her/his account, so we will connect to it and **create a new account for WordPress**

Connect to the MySQL CLI

```
sudo mysql --user=root
```

Create a new user

```
CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'password';
```

Note: Keep a record of the username and password.

Create a new database

```
CREATE DATABASE wordpress;
```

Give WordPress user all privileges on the new database

```
GRANT ALL ON wordpress.* TO 'wordpress'@'localhost';
```

Quit the MySQL CLI

```
quit
```

Note: Keep a record of these details

PHP MySQL package

To allow PHP to connect to MySQL, you require one more package php-mysql

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

You will have to restart Apache to apply the changes

```
sudo service apache2 restart
```

Test

Again, let's do a **quick test** to make sure the connection is functional:

Note: Don't forget the ; at the end of each SQL command!

Connect to MySQL with the user wordpress and the password password

```
mysql -u wordpress -p
```

Enter the user password

Check if the user can view the new database

```
SHOW DATABASES;
```

Quit

```
quit
```

If you see at least the wordpress database, then it's okay, you can continue and move to the WordPress installation itself

Step 4 – Install WordPress

Download WordPress from WordPress.org

WordPress offers its software under two versions

A version hosted on their servers, WordPress.com

A version to download, WordPress.org

Because you are going to install WordPress on the Pi you have to use the second option

Download WordPress from the official website
Copy the link (right click on it) and download it to the Raspberry Pi via SSH:

- sudo wget https://wordpress.org/latest.zip -O /var/www/html/wordpress.zip

Uncompress the downloaded file

Move to the web folder and uncompress the file:

- cd /var/www/html

Check the file is there

ls

If it downloaded correctly you will see wordpress.zip in the var/www/html folder
Unzip the file

sudo unzip wordpress.zip

Permissions

To avoid permission problems with WordPress, and to remove the need for sudo later, we can modify the rights on the WordPress files.

- sudo chmod 755 wordpress -R
- sudo chown www-data wordpress -R

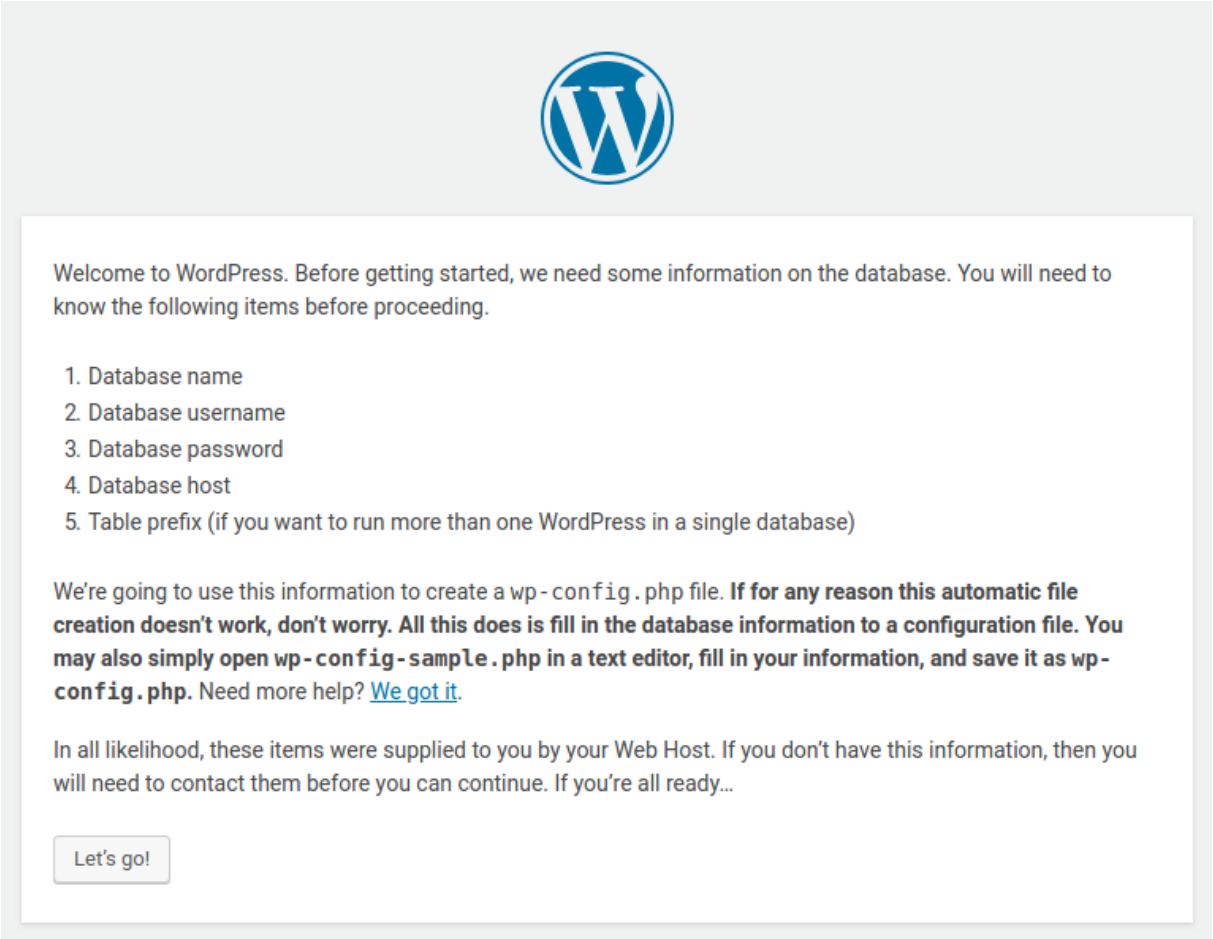
This will give full rights to Apache, and read/execution to others

Configuration

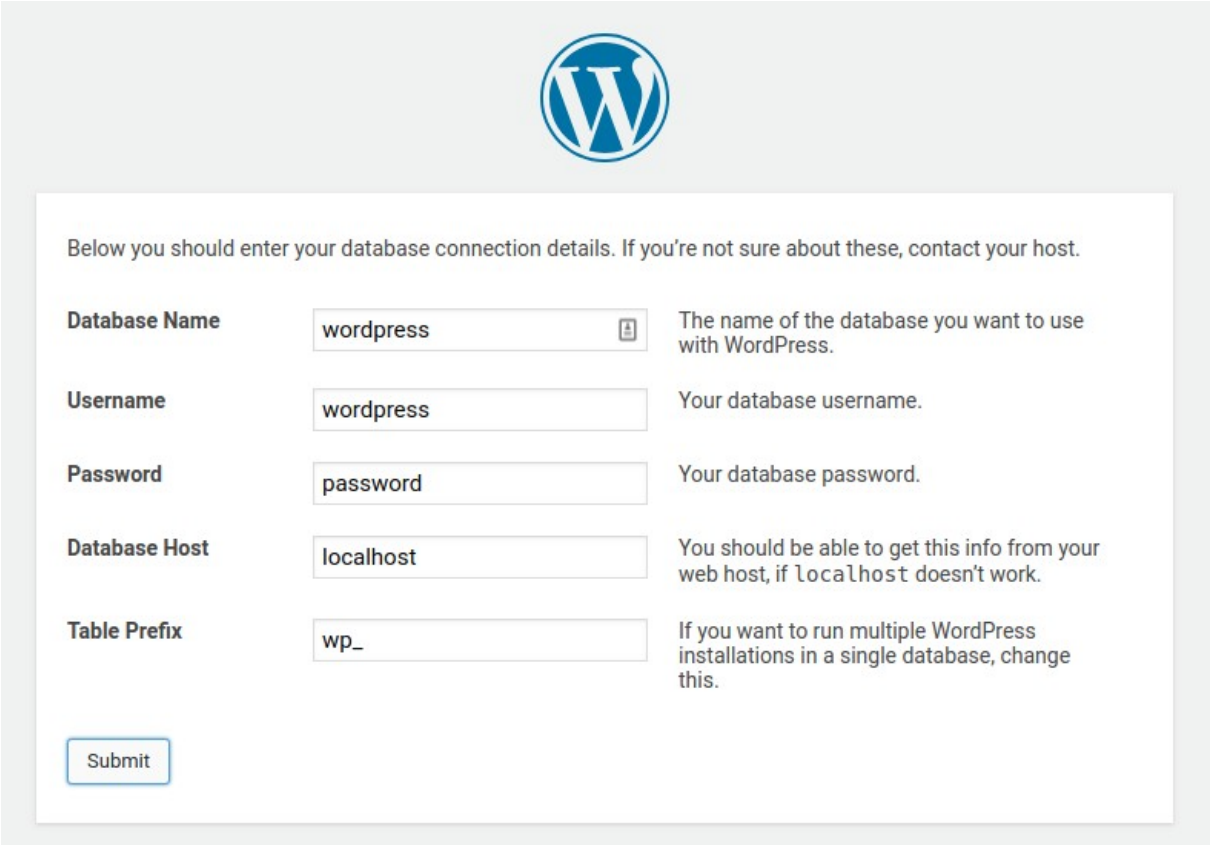
The code of WordPress is now in place; it remains only to configure it, i.e.:

- Configure the connection to the database
- Create a login for the administration of WordPress

Access the installation wizard by pointing your browser at http://X.X.X.X/wordpress

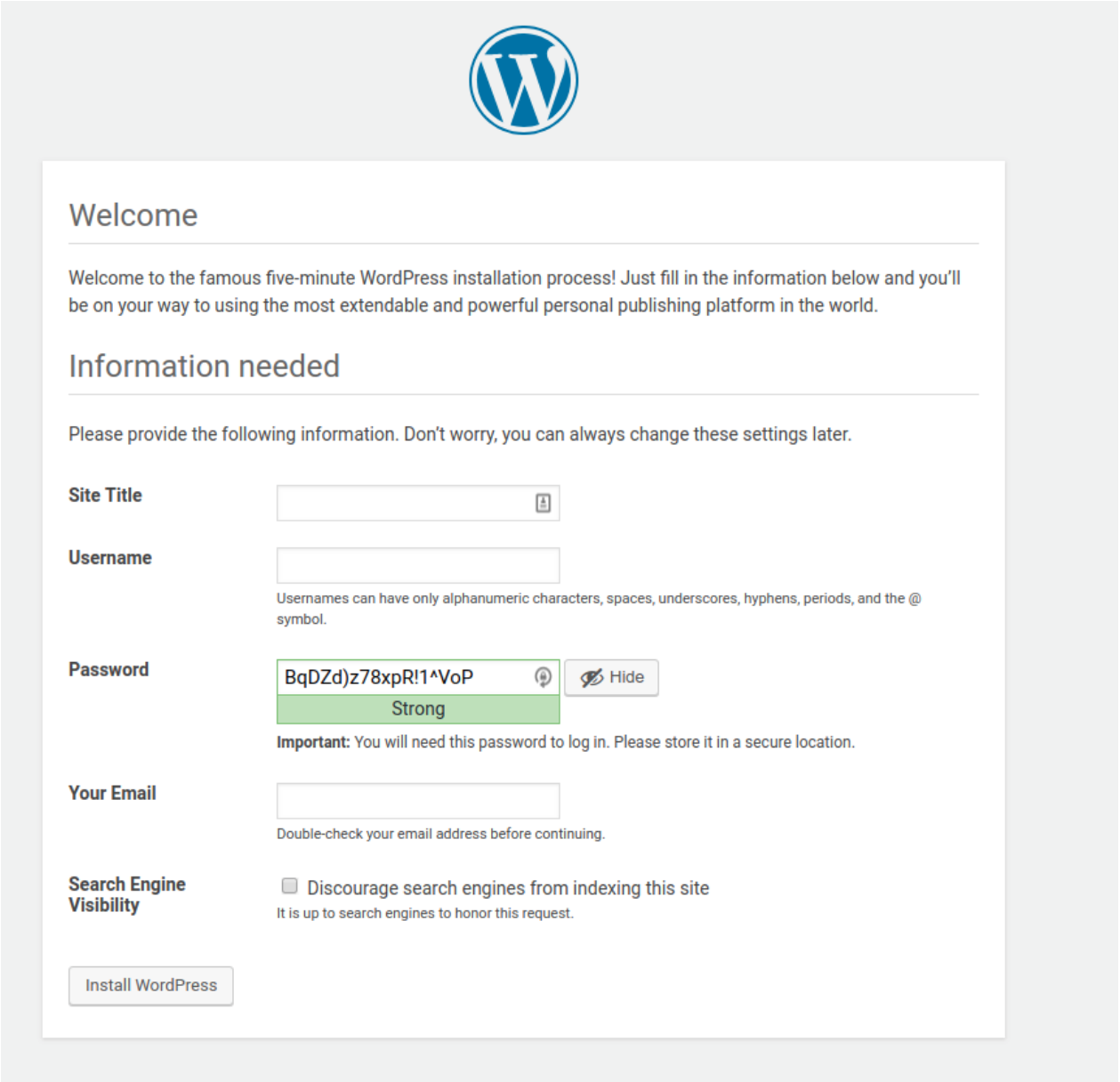


Click Let's Go Button
On the next screen, fill the form with the MySQL user created before
You should have something like this



Validate, and the wizard will ask you to run the installation
Click the button and wait

On the next screen you have to choose the site name, and create the administrator user



Fill in the fields with what you want, then click Install WordPress. Make sure you haved kept a copy somewhere safe.

There you are!

The configuration is complete; you can go back to the address http://nnn.nnn.nnn.nnn/wordpress to see your WordPress website up and running

The wizard offers you to go directly to the administration page!

I hope you made a note of your username and password

A few tips

WordPress introduction

Because of the popularity of WordPress there are many tutorials on using it.

Admin and Front

WordPress is composed of two parts:

- **Administration:** accessible by adding / wp-admin to the URL, it allows you to configure your site and add content
- **Front:** this is the part visible to all visitors

When you are logged in, the top tool bar allows you to switch from one to the other easily

Appearance

WordPress comes with a basic design, but it is possible to customize your site as you wish
To get started, go to Appearance> Themes
Here you can **add free themes** from the list, I like 2016. Some themes cost to install them. Most are free.

In the Appearance menu, you can also manage the menus of your site, and the widgets
A widget is a block that can be integrated into the sidebar for example to display a search engine or an image

Plugins

A bit like Raspberry Pi, WordPress has a huge foundation of people always adding to and enhancing its core.It is possible to install plugins, to add additional features to your website or your administration functions

Go to Plugins > Add new to see a list of all plugins available. Again some are free and some are not.

Pages and Posts

Once your site is personalized, it’s time to **add content**

For this you can create two types of content:

- **Pages:** These are static pages, which contain for example your homepage or a contact form
- **Posts:** that’s all the rest of the content, they are grouped into categories to make it easier for them to search later

You will be able to add your pages and categories to your menus, and the posts will appear automatically

Back-end management

Let’s go back to our services that allow you to run WordPress: Apache, PHP, and MySQL

You need to know that there are commands to start or stop them
This may be useful in case of a crash, or if you want to stop them to cut access to the site
Here are the commands:

- sudo service apache2 start | stop | restart | reload
- sudo service mysql start | stop | restart | reload

As mentioned above PHP is a module of Apache, so there is no particular command to launch it

If Apache is running, the PHP pages will be displayed correctly

Services configuration files

if you ever want to make changes here is where to find the configuration files

Apache : /etc/apache2
PHP : /etc/php
MySQL : /etc/mysql

In the case of a WordPress installation, you normally do not need to touch it

PHPMyAdmin

PHPMyAdmin is a handy tool that you can use on a LAMP installation
This is a web interface that will allow you to access your MySQL databases in a more intuitive way
You will be able to view and modify the data, create users, manage rights and supervise the MySQL server.

To install it, use the following command

- sudo apt-get install phpmyadmin

Choose apache2 as your web server when asked
You can skip database configuration

Once the installation is complete, the interface is available at http://nnn.nnn.nnn.nnn/phpmyadmin
Log in with your wordpress account to see the WordPress database

Share the website on the Internet

If you want to share this newly created website on the internet you must have access to the internet.

You simply have to redirect a port from your router on the port 80 of Raspberry Pi = to be able to connect to the site by using the public IP address of your Internet connection
Public IP:PORT => Raspberry Pi IP: 80

So you can access the website with http://<Public IP address>:PORT/wordpress

<Public IP address> is your public IP, and PORT the port you choose most people choose port 80 or 8080

If you do not have a fixed IP address, you can use dynamic DNS services that allow you to use a web address that will be constantly updated with your new IP address if it changes.

Finally

Well done! you learned how to install a LAMP server on Raspberry Pi and to install WordPress

You can access your website on your local network by typing

    http://<IP Address>/wordpress/index.php

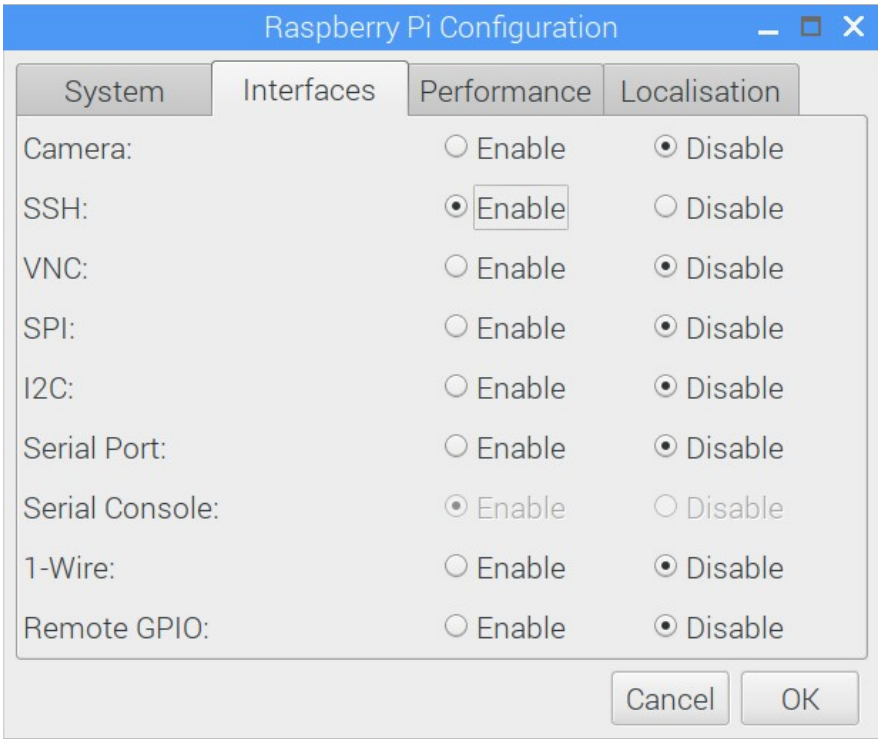
Good Luck!

Appendices

Appendix 1 How to enable SSH on a Raspberry Pi?

From the desktop

For security reasons, the SSH service is not enabled by default on Raspberry Pi
To must activate it yourself go to the menu Start> Preferences> Raspberry Pi Configuration
Go to the “Interfaces” tab and enable SSH



From a terminal

The same thing from a terminal, you have to start the SSH service manually
To do this, type the following command:

sudo service ssh start

You can also check the service status by typing:

sudo service ssh status

Enabling SSH at start-up

The SSH service is not started automatically when the Raspberry Pi is started
If necessary you can start it with each reboot, by typing this command:

- sudo crontab -e

And adding this line:

- @reboot /usr/sbin/service ssh start

Appendix 2 Installing PuTTY

Putty is not always installed on Linux so it may be necessary to install it, This is the case for Raspbian which is a distribution based on Debian:

I always get my computer up to date before installing any new software.

sudo apt-get update

sudo apt-get upgrade

Then I install the new software, in this case SSH

sudo apt-get install ssh

The command to connect to your Raspberry Pi from a remote computer will be something like this:

ssh pi@<IP address>

where <IP address> is the IP address of your Pi

Replace <IP address> by the IP address of your Raspberry Pi

Note: You can find the address of your Pi by using ifconfig or hovering the mouse pointer over the wireless icon on the right of the screen

As a reminder, **the default SSH login and password are:**
– Login: pi
– Password: raspberry