Installation guide Rasberry PI 2 – Server Hosting

INFO

Spring Semester – SAE 2.03

Year 2022/23

LESCURE David

ROSTAING Mathieu

ROCHE Rémi

1. Preface

This is an installation guide that aim the creation of a secured WordPress website using a raspberry. This installation guide occurred in the context of an improvement of the development system. This system will run a secured website accessible on a local network. Installation guide written by LESCURE David, ROSTAING Mathieu & ROCHE Rémi.

TRADEMARKS

2. Table of content

3. Introduction : Using this guide

3.1. Purpose

Welcome to this guide,

This guide is made for you, is made for make you first raspberry PI. In this guide you can found the different step to install and configure your raspberry pi.

This guide aims for the of a WordPress web server using a limited hardware as a raspberry.

3.2. Scope

This is an installation guide for a WordPress web server on a Raspberry Pi 2 for a local network. It is supposed you have a functional network. You will need a Raspberry Pi 2 (or later/more powerful), an 8Gb micro-SD card (optional: micro-SD to SD adaptor, micro-SD/SD to USB adaptor), a USB Type A to micro-USB Type B cable, an HDMI cable (or a HDMI adaptor) and an RJ45 cable (ethernet cable).

A screen, a keyboard and a conform power source (USB Type-A plug on a computer for example).

3.3. System Organisation

The guide is organised in nine parts.

You will:

- Burn an image of the OS on the micro-SD card;
- Set up the Raspberry and basic configuration;
- Set up SSH;
- Create a sudoer;
- Set up the server;
- Set up securities.

The sections are described as much as possible and illustrated. You will have commands to copy.

Then there will be a testing section as well as a troubleshooting and a glossary.

3.4. Technical Support

If you encounter issues related to DietPi OS you could find help on their forum.

https://dietpi.com/forum/

You can also check out the Raspberry Pi Documentation.

https://www.raspberrypi.com/documentation/

If you encounter issues with WordPress, you can contact their support team via their website.

https://wordpress.com/support/contact/

Technical information about the Debian environment and functionalities can be found at this address:

https://www.debian.org/support

The support includes:

Documentation Help in real time with IRC Forum with other users Help with known problem System of bug follow If your problem persists, please contact:

debian-boot@lists.debian.org

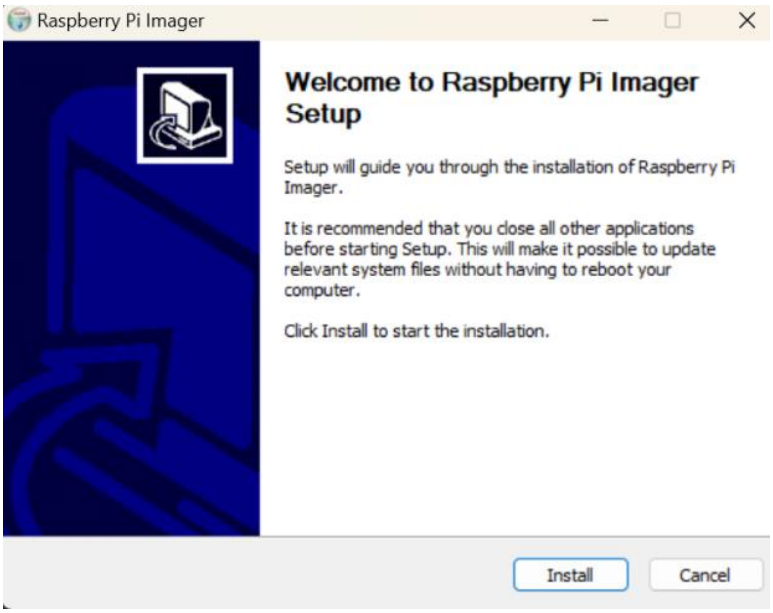
4. Burning the OS image

4.1. Installing the imager

Search “raspberrypi.com” in your web browser.

Click on “Software”.

Run the downloaded file and then follow the instructions.



4.2. Downloading the OS image

In your browser, type “dietpi.com”.

Then go to the download section and select “Raspberry Pi”.

You must now choose the “Raspberry Pi 2 PCB v1.1” raspberry and download the image.

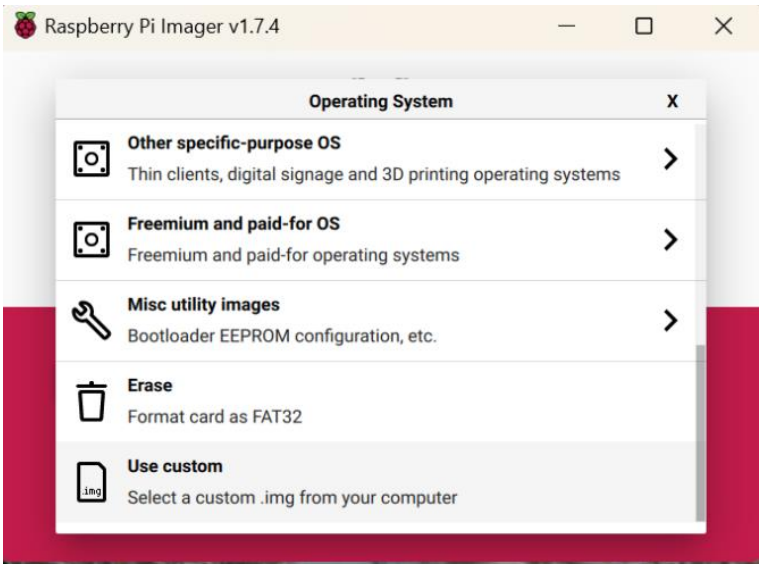
Unzip the file (you might need to install an unzipper like 7zip).

DietPi_RPi-ARMv7-Bullseye.7z

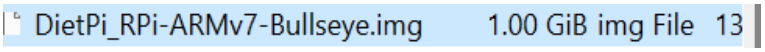
4.3. Burning the OS image onto the SD card

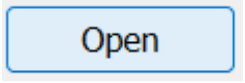
Launch the Raspberry Pi Imager previously installed.

You must choose the OS image file clicking on “CHOOSE OS”.



Select “Use custom” and browse your file.



Open it ➔ 

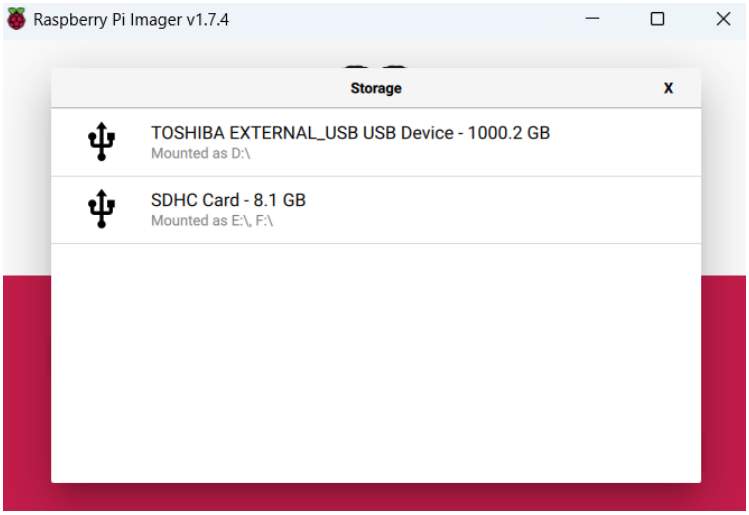
You must plug your SD card.

Then you have to click on “CHOOSE STORAGE”

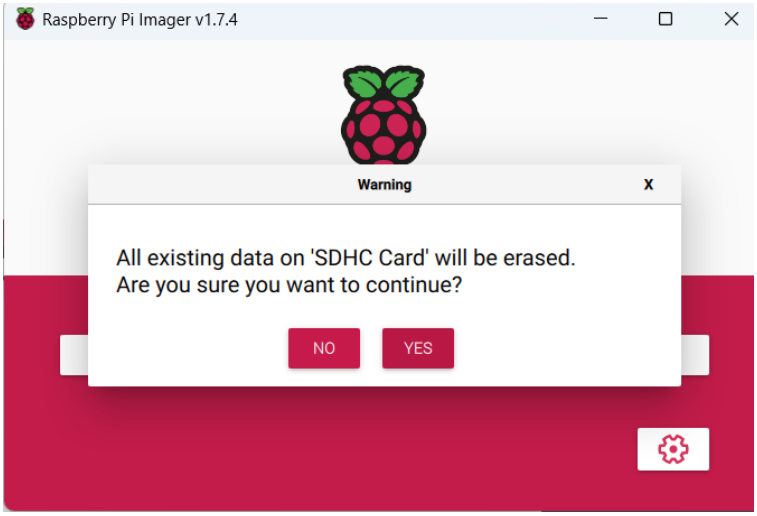
You might have more than one storage if you're using an adaptor.

Please pay attention to the one you choose, if you select a storage of yours, your data will be erased.

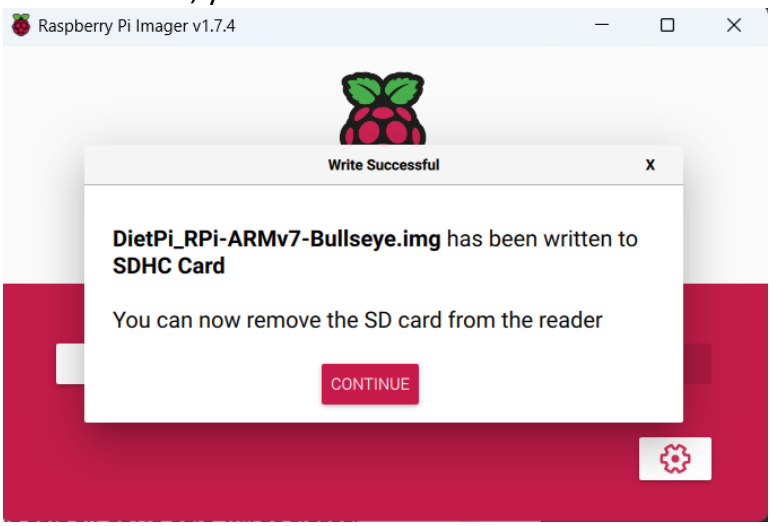
Select the "SDHC Card – 8.1 GB" storage.



Go back to the menu of the software and click on "WRITE" and select "YES".



Once it’s done, you can click on “CONTINUE”.



You have burnt the OS image on the SD card.

5. Setting up the Raspberry

5.1. Connecting the Raspberry

First, plug the micro-SD card into your Raspberry.

Then connect the RJ45 cable, the HDMI cable (and possible adaptors), and the keyboard.

Finally plug the power cable of the raspberry.

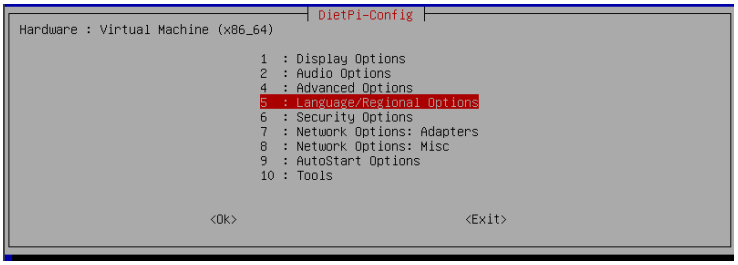
As the Raspberry starts you should end up in the update screen.

5.2. Changing keyboard settings

You might have to change your keyboard configuration (by default it is qwerty).

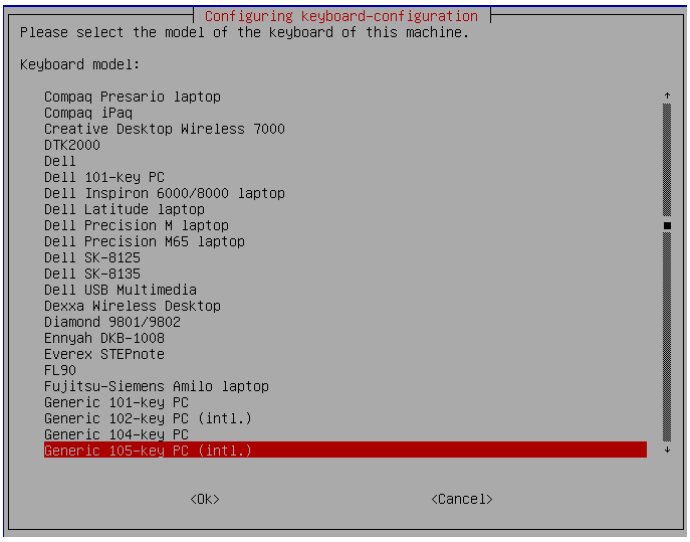
Select “**DietPi config**”. (or type “sudo dietpi-config” if you already signed in).

To change it you have to go to “5 : Language/Regional Options”.

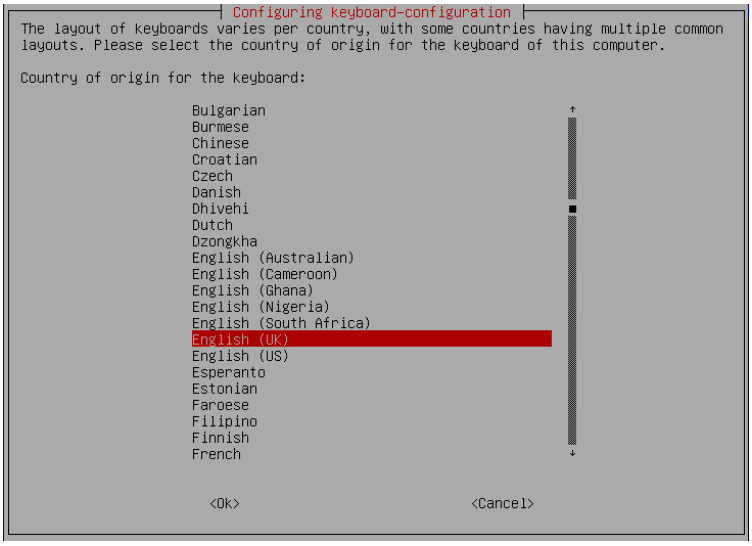


Select “Keyboard”

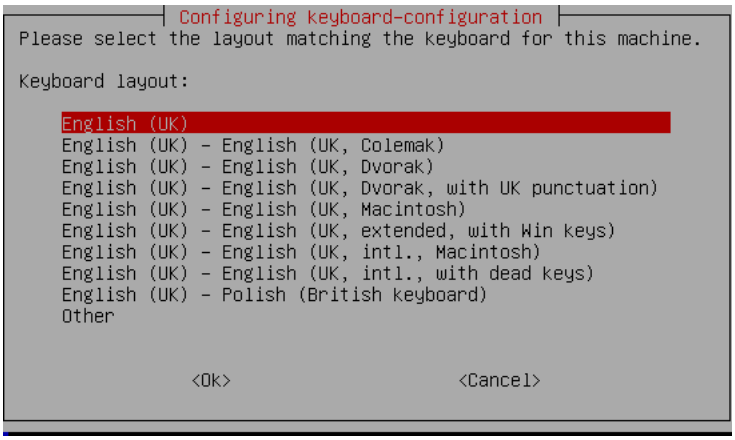
Choose the “Generic 105-key PC (intl.)”



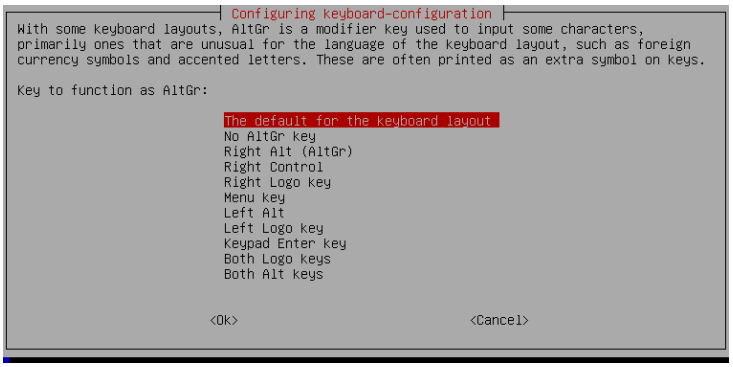
If there is not your language, please select “Other” to find yours.



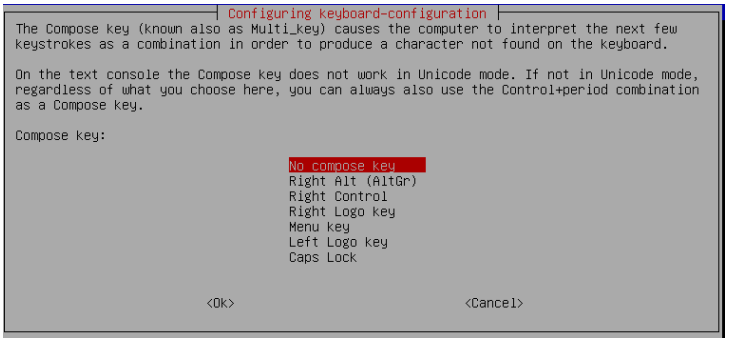
Select your keyboard layout.



Select the default keyboard layout.



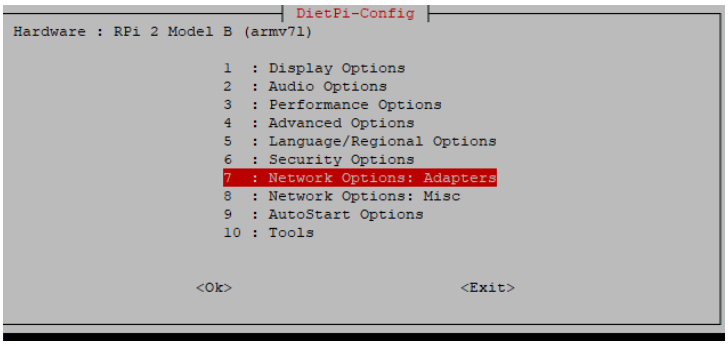
Select “No compose key”.



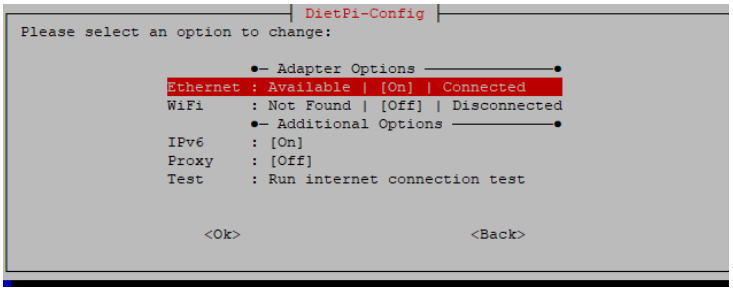
5.3. Networking

You will have to change your keyboard’s configuration.

You must select the seventh option: “7 : Network Options: Adapters”



Then select the Ethernet option.

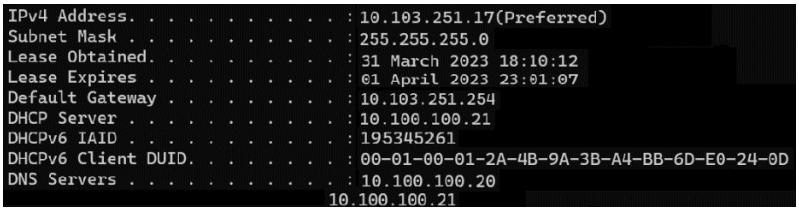


You must change the DHCP setting to STATIC to configure the IP of your server.

To help you to do it is recommended to open a console on another device to easily get the right configuration.

In CMD (Windows) type “ipconfig /all” which will return information

Find those lines.

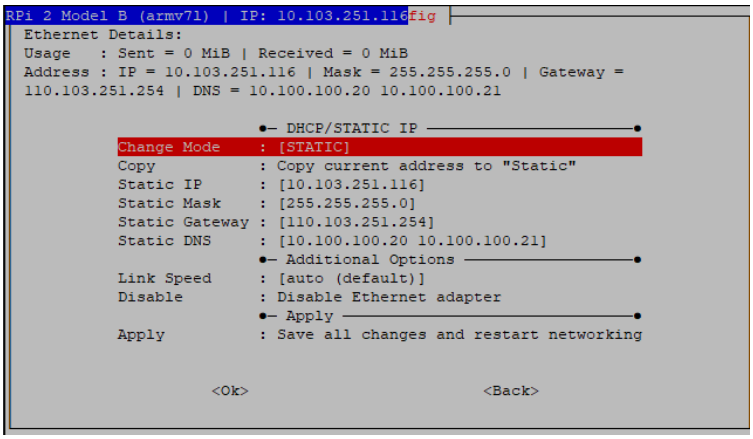


You should change “Static IP” into the one of the IPv4 Address of the CMD, except you have to change the host part of the address in order **to choose an unused one**.

If your experiencing difficulties to do that you can go to the glossary section to the “IP address” entry.

Change each of the 3 below (mask, gateway, DNS) to the corresponding one in the CMD.

Select Apply, then it should ask you if you want to drop connection, select Yes, then the same question but for Wi-Fi, this time skip.



Finally:

- Get back to the DietPi Configuration screen

- Exit the DietPi Configuration (which should bring you back to the Updating Screen)

- Select Retry

It should update.

6. Setting up SSH

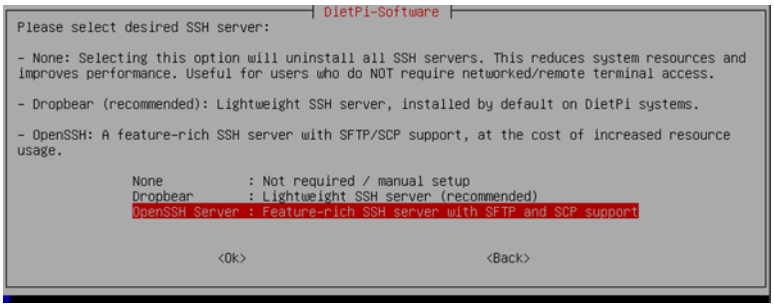
6.1. Installing OpenSSH

Sign in as root (default password is “dietpi”)

Type “**sudo diet-software**”

Select SSH Server.

Choose OpenSSH instead of DropBear.



Validate and go back to the previous screen.

Select Install (it should ask you if you want to proceed).

Select Yes.

Wait until the downloading finishes.

On the Windows’CMD type: **ssh user@IP** to test.

7. Creating a sudo user

To create a new user, you must use the command:

```
useradd -mk /etc/skel -s /bin/bash username
```

You shall replace 'username' with the chosen name of your future user.

Next create a password for your user:

```
Passwd username
```

To give sudo permission write this command:

```
usermod -aG sudo username
```

8. Setting up the server

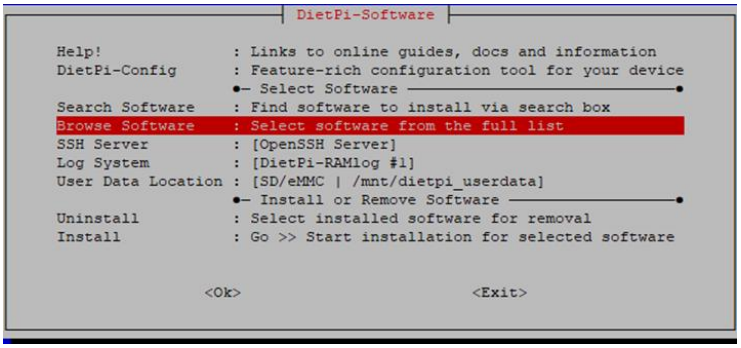
8.1. Downloading WordPress

Sign in as either root or the sudoer.

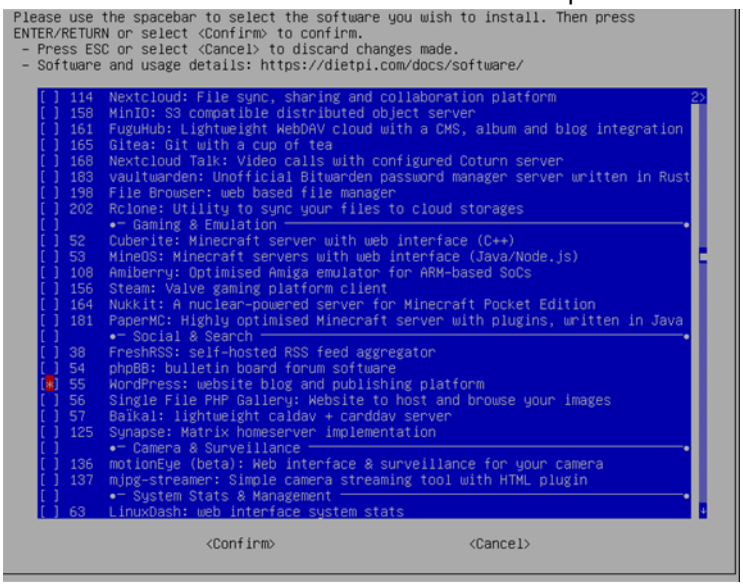
You must type:

```
sudo dietpi-software
```

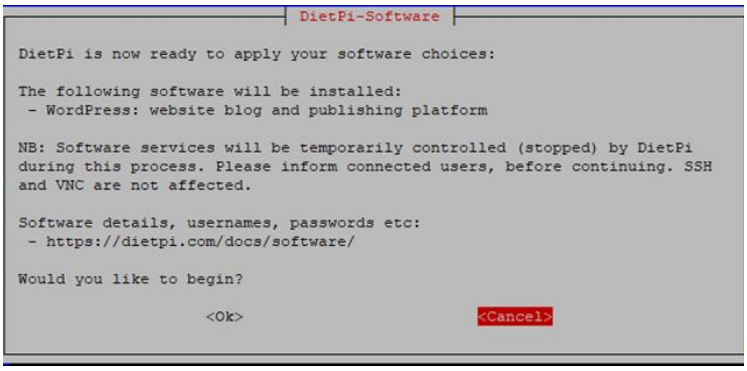
Select “Browse software”



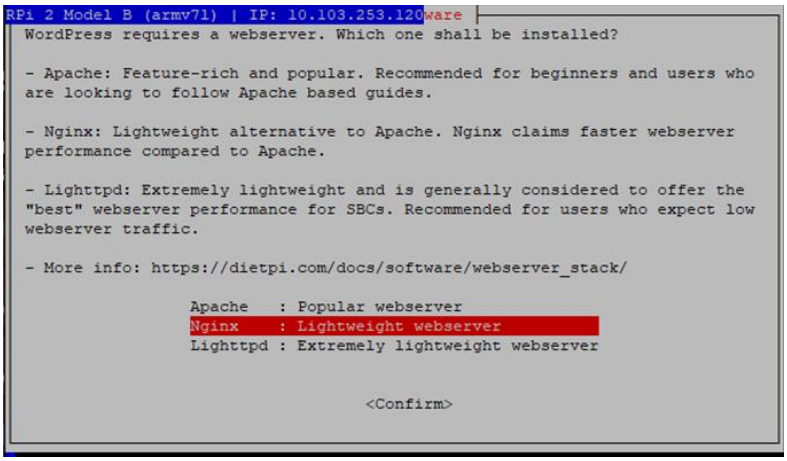
You have to search for “Wordpress”.



You may run the installation.



Next you have to select nginx as the webserver.



The install shall begin.

Please note that it will also install PHP and MariaDB.

This whole package is a LEMP stack.

8.2. Configuring MariaDB

To get a functional and customised WordPress server you have to configure your database properly.

To start the configuring the database you have to run:

```
sudo mysql_secure_installation
```

Then you will have to sign in as MariaDB's root user, there's none password set by default, but it is recommended to set one.

```
root@DavidRemiMathieuSAE3:~# sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
```

It could ask you if you want to switch to a Unix socket, respond “n” (which stands for no), it is needed to remain under TCP/IP socket.

You can change MariaDB’s root password. Respond “y” if you haven’t already set one, else “n”.

```
You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
```

Then you must respond “y” to these questions:

```
Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
```

Finally, you will create your own MariaDB’s user.

You have to access MariaDB’s prompt typing **“mysql -u root -p**” and entering the password previously set.

Then run the following command (please note that in this prompt, every command ends with “;”):

CREATE USER 'username'@localhost IDENTIFIED BY 'password';

You will grant your user access to wordpress database:

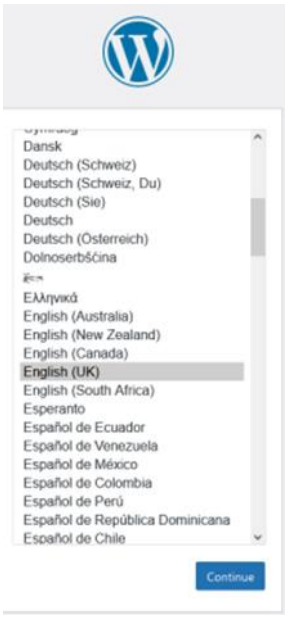
GRANT ALL PRIVILEGES ON wordpress.* TO 'username'@'localhost' IDENTIFIED BY 'password';

FLUSH PRIVILEGES;

8.3. Linking WordPress to the database

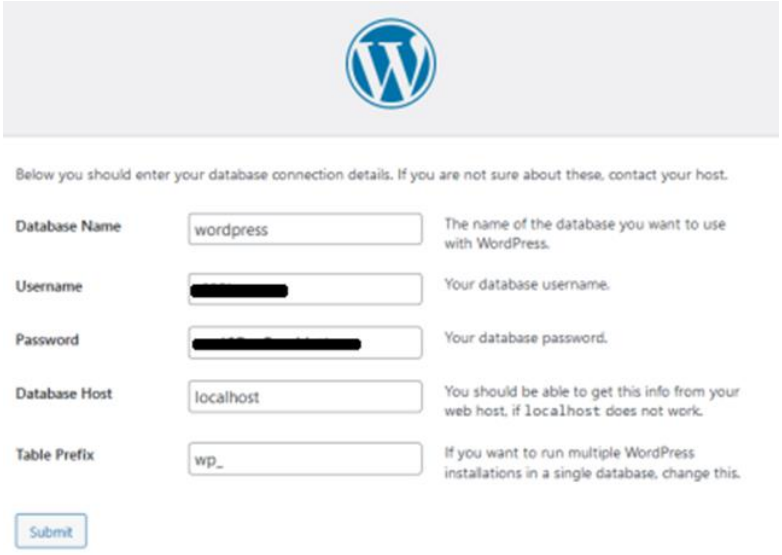
In your browser search for the IP you've configured and join "/wordpress" to it.

Select your language.

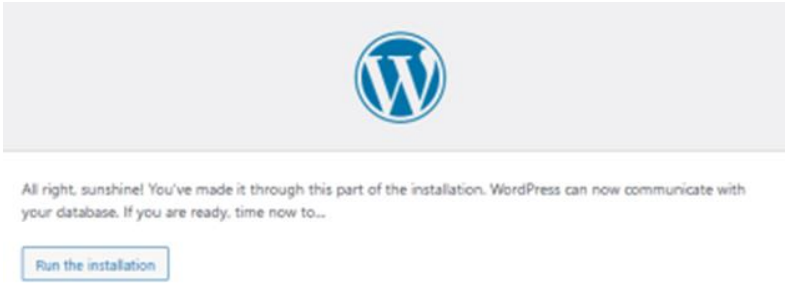


Input your data base information according to the following screen.

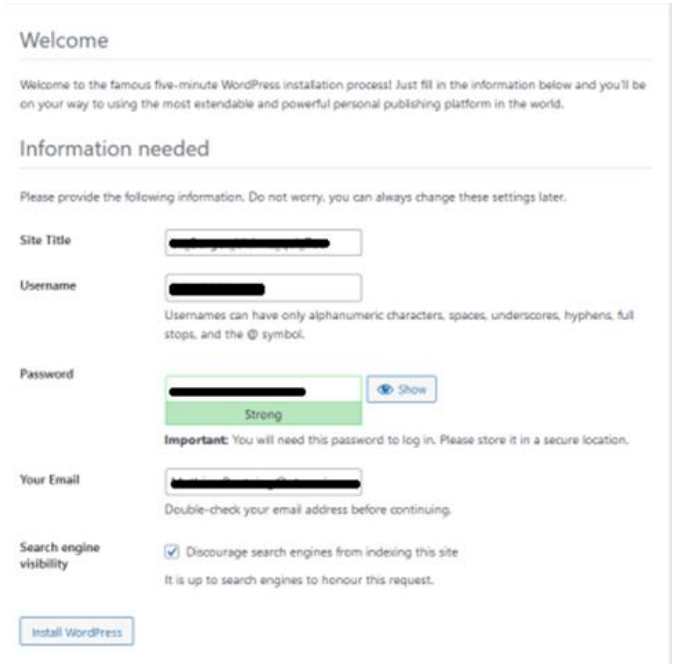
The username and the password are the ones you've previously set during the MariaDB configuration.



Click on submit then run the installation.



You may now create your website and it's admin account (it is recommended not to call it either root nor admin).



8.4. Further configuring (optional)

If you think you will have to move your raspberry to another place and so another IP address that part might be useful.

```
sudo nano/var/www/wordpress/wp-config.php
```

Add these lines

define('WP_HOME', 'http://'.$_SERVER['SERVER_ADDR'. '/wordpress');

define('WP_SITEURL', 'http://'.$_SERVER['SERVER_ADDR'. '/wordpress');

Connect to MariaDB database

```
sudo mysql -u "user"-p
```

```
use wordpress

update wordpress.wp_options set option_value='http://ip/wordpress' where option_name in ('siteurl','home');

exit;
```

```
sudo systemctl restart nginx
```

9. Setting up securities

9.1. Getting an OpenSSL certificate

Move to nginx folder:

```
cd /etc/nginx
```

Create a folder named “ssl” in which you will move.

```
root@DietPi:~# cd /etc/nginx
root@DietPi:/etc/nginx# sudo mkdir ssl
root@DietPi:/etc/nginx# cd ./ssl
root@DietPi:/etc/nginx/ssl#
```

You must create a private key and an authority.

```
sudo openssl genrsa -aes256 -out yourhostname.key 2048
```

```
root@DietPi:/etc/nginx/ssl# openssl genrsa -aes256 -out myhostname.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
..............................+++++
......+++++
e is 65537 (0x010001)
Enter pass phrase for myhostname.key:
Verifying - Enter pass phrase for myhostname.key:
root@DietPi:/etc/nginx/ssl#
```

```
sudo openssl req -new -key yourhostname.key -out youthostname.csr
```

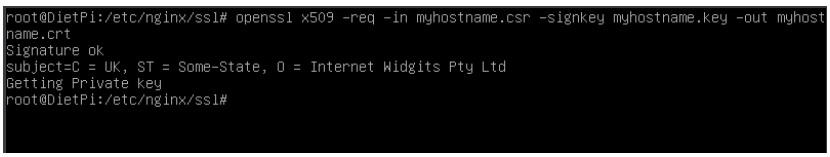
```
root@DietPi:/etc/nginx/ssl# openssl req -new -key myhostname.key -out myhostname.csr
Enter pass phrase for myhostname.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:UK
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@DietPi:/etc/nginx/ssl# _
```

```
sudo cp yourhostname.key yourhostname.key.pw
```

```
root@DietPi:/etc/nginx/ssl# cp myhostname.key myhostname.key.pw
root@DietPi:/etc/nginx/ssl# openssl rsa -in myhostname.key.pw -out myhostname.key
Enter pass phrase for myhostname.key.pw:
writing RSA key
root@DietPi:/etc/nginx/ssl# _
```

You now have to make the authority sign your key which will provide you a certificate

```
sudo openssl x509 -req -in yourhostname.csr -signkey yourhostname.key -out yourhostname.crt
```



The last step is to enable ssl on your website.

```
sudo nano/etc/nginx/sites-available/default
```

Add those lines to the file:

listen 443;

ssl on;

ssl_certificate /etc/nginx/yourhostname.crt ;

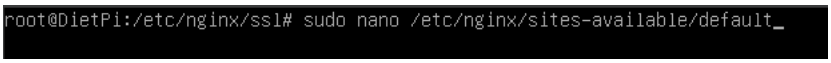
ssl_certificate_key /etc/nginx/yourhostname.key;

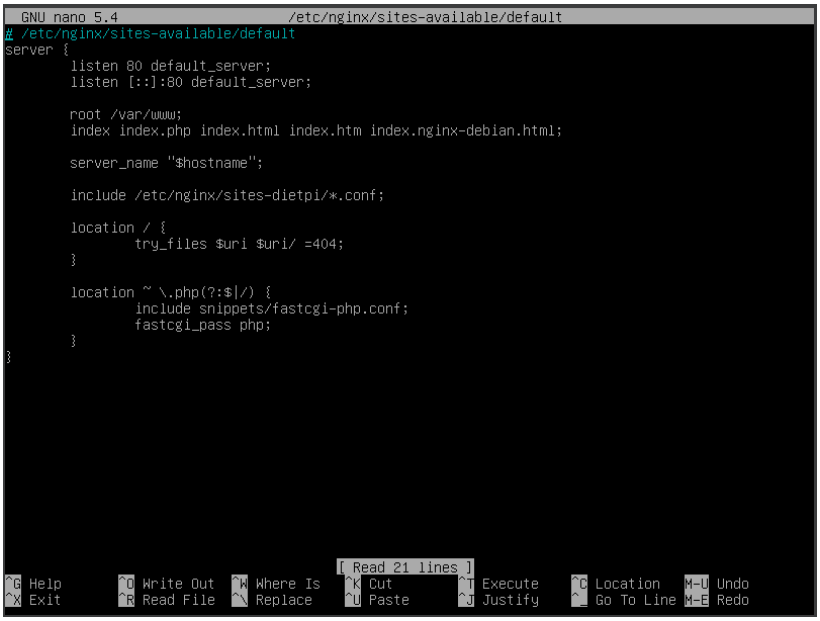
ssl_session_timeout 5m;

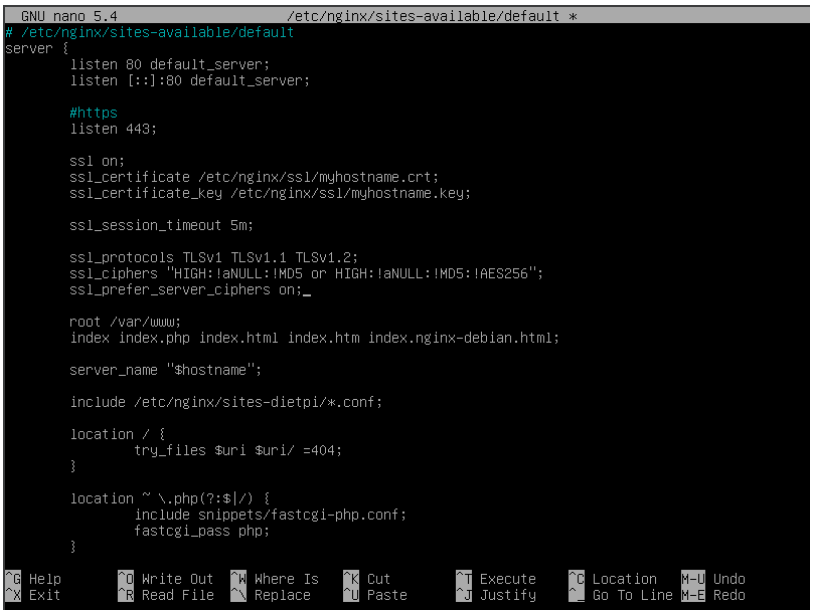
ssl_protocols TLSv1 TLSv1.1 TLSv1.2

ssl_ciphers "HIGH:!aNULL :!MD5 or HIGH:!aNULL:!MD5:!AES256";

ssl_prefer_server_ciphers: on;

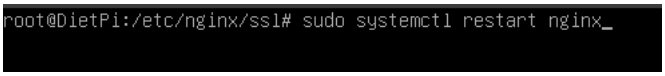






Finally you must restart nginx services.

sudo systemctl restart nginx



9.2. Configuring Fail2Ban

You will first configure it for SSH.

Move yourself to the fail2ban folder:

```
cd/etc/fail2ban
```

Then create a new jail:

```
sudo nano/etc/fail2ban/jail.d/sshd.conf
```

Finally add these lines:

[sshd]

enabled = true

port = ssh

filter = sshd

logpath = /var/log/auth.log

maxretry = 5

findtime = 600

bantime = 8600

You will now configure fail2ban for WordPress.

First connect to your WordPress dashboard.

Download “WP fail2ban” plugin and activate it.

Back on the console type this command to transfer a strong template into a new jail named wordpress.

```
sudo curl https://plugins.svn.wordpress.org/wp-fail2ban/trunk/filters.d/wordpress-hard.conf > /etc/fail2ban/filter.d/wordpress.conf
```

Finally add those lines:

enabled = true

port = http,https

filter = wordpress

logpath = /var/log/auth.log

maxretry = 4

findtime = 600

bantime = 8600

You may know restart fail2ban services doing so:

```
sudo systemctl restart fail2ban
```

In case you want to unban an IP here’s the command:

fail2ban-client set jailName unbanip IPaddress

9.3. Disabling access to root user

To disable access to your default users.

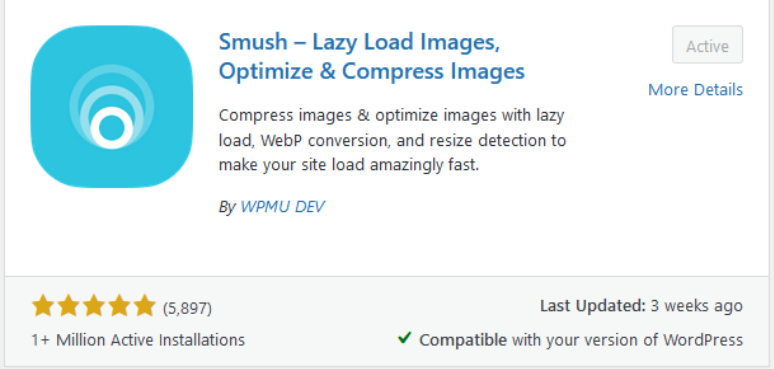
Which will prevent anyone to log in using these common names.

```
sudo usermod -L ‘username’
```

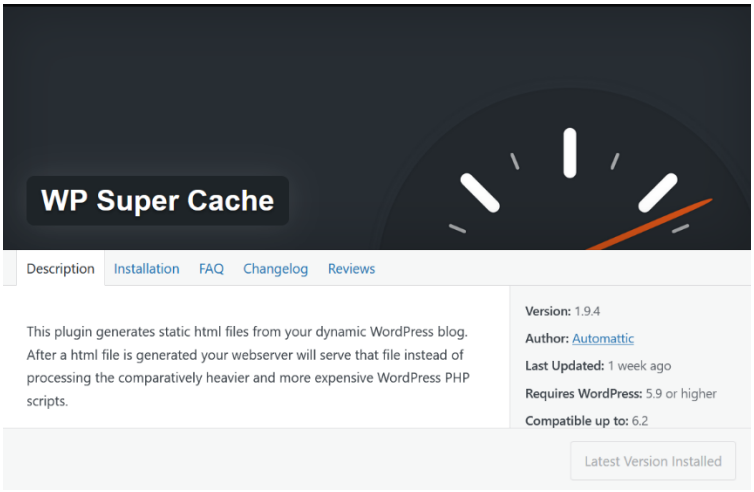
Be careful if you don’t have a sudo user available you won’t be able to change it back.

9.4. Adding WordPress plugins

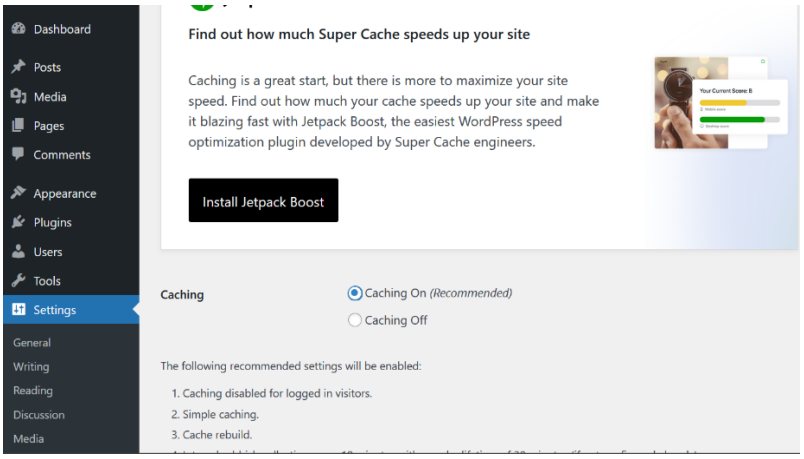
You will install the plugin name Smush (by WPMU DEV). This plugin shall speed up your website compressing images, using lazy-load.



Another additional plugin you can install is WP Super Cache which will create cache for your webpages.

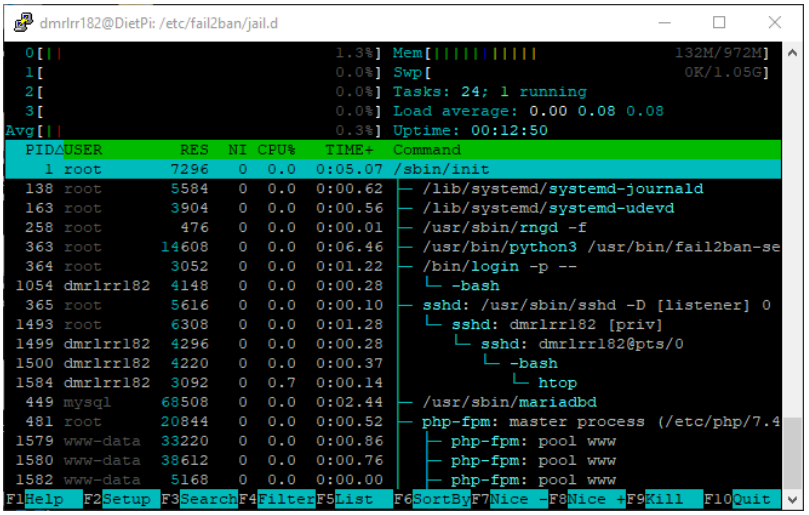


Don’t forget to enable caching.



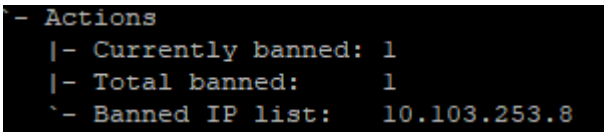
10. Testing

During normal conditions test the performances of the raspberry are the following:

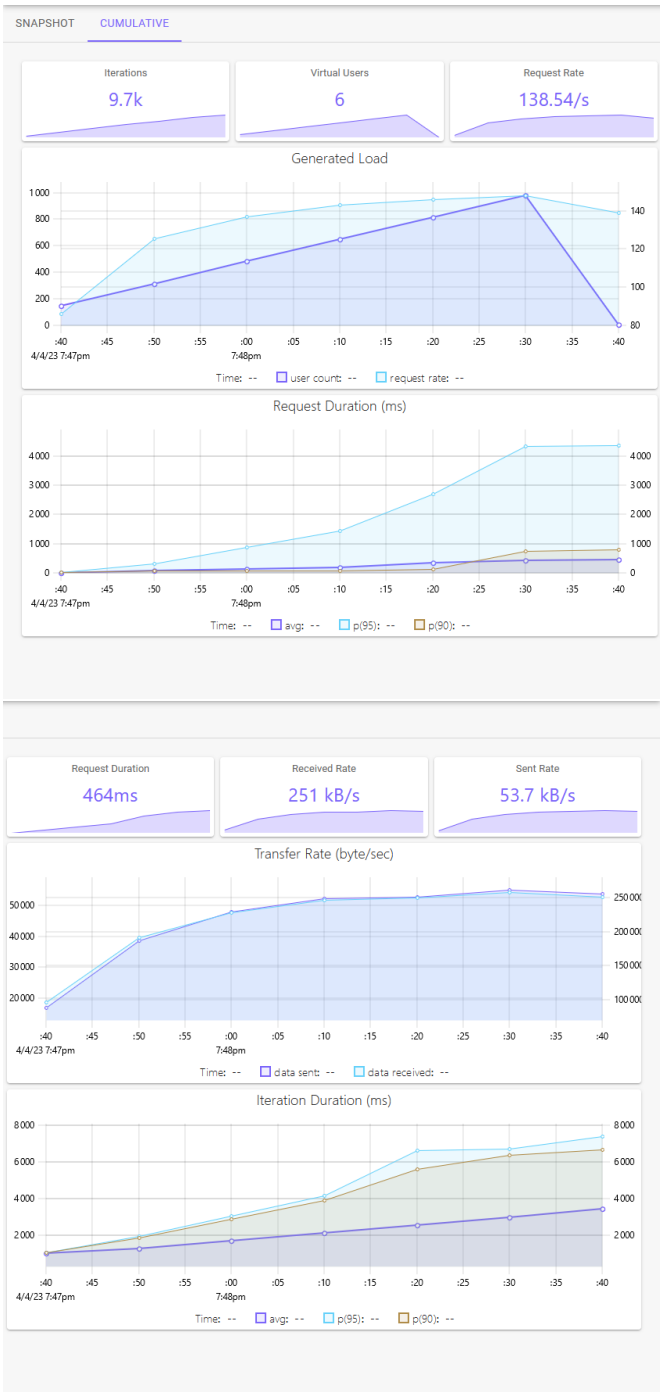


If someone try to connect to your user using SSH, he will be IP banned for the ban time you have set.

Same case with your administrator account on WordPress.



During intensive using of the website, your raspberry can bear more than 138 request per second. Your server still might slow down.



(Screens from Cassis)

Your server should be able to resists brute force attack upon users (SSH & WordPress). And can support many connections.

11. Reference materials (Error messages, troubleshooting,etc.)

11.1. Failed to connect to bus

This error occurred when you aren't connect as the super user (root), to solve it please join "sudo" at the beginning of the command.

```
dietpi@DietPi:~$ systemctl status nginx
Failed to connect to bus: No such file or directory
```

11.2. Permission denied

This error occurred when you don't have the permissions to modify files in the current directory.

To solve it you may:

- change the directory permissions.
- Join "sudo" at the beginning of the command to bypass permissions. (preferred)

```
dietpi@DietPi:/var/www$ mkdir t
mkdir: cannot create directory 't': Permission denied
```

11.3. Shortcuts

Ctrl+C: to kill the current process.

11.4. Menu navigation

Directional arrows: to move.

Space/Enter: to confirm.

Tab: to move to main button.

12. Glossary

AES256: AES-256 encryption uses the 256-bit key length to encrypt as well as decrypt a block of messages.

Ciphers: a cipher is an algorithm for encrypting and decrypting data.

CMD: Command prompt, or CMD, is the command-line interpreter of Windows operating systems.

CMS: A content management system is computer software used to manage the creation and modification of digital content.

DDOS: A distributed denial-of-service (DDoS) attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers.

DNS: The Domain Name System (DNS) is the phonebook of the Internet. DNS translates domain names to IP addresses so browsers can load Internet resources.

LEMP: The LEMP stack is a collection of Linux OS, Nginx Server, MySQL Database, and PHP. LEMP is also open source and is used to build dynamic web applications. The acronym stands for Linux, Nginx, MariaDB (MySQL), and PHP.

PHP: PHP is a general-purpose scripting language geared toward web development.

SSH: The Secure Shell Protocol (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.

SSL: Secure Sockets Layer (SSL) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser, or a mail server and a mail client.

Subnet mask: A subnet mask is defined as a 32-bit address that segregates an IP address into network bits that identify the network and host bits that identify the host device operating on that network.

Sudo user: user that can execute commands as if it was the root user.

TSL: Transport Layer Security is a cryptographic protocol designed to provide communications security over a computer network. The protocol is widely used in applications such as email, instant messaging, and voice over IP, but its use in securing HTTPS remains the most publicly visible.

WordPress: WordPress is an open-source content management system (CMS).

x509: In cryptography, X.509 is an International Telecommunication Union standard defining the format of public key certificates. X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS, the secure protocol for browsing the web.

13. Index