How to install wordpress on Raspberry Pi 3+ – beginner guide

Read the whole document before you start, this will be less of a headache

1. Insert a micro SD into your computer (or use an external card reader)

2. Download Win32 Disk Imager on your computer - https://sourceforge.net/projects/win32diskimager/

   (or balena echer https://www.balena.io/etcher/)

3. Download the Raspberry Pi OS from – https://www.raspberrypi.org/downloads/raspberry-pi-os/

   I recommend the full version, easier GUI.

4. Write the file onto the SD micro card

   Now you have two options, either do the configuration directly on the raspberry, for this you will need an external mouse, keyboard and monitor or you can do it remotely through ssh from your own computer.
   If you want to do the latter keep reading, if not, skip to step number 5.

   In order to do it completely remotely you will need to have access to the local internet router, in order to obtain the ip-address of the raspberry.

   a) Just simply create a file name “ssh.txt” or “ssh” in the sd card when it is in your computer.

   b) Insert the SD card into your raspberry, plug in an ethernet cable and power it up.

   c) Find the IP address of your raspberry device through accessing your router.
   (this can be done through opening the command prompt and typing “ipconfig” on windows).

   d) download putty -https://www.putty.org/

   e) Click the option of connecting through SSH and you should connect to the raspberry.
   Default username: *pi* default password: *raspberry*

   Hopefully this worked, and you can continue on step 7.

5. Plug in your monitor, keyboard, mouse and SD card – then power up the raspberry. This will take some minutes.

6. Connect to the internet, Wi-Fi or ethernet, only affects the speed for this installation.

7. Input the following commands in the terminal:

   ```
   >sudo apt-get update
   >sudo apt-get upgrade
   >reboot
   ```

   This will update everything and reboot the raspberry.

After this initial configuration, now begins the hosting part, we are going to install a web server, PHP and a database.

8. **Web server – NGINX**
   works well with the resources available to the raspberry

   Input the following commands in the terminal:

   ```
   >sudo apt-get install nginx
   >sudo nginx -t
   >sudo service nginx start
   >sudo service nginx status
   ```

To check if everything is working, open the web browser on you raspberry and type in "localhost" or the raspberry pi ip address, you should see the nginx page.

9. **PHP - a general-purpose scripting language that is especially suited to web development.**

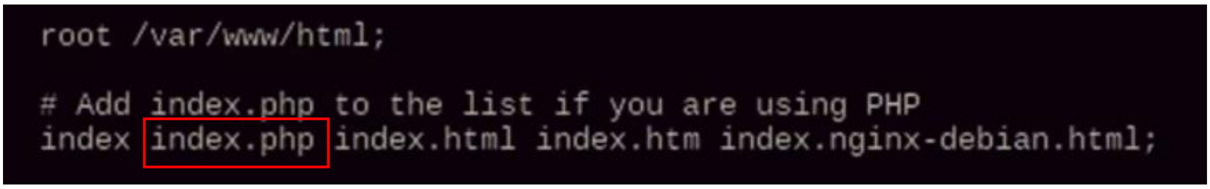
Input the following commands in the terminal (7.3 was the newest working version at this time, it might need an update):

```
>sudo apt-get install php7.3-fpm php7.3-mbstring php7.3-mysql php7.3-gd php7.3-curl php7.3-zip php7.3-xml -y
```

```
>sudo nano /etc/nginx/sites-enabled/default
```

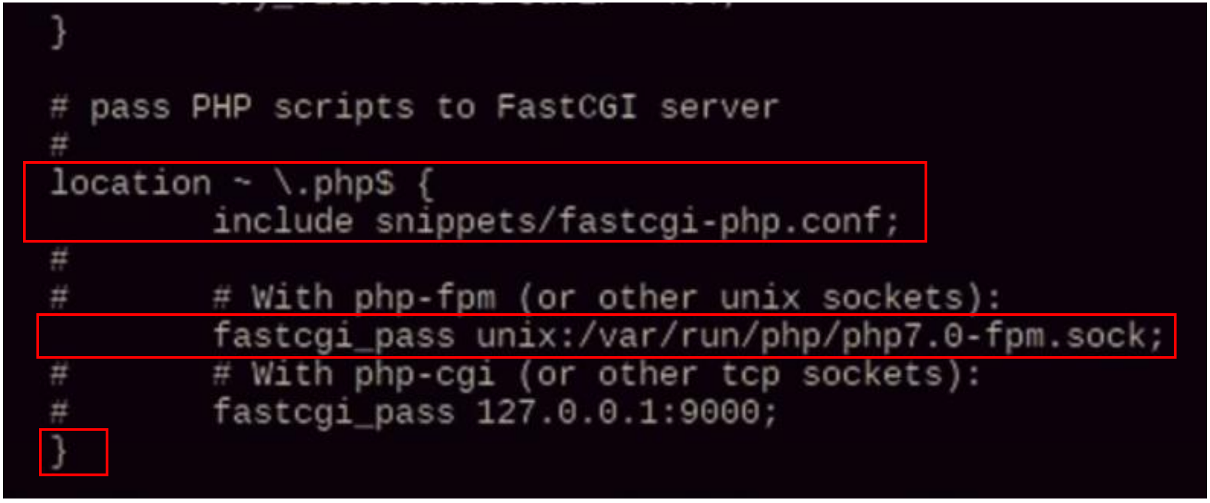
This will pull-up a new window:

Add "index.php"



```
root /var/www/html;

# Add index.php to the list if you are using PHP
index index.php index.html index.htm index.nginx-debian.html;
```

Enable/edit the following:



```
    }

    # pass PHP scripts to FastCGI server
    #
    location ~ \.php$ {
            include snippets/fastcgi-php.conf;
    #
    #       # With php-fpm (or other unix sockets):
            fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    #       # With php-cgi (or other tcp sockets):
    #       fastcgi_pass 127.0.0.1:9000;
    }
```

Save & exit

Input the following command in the terminal to make nginx know to look for php:

>sudo systemctl reload nginx

>sudo nano /var/www/html/index.php

This will pull up a new window, input:

<?php phpinfo(); ?>

save & exit

To check if everything is working, open the web browser on you raspberry and type in "localhost", you should see the php page.

10. **Database – Mariadb**

It is compatible and have a lot of features

Input the following command in the terminal:

>sudo apt-get install mariadb-server mariadb-client

>sudo mariadb

NB new commandline, (different syntax)

Input the following, in order to create a user with access to the database "wordpress":

>DROP USER 'root'@'localhost';

>CREATE USER 'root'@'localhost' IDENTIFIED BY 'toor';

>GRANT ALL PRIVILEGES ON *.* TO ‘root’@’localhost’;

>CREATE DATABASE wordpress;

>exit

11. **Wordpress**

Input the following command in the terminal (everytime a new consol pops up, see the images on the next page):

>cd /var/www

>sudo wget https://wordpress.org/latest.tar.gz

>sudo tar xvfz latest.tar.gz

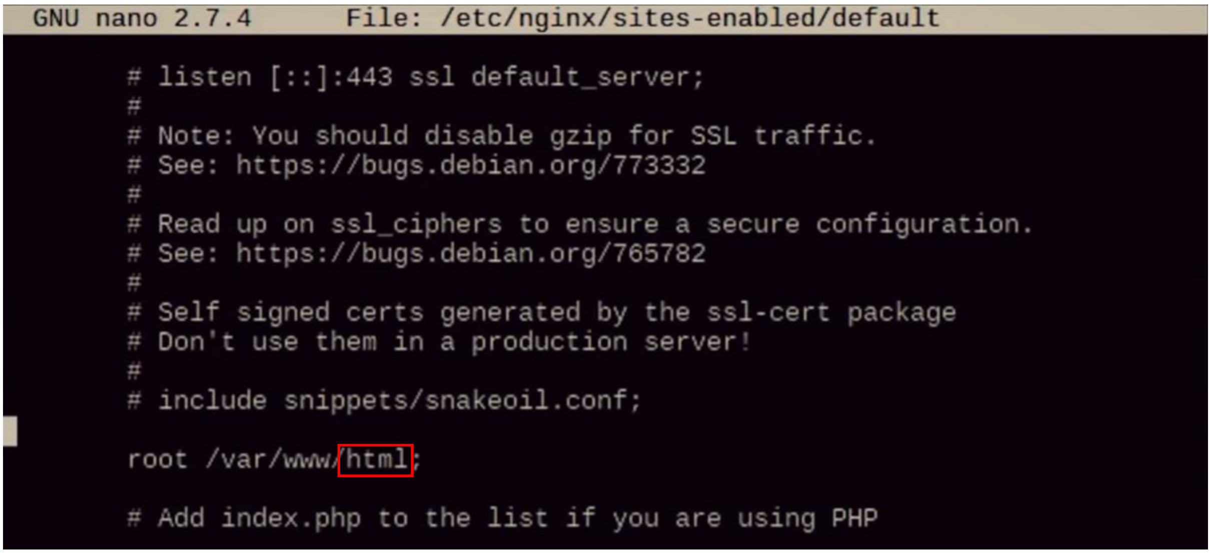
>sudo rm latest.tar.gz

>sudo mv wordpress titleofyourpage

>sudo chown -R www-data:www-data /var/www/titleofyourpage

>sudo nano /etc/nginx/sites-enabled/default

Change html to titleofyourpage

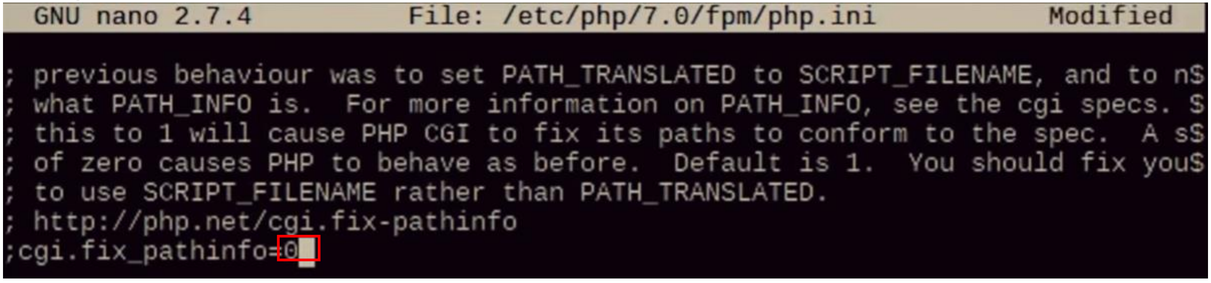


```
  GNU nano 2.7.4        File: /etc/nginx/sites-enabled/default

        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
```

save & exit

>sudo nano /etc/php/7.3/fpm/php.ini

Change from 1 to 0



```
  GNU nano 2.7.4              File: /etc/php/7.0/fpm/php.ini              Modified

; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to n$
; what PATH_INFO is.  For more information on PATH_INFO, see the cgi specs. $
; this to 1 will cause PHP CGI to fix its paths to conform to the spec.  A s$
; of zero causes PHP to behave as before.  Default is 1.  You should fix you$
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
;cgi.fix_pathinfo=0
```

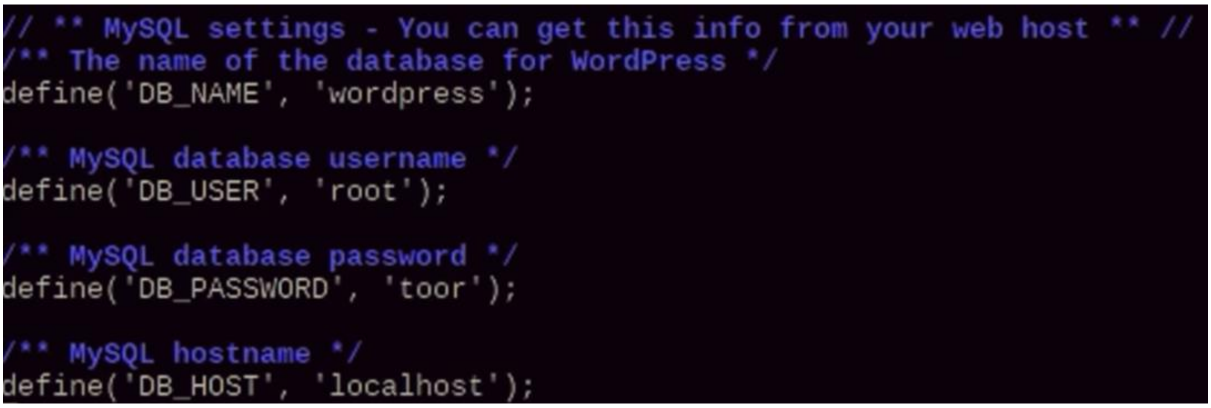
>sudo systemctl restart php7.3-fpm.service

>sudo systemctl restart nginx.service

>sudo mv /var/www/titleofyourpage/wp-config-sample.php /var/www/titleofyourpage/wp-config.php

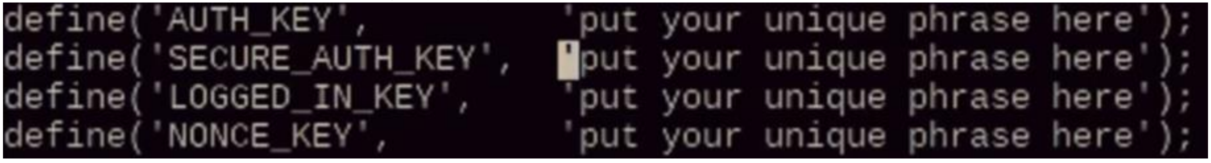
>sudo nano titleofyourpage/wp-config.php

Edit the content to this:



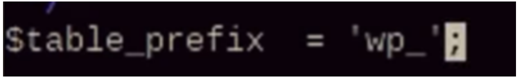
```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'toor');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

Insert a secure key where it says ‘put your unique phrase here’



```
define('AUTH_KEY',         'put your unique phrase here');
define('SECURE_AUTH_KEY',  'put your unique phrase here');
define('LOGGED_IN_KEY',    'put your unique phrase here');
define('NONCE_KEY',        'put your unique phrase here');
```

Also here (after **wp_)**



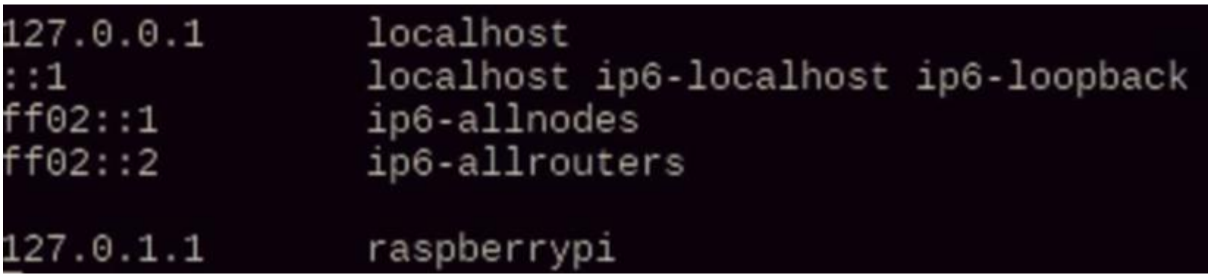
```
$table_prefix  = 'wp_';
```

Refresh the browser, and you are ready to configure the wordpress site.

In order to have your browser redirect you to the name titleofyourpage you have to input the following in the terminal:

>sudo nano /etc/hosts

And change raspberrypi to titleofyourpage



```
127.0.0.1       localhost
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

127.0.1.1       raspberrypi
```

NB! Also change the url in the wordpress settings!

If everything went correctly, you will now you have a wordpress site on your local network.

Next: install an owncloud/nextcloud instance on the RPI
- “link to locally available owncloud”
- see: owncloud = SchoolConnectivity-KE = RPI content