

**B.U.T. INFO 1**  
**Semester 2**  
**Year 2023-24**

## SAÉ 2.03 | Service Réseau

Installation and Configuration Guide for  
WordPress on Linux with Raspberry Pi

*Authors: Baptiste CREPON,  
Tanguy ABDOULVAID &  
Melih CETINKAYA*

Release 1.6  
2024/March/28

# INFO

©2024, CAC

Five ways of rainbow

All rights reserved.

Published 03/28/2024.

Printed in France

The products and/or methods of use described in this document are covered by applicable patents and other intellectual property rights.

This document may not, in whole or in part, copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from IUT ANNECY.

Every effort has been made to ensure the accuracy of this manual. However, CAC makes no warranties with respect to this documentation and denies any implied warranties of merchantability and fitness for a particular purpose. CAC shall not be liable for any error or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information in this documentation is subject to change without notice.

## Trademarks

WordPress is a registered trademark of the WordPress Foundation in the United States and/or other countries.

Linux is a trademark of Linus Torvalds in the United States and other countries.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

Apache is a registered trademark of The Apache Software Foundation.

MySQL is a registered trademark of Oracle Corporation.

PHP is a registered trademark of The PHP Group.

Microsoft, Windows, Windows NT, Windows Server, Windows Vista, and Active Directory are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

This guide provides instructions for installing and configuring WordPress on a Linux-based system using a Raspberry Pi. It aims to assist users in setting up a functional WordPress website efficiently. Throughout the guide, readers will find step-by-step instructions, tips, and important considerations for a successful installation process.

We hope this guide proves to be valuable in your endeavors to deploy a WordPress site on a Linux environment, and we welcome any feedback or suggestions for improvement.

M. Cetinkaya, T. Abdoulvaid & B. Crépon.

Students of EIA (Education Institute of Annecy),

+1 (213) 509-6995,

+1 (213) 510-4269.

<i>Introduction</i> .....	4
1.1. Purpose .....	4
1.2. Scope .....	4
1.3. System Organization .....	4
<i>Installation procedures</i> .....	5
2.1 Describing the main steps .....	5
2.2 Before installing: Prerequisites.....	5
2.3 Preparing to install .....	5
<i>Installation de L'OS</i> .....	6
<i>Configuration de l'IP fixe</i> .....	8
<i>Configuration PuTTY et Connexion au Raspberry Pi</i> .....	9
5.1. Configuration de la console PuTTY :.....	9
5.2. Connexion au Raspberry .....	10
5.3. Configuration du Raspberry .....	10
<i>Installation du serveur Web et Maria DB</i> .....	12
<i>Installation et Configuration de WordPress</i> .....	15
7.1. Création d'une base de données WordPress .....	15
7.2. Configuration de Nginx pour WordPress .....	15
7.3. Téléchargement et Configuration de WordPress .....	17
7.3.1. Procédure pré-téléchargement.....	17
7.3.2. Connection avec la BD.....	18
7.3.3. Installation de WordPress .....	19
<i>Sécurité</i> .....	20
8.1 Création nouvelle utilisateur .....	20
8.2 Installation fail2ban .....	21
8.3 Installation et configuration du pare-feu .....	21
8.4 Changer le port par défaut du SSH.....	22
<i>Test</i> .....	23
<i>Glossary</i> .....	26

# Introduction

---

## 1.1. Purpose

---

The purpose of this guide is to provide **comprehensive instructions** for **setting up** and **securing** a **WordPress** instance on a **Raspberry Pi**. It aims to assist users in efficiently establishing a fully functional WordPress website. By following the outlined steps, users can successfully deploy WordPress on a Linux system using a Raspberry Pi.

## 1.2. Scope

---

This guide covers the installation and configuration of WordPress within a Linux environment, incorporating **Nginx**, **MySQL**, and **PHP** components. It offers detailed guidance for each stage, from initial deployment to post-installation testing, performance enhancement, and security fortification. Users will gain proficiency in Linux system administration and adept web server management.

## 1.3. System Organization

---

The guide simplifies the process of WordPress installation on Linux. It begins with defining **objectives** and **prerequisites**, justifying the **choice of Linux distribution**, detailing **Raspberry Pi configuration**, explaining the **WordPress installation process**, integrating it into the IUT network, conducting **performance tests**, and documenting steps and steps.

# Installation procedures

---

## 2.1 Describing the main steps

---

The installation process encompasses **essential steps** crucial for the successful deployment of the system/service. These steps typically include:

1. Setting up the hardware environment (Raspberry Pi OS).
2. Configuring the settings of the Linux distribution.
3. Installing and configuring essential software components (Nginx, MySQL, PHP).
4. Deploying and configuring WordPress.
5. Implementing security measures across all components.

## 2.2 Before installing: Prerequisites

---

Before initiating the **installation process**, it is essential to ensure that **all prerequisites** meet. This includes:

- Raspberry Pi board (compatible model).
- MicroSD card with adequate storage.
- Compatible power supply.
- Keyboard, mouse, and monitor.
- Stable internet connection.

## 2.3 Preparing to install

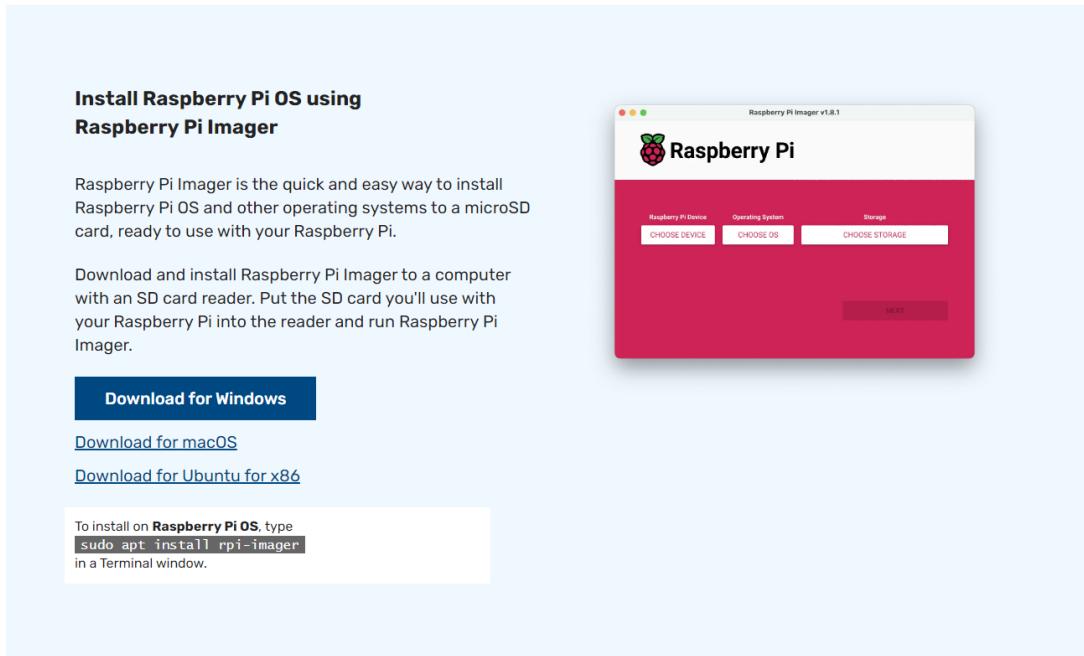
---

To **prepare** for the **installation process**, follow these steps:

1. Power off Raspberry Pi.
2. Insert MicroSD card.
3. Connect peripherals.
4. Power on Raspberry Pi.
5. Access appropriate interface.
6. Verify network connectivity.

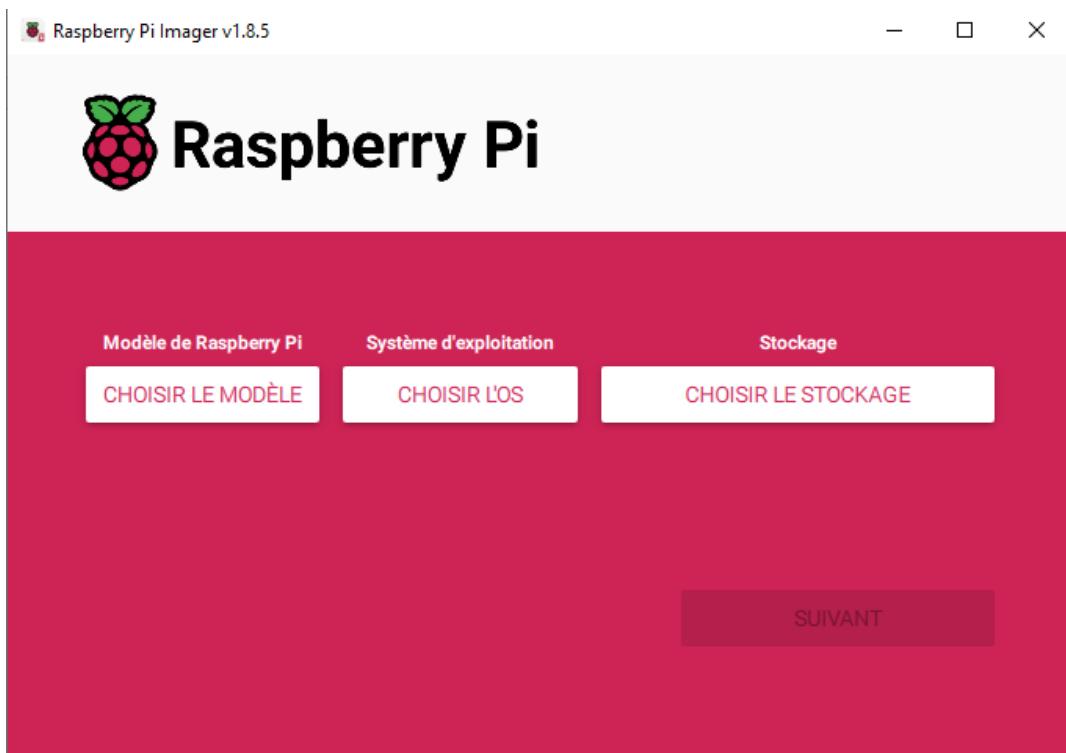
## Installation de L'OS

Aller sur le site officiel de Raspberry pi pour installer le Raspberry pi "Imager".

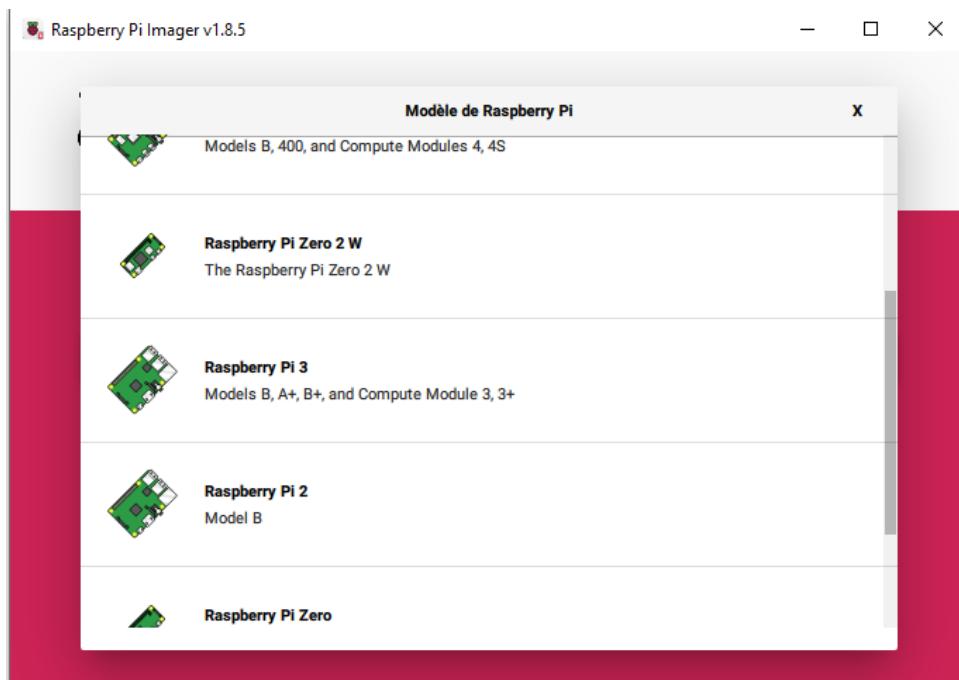


The screenshot shows the official Raspberry Pi Imager download page. At the top, it says "Install Raspberry Pi OS using Raspberry Pi Imager". Below this, there's a brief description: "Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi." It then instructs users to "Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager." There are three download links: "Download for Windows" (in a dark blue button), "Download for macOS", and "Download for Ubuntu for x86". At the bottom, there's a note: "To install on Raspberry Pi OS, type `sudo apt install rpi-imager` in a Terminal window." To the right of the text, there's a small screenshot of the Raspberry Pi Imager software interface.

Cliquez sur "**Download**" en fonction de votre système d'exploitation puis lancez le programme d'installation :

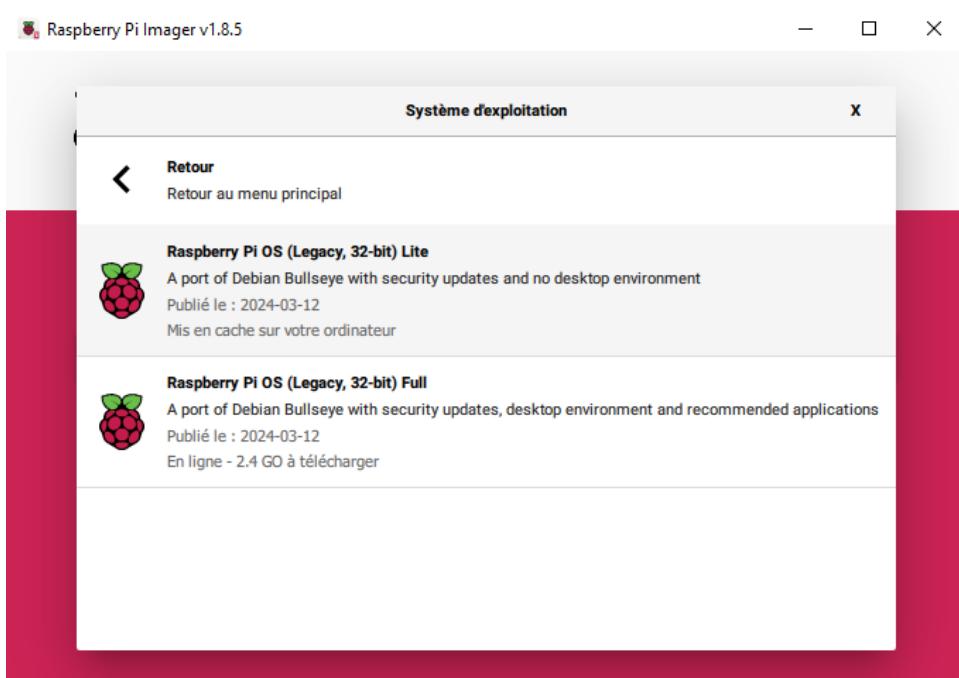


Choisissez votre modèle de Raspberry pi :



Choisissez le système d'exploitation :

Nous choisirons le système d'exploitation Raspberry pi sans environnement de bureau : Raspberry pi OS est un système d'exploitation basé sur Debian et conçu pour Raspberry pi.



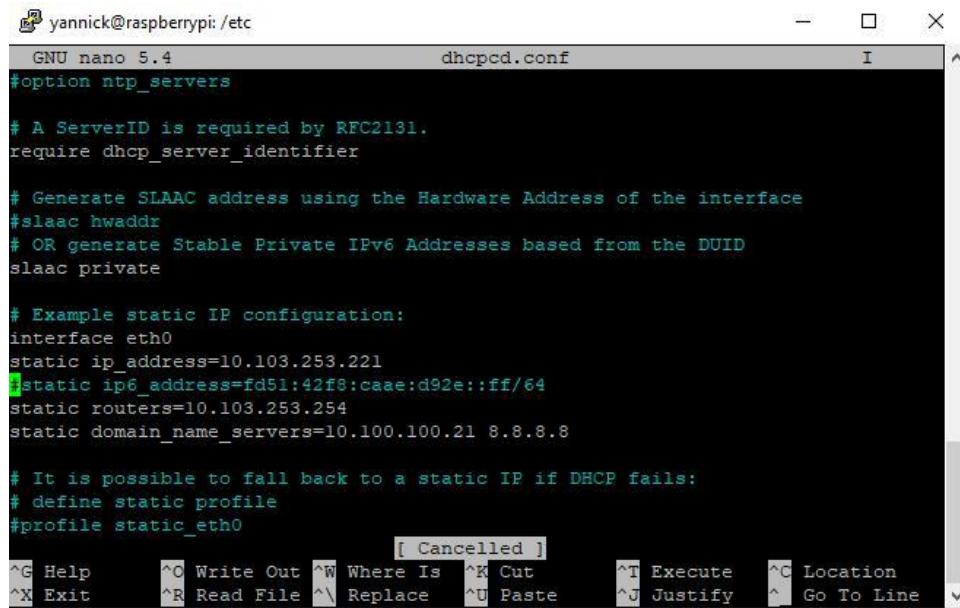
Ensute, mettez-le sur la carte SD, insérez-la dans votre appareil et allumez le Raspberry pi.

## Configuration de l'IP fixe

Pour mettre en place une adresse IP fixe il faudra éditer le fichier de configuration "**dhcp**"

**sudo nano /etc/dhcpcd.conf**

Et modifier les lignes de l'adresse IP fixe, de la passerelle et du "**dns**" dans la partie eth0 (connexion filaire avec câble rj45) comme ceci :



```
yannick@raspberrypi: /etc
GNU nano 5.4          dhcpcd.conf
#option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
interface eth0
static ip_address=10.103.253.221
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=10.103.253.254
static domain_name_servers=10.100.100.21 8.8.8.8

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
[ Cancelled ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^
^          Go To Line  v
```

Puis redémarrer la machine pour appliquer les changements :

**sudo reboot**

# Configuration PuTTY et Connexion au Raspberry Pi

## 5.1. Configuration de la console PuTTY :

Il faut d'abord trouver le **Raspberry** sur le **réseau**. Pour cela, il faut être connecté au **même réseau**. Le groupe de travail du Raspberry est par défaut le même que celui de Windows (WORKGROUP), le nom d'hôte (host NAME) du Raspberry est Raspberry Pi. Si vous êtes sur le même réseau, il vous sera facile de vérifier sa présence en ouvrant l'invite de commande de Windows et en envoyant un "**ping Raspberry Pi**", il devrait vous retourner son adresse IP.

```
Invite de commandes
Microsoft Windows [version 10.0.19041.630]
(c) 2020 Microsoft Corporation. Tous droits réservés.

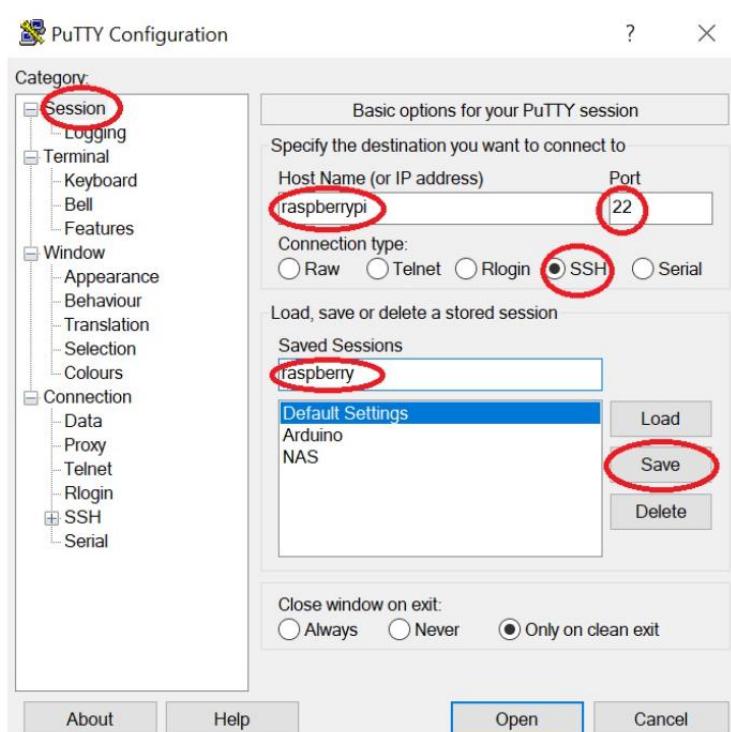
C:\Users\fabri>ping raspberrypi

Envoi d'une requête 'ping' sur raspberrypi.local [192.168.0.103] avec 32 octets de données :
Réponse de 192.168.0.103 : octets=32 temps=2 ms TTL=64
Réponse de 192.168.0.103 : octets=32 temps=2 ms TTL=64
Réponse de 192.168.0.103 : octets=32 temps=3 ms TTL=64
Réponse de 192.168.0.103 : octets=32 temps=6 ms TTL=64

Statistiques Ping pour 192.168.0.103:
Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 2ms, Maximum = 6ms, Moyenne = 3ms

C:\Users\fabri>
```

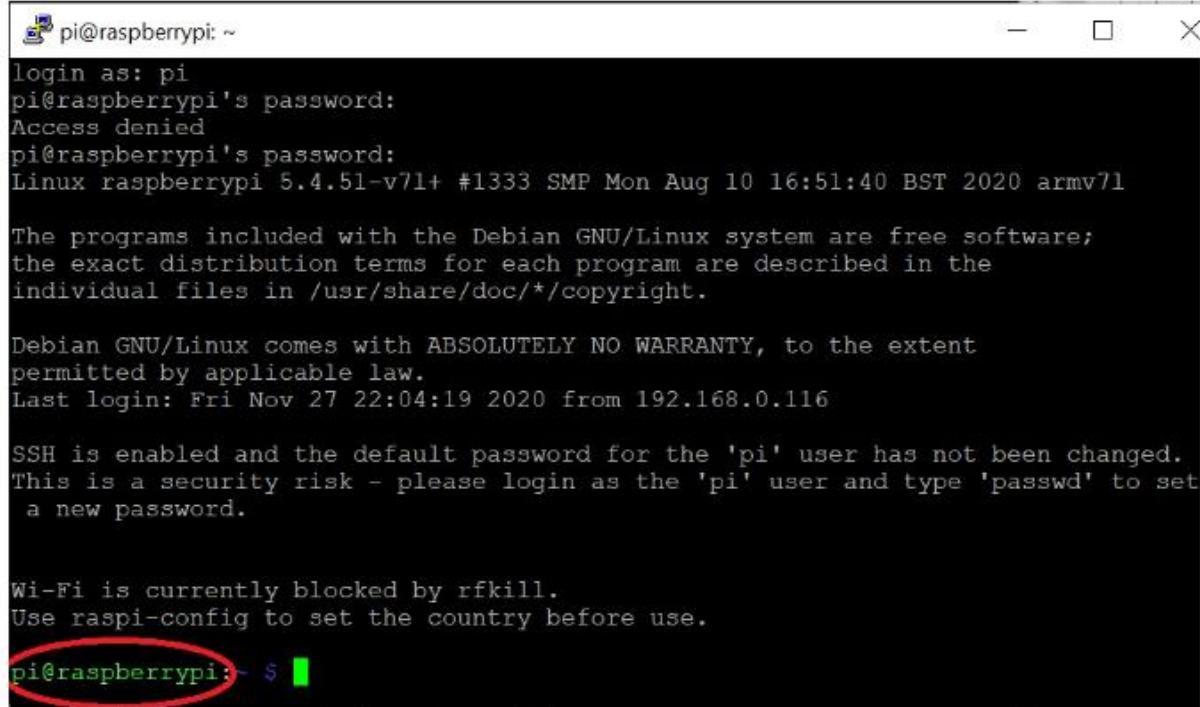
Maintenant aller sur PuTTY, installer le si vous ne l'avez pas et vérifier que vous avez bien ceci :



Si le nom d'hôte ne fonctionne pas vous pouvez le remplacer par son adresse IP qui est l'adresse fixe configurée précédemment.

## 5.2. Connexion au Raspberry

L'invite de commande du Raspberry nous demande de nous logger. Pour cela nous allons utiliser l'utilisateur par défaut "pi". Son mot de passe par défaut est "**Raspberry**" :



```
pi@raspberrypi: ~
login as: pi
pi@raspberrypi's password:
Access denied
pi@raspberrypi's password:
Linux raspberrypi 5.4.51-v7l+ #1333 SMP Mon Aug 10 16:51:40 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 27 22:04:19 2020 from 192.168.0.116

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~ $
```

Mettre à jour le cache des paquets installés :

```
sudo apt-get update
```

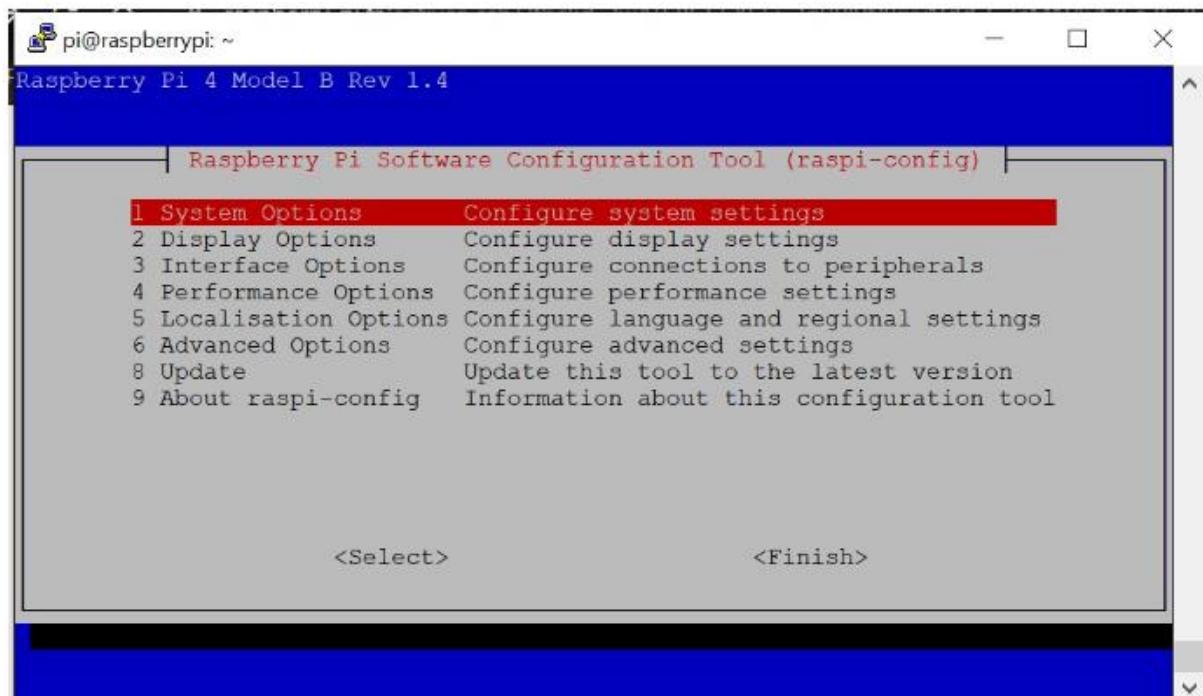
Mettre à jour le Raspberry :

```
sudo apt-get upgrade
```

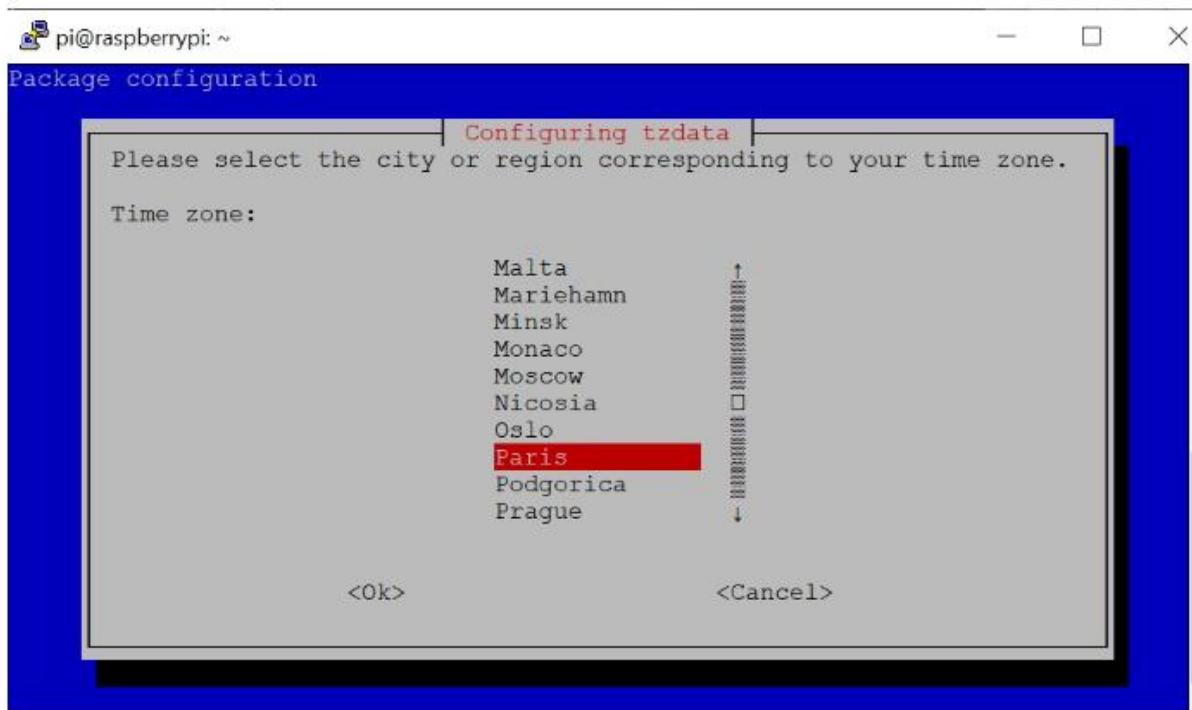
## 5.3. Configuration du Raspberry

Écrire la commande suivante pour accéder à l'**utilitaire** propre au Raspberry :

```
sudo raspi-config
```



Par exemple, vous pouvez régler le fuseau horaire :



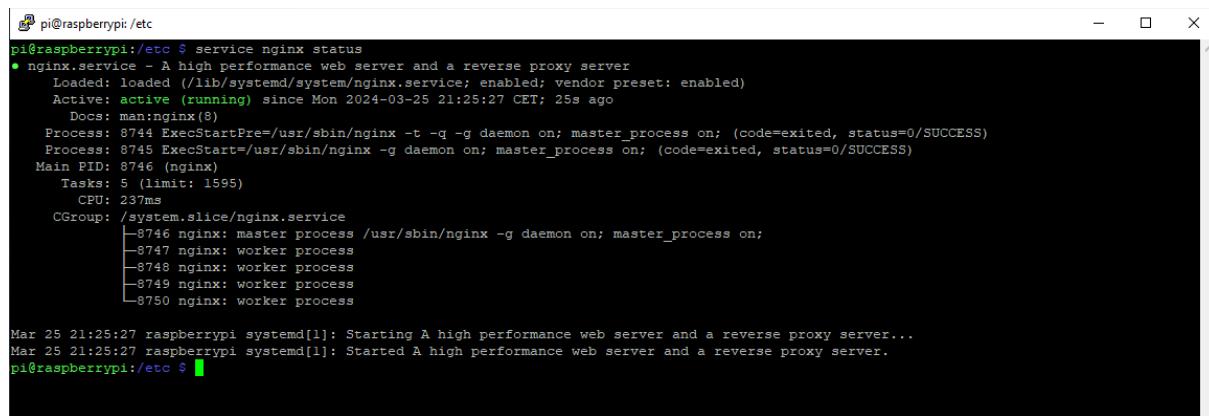
## Installation du serveur Web et Maria DB

Il existe plusieurs **serveurs Web/html**, open source. Vous avez certainement entendu parler d' **Apache**. Notre objectif étant d'optimiser les ressources, nous allons en utiliser un autre, il s'agit de **Nginx**, il est lui aussi largement utilisé et a l'avantage de consommer moins de ressources. En contrepartie certaines fonctionnalités tel que .httacces de apache ne seront pas disponibles.

Nous allons une nouvelle fois utiliser la commande "**apt-get**" suivit de "**install**", il est possible d'installer plusieurs **packages** en une seule commande :

```
sudo apt-get install nginx
```

Nginx installé :



```
pi@raspberrypi:~ $ service nginx status
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2024-03-25 21:25:27 CET; 25s ago
    Docs: man:nginx(8)
 Process: 8744 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 8745 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 8746 (nginx)
   Tasks: 5 (limit: 1595)
     CPU: 237ms
    CGroup: /system.slice/nginx.service
            └─8746 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              ├─8747 nginx: worker process
              ├─8748 nginx: worker process
              ├─8749 nginx: worker process
              └─8750 nginx: worker process

Mar 25 21:25:27 raspberrypi systemd[1]: Starting A high performance web server and a reverse proxy server...
Mar 25 21:25:27 raspberrypi systemd[1]: Started A high performance web server and a reverse proxy server.
pi@raspberrypi:~ $
```

Maintenant il faut le configurer, pour cela nous allons utiliser **un éditeur de texte : NANO** ; Le fichier de configuration se trouve sous le **répertoire "/etc"** ce sera le cas de la plupart des fichiers de configuration.

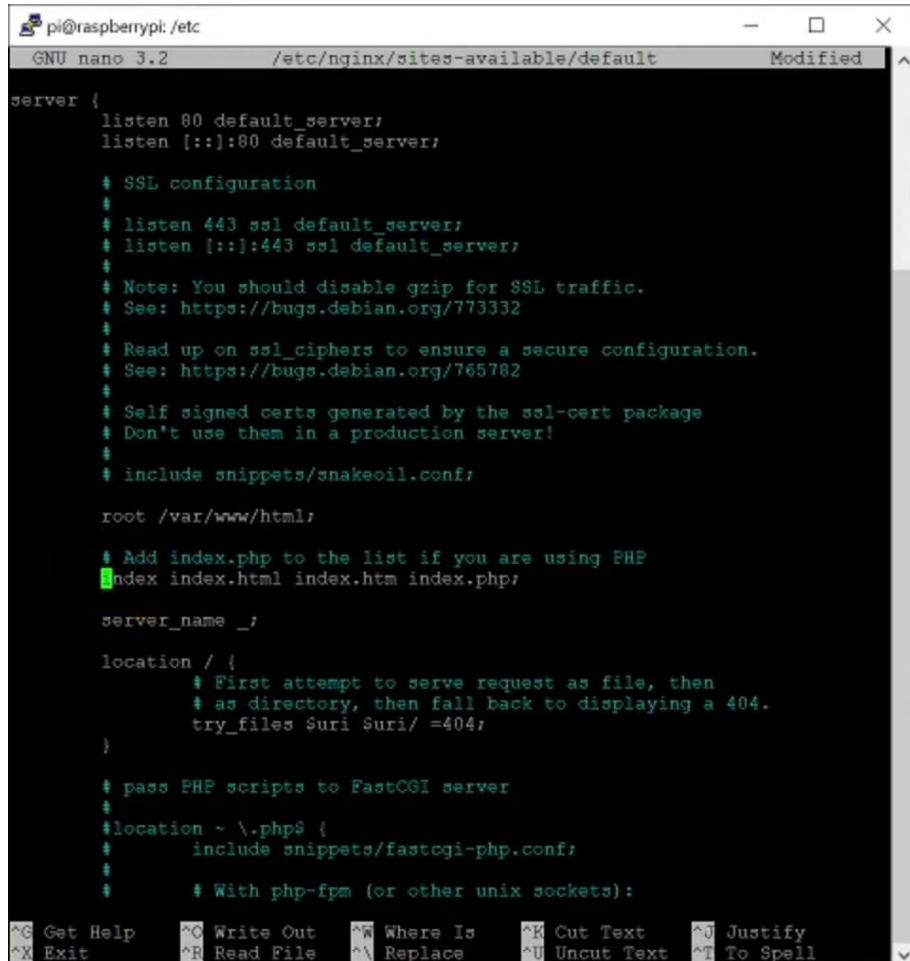
```
sudo nano /etc/nginx/sites-available/default
```

On va venir remplacer la ligne

```
index index.html index.htm index.nginx-debian.html;
```

Par

```
index index.html index.htm index.php;
```



```
pi@raspberrypi: /etc
GNU nano 3.2          /etc/nginx/sites-available/default      Modified ^

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # 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
    #index index.html index.htm index.php;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #
    #    # With php-fpm (or other unix sockets):
    #}
}

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
^X Exit     ^R Read File   ^A Replace   ^U Uncut Text  ^T To Spell
```

Puis décommenter ces lignes pour obtenir :

```
location ~ \.php$ {
include snippets/fastcgi-php.conf;
fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
}
```

```

GNU nano 3.2           /etc/nginx/sites-available/default

root /var/www/html;

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

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    #
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/run/php/php7.3-fpm.sock;
    # With php-cgi (or other tcp sockets):
    # fastcgi_pass 127.0.0.1:9000;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}

#
# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink to
# sites-enabled/ to enable it.
#
server {

```

[ Wrote 91 lines ]

[C] Get Help [O] Write Out [W] Where Is [E] Cut Text [J] Justify  
 [Q] Exit [R] Read File [P] Replace [U] Uncut Text [M] To Spell

Et finalement relancer Nginx :

**sudo /etc/init.d/nginx restart**

```

pi@raspberrypi:/etc $ sudo nano /etc/nginx/sites-available/default
pi@raspberrypi:/etc $ sudo /etc/init.d/nginx restart
[ ok ] Restarting nginx (via systemctl): nginx.service.
pi@raspberrypi:/etc $ 

```

Maintenant nous allons modifier les droits du dossier web afin de pouvoir le modifier depuis l'utilisateur pi. Pour cela nous mettons le dossier accessible au groupe pi (appartient à pi) :

**sudo chown -R www-data:pi /var/www/html/**

**sudo chmod -R 770 /var/www/html/**

```

pi@raspberrypi:/etc $ cd /var/www/html/
pi@raspberrypi:/var/www/html $ sudo chown -R www-data:pi /var/www/html/
pi@raspberrypi:/var/www/html $ sudo chmod -R 770 /var/www/html/

```

La base de données **MySQL** n'est plus libre. Elle est souvent remplacée par **Maria DB** qui est une base de données open source et compatible MySQL :

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

## Installation et Configuration de WordPress

---

### 7.1. Création d'une base de données WordPress

---

Une fois **Maria DB** installée et configurée sur votre serveur, créez un **utilisateur** et **une base de données** spécialement pour l'installation de WordPress. Pour ce faire, connectez-vous au serveur Maria DB à l'aide de la commande "**mysql -u root -pde**" et suivez les étapes décrites ci-dessous :

1. mysql -u root -p

Enter password:

2. MariaDB [mysql]> CREATE DATABASE wordpress\_db;

Query OK, 1 row affected (0.00 sec)

3. MariaDB [mysql]> GRANT ALL ON wordpress\_db.\* TO 'wpuser'@'localhost' IDENTIFIED BY 'Passw0rd!' WITH GRANT OPTION;

Query OK, 0 rows affected (0.00 sec)

4. MariaDB [mysql]> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.00 sec)

5. MariaDB [mysql]> exit

### 7.2. Configuration de Nginx pour WordPress

---

Pour commencer, créez le dossier racine de votre installation WordPress.

```
mkdir -p /var/www/html/wordpress/public_html
```

Pour créer un bloc de serveur NGINX pour votre domaine WordPress, accédez au "**/etc/nginx/sites-available**". Il s'agit de l'emplacement par défaut des blocs du serveur NGINX.

```
cd /etc/nginx/sites-available
```

Utilisez NANO précédemment qui a été utilisé, pour créer un fichier de configuration pour le bloc de serveur NGINX et modifiez-le comme ci-dessous.

## nano wordpress.conf

```
server {  
    listen 80;  
    root /var/www/html/wordpress/public_html;  
    index index.php index.html;  
    server_name SUBDOMAIN.DOMAIN.TLD;  
  
    access_log /var/log/nginx/SUBDOMAIN.access.log;  
    error_log /var/log/nginx/SUBDOMAIN.error.log;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/run/php/php7.2-fpm.sock;  
    }  
  
    location ~ /\.ht {  
        deny all;  
    }  
  
    location = /favicon.ico {  
        log_not_found off;  
        access_log off;  
    }  
  
    location = /robots.txt {  
        allow all;  
        log_not_found off;  
        access_log off;  
    }  
  
    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {  
        expires max;  
        log_not_found off;  
    }  
}
```

Vérifiez l'exactitude du fichier de configuration ci-dessus en utilisant :

**nginx -t**

Pour activer le blocage du serveur, créez un lien symbolique vers le fichier de configuration ci-dessus dans "**/etc/nginx/sites-enabled**" dossier.

**cd /etc/nginx/sites-enabled  
ln -s ../sites-available/wordpress.conf .**

Rechargez NGINX pour appliquer les nouveaux paramètres de domaine WordPress.

**systemctl reload nginx**

---

### [7.3. Téléchargement et Configuration de WordPress](#)

---

#### [7.3.1. Procédure pré-téléchargement](#)

---

Téléchargez le fichier WordPress archivé à l'aide "**wget**" et décompressez-le à la racine de l'installation WordPress que nous avons créée à l'étape précédente. Pour ce faire, exécutez les commandes suivantes depuis le terminal :

## **1. `cd /var/www/html/wordpress/public\_html`**

Change le répertoire de travail actuel vers `/var/www/html/wordpress/public_html`.  
Toutes les commandes suivantes seront exécutées dans ce répertoire.

## **2. `wget https://wordpress.org/latest.tar.gz` :**

Télécharge le fichier compressé `latest.tar.gz` depuis le site officiel de WordPress.

## **3. `tar -zxvf latest.tar.gz` :**

Extrait le contenu du fichier `latest.tar.gz` dans un dossier nommé `wordpress`.

## **4. `mv wordpress/\* .` :**

Déplace tous les fichiers et dossiers contenus dans le dossier `wordpress` vers le répertoire actuel.

## 5. `rm -rf wordpress` :

Supprime récursivement et de force le dossier wordpress une fois que son contenu a été déplacé, pour nettoyer le répertoire.

Modifiez la propriété et appliquez les autorisations appropriées aux fichiers et dossiers WordPress extraits. Pour ce faire, utilisez les commandes suivantes depuis le terminal :

### 1. cd /var/www/html/wordpress/public\_html :

Changez le répertoire courant vers le dossier de votre installation WordPress.

### 2. sudo chown -R www-data:www-data \* :

Donne au serveur web les permissions appropriées sur tous les fichiers et répertoires.

### 3. sudo chmod -R 755 \* :

Définit les permissions de manière sécurisée pour les fichiers et répertoires.

#### 7.3.2. Connection avec la BD

Fournissez maintenant **le nom de la base de données**, **l'utilisateur de la base de données** et **le mot de passe** dans le fichier de configuration WordPress afin qu'il puisse se connecter à la base de données Maria DB que nous avions créée précédemment. Par défaut, WordPress fournit un exemple de fichier de configuration et nous l'utiliserons pour créer notre propre fichier de configuration. Pour ce faire, renommez d'abord l'exemple de fichier de configuration WordPress en "**wp-config.php**" et modifiez-le avec NANO :

```
# cd /var/www/html/wordpress/public_html
# mv wp-config-sample.php wp-config.php
# vi wp-config.php
...
...
define('DB_NAME', 'wordpress_db');
define('DB_USER', 'wpuser');
define('DB_PASSWORD', 'Passw0rd!');
...
...
```

Pour sécuriser votre site WordPress, ajoutez **la clé de sécurité dans le fichier de configuration WordPress** ci-dessus juste après les options de configuration de la base de données en la générant via ce lien : <https://api.wordpress.org/secret-key/1.1/salt/>

Vous êtes maintenant prêt à installer votre site WordPress en utilisant votre navigateur préféré.

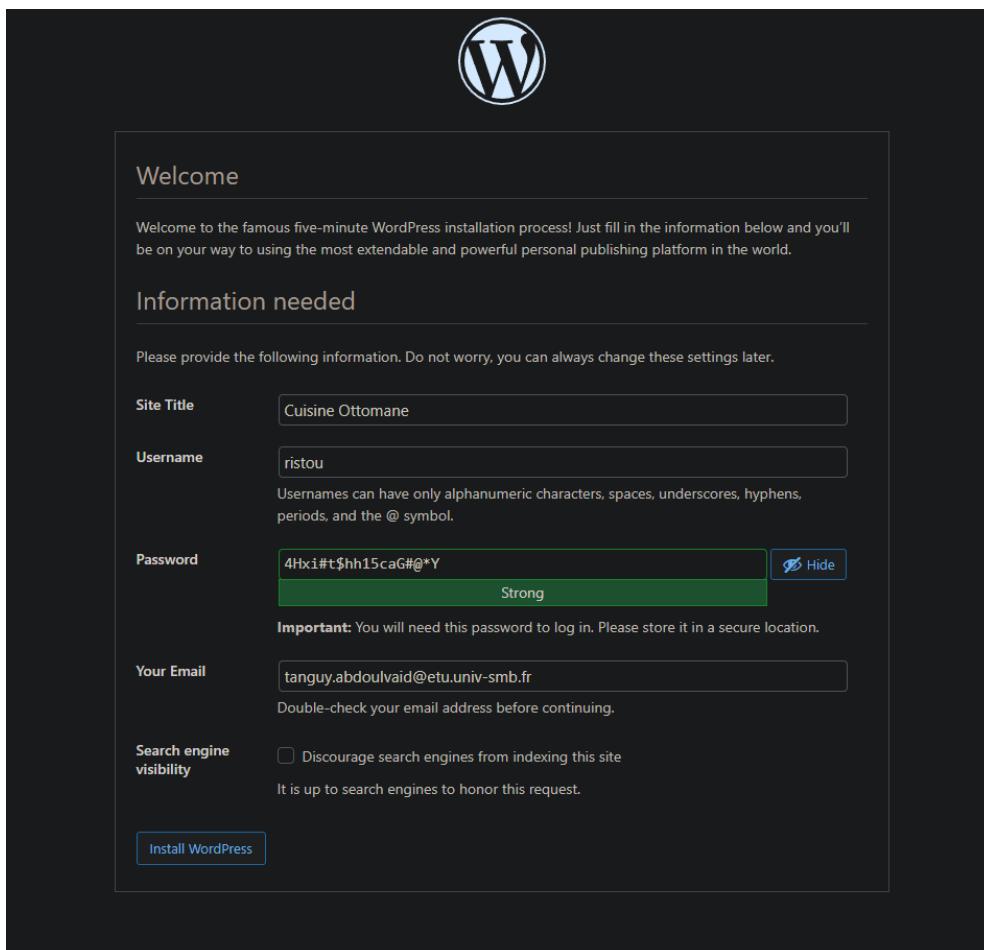
### 7.3.3. Installation de WordPress

---

Pour terminer l'installation de WordPress, allez dans votre navigateur Web préféré sur SUBDOMAIN.DOMAIN.TLD et suivez les étapes décrites ci-dessous :

Le programme d'installation vous demandera de choisir une langue. Choisissez une langue et cliquez sur « Continuer ».

Fournissez maintenant les informations du site telles que le titre du site, le nom d'utilisateur, le mot de passe, l'e-mail et cliquez sur le bouton « Installer WordPress ».



The screenshot shows the initial step of the WordPress installation process. At the top, there's a large white 'W' logo. Below it, the word 'Welcome' is centered. A brief introduction text follows: 'Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.' A section titled 'Information needed' contains fields for Site Title (Cuisine Ottomane), Username (ristou), Password (4Hxi#t\$hh15caG#@\*Y), and Your Email (tanguy.abdoulvaid@etu.univ-smb.fr). The password field includes a 'Strong' indicator and a 'Hide' button. Below the email field is a note: 'Important: You will need this password to log in. Please store it in a secure location.' Under 'Search engine visibility', there's a checkbox labeled 'Discourage search engines from indexing this site' with the note 'It is up to search engines to honor this request.' At the bottom left is a blue 'Install WordPress' button.

Vous avez terminé l'installation du site WordPress. Cliquez sur « Connexion » pour vous connecter au tableau de bord et procéder à la configuration des plugins et des thèmes pour votre site.

# Sécurité

---

## 8.1 Crédation nouvelle utilisateur

---

Pour sécuriser au maximum le **Raspberry**, la création d'un nouvel utilisateur est conseillée :

### 1. `sudo adduser <utilisateur>`

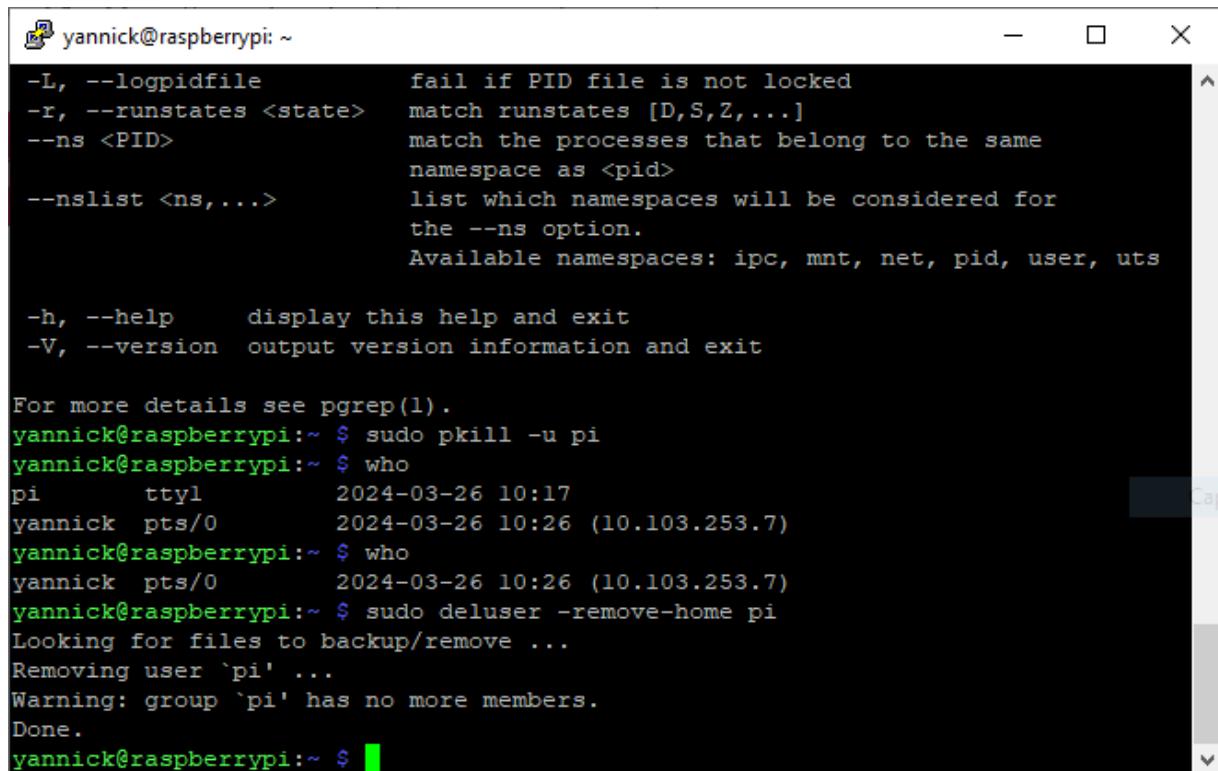
Créez un nouvel utilisateur, remplacez <utilisateur> par le nom voulu.

### 2. `sudo adduser <utilisateur> sudo`

Donnez à l'utilisateur créé les droits d'utilisation de la commande sudo.

### 3. `sudo deluser -remove-home pi`

Supprimez l'utilisateur pi par défaut.



The screenshot shows a terminal window titled "yannick@raspberrypi: ~". It displays the following command-line session:

```
-L, --logpidfile      fail if PID file is not locked
-r, --runstates <state>  match runstates [D,S,Z,...]
--ns <PID>            match the processes that belong to the same
                      namespace as <pid>
--nslist <ns,...>     list which namespaces will be considered for
                      the --ns option.
                      Available namespaces: ipc, mnt, net, pid, user, uts

-h, --help      display this help and exit
-V, --version   output version information and exit

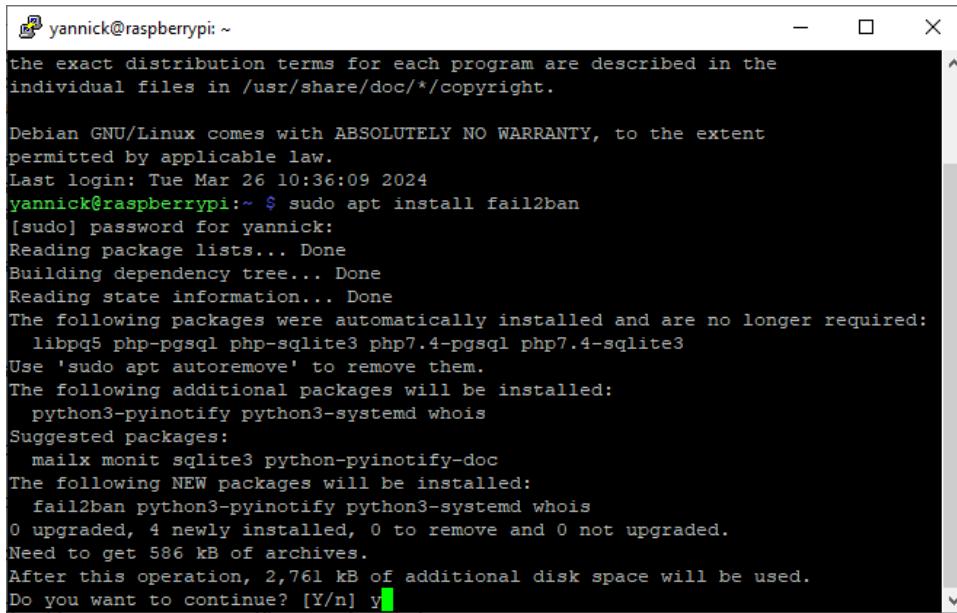
For more details see pgrep(1).
yannick@raspberrypi:~ $ sudo pkill -u pi
yannick@raspberrypi:~ $ who
pi        ttys          2024-03-26 10:17
yannick  pts/0        2024-03-26 10:26 (10.103.253.7)
yannick@raspberrypi:~ $ who
yannick  pts/0        2024-03-26 10:26 (10.103.253.7)
yannick@raspberrypi:~ $ sudo deluser -remove-home pi
Looking for files to backup/remove ...
Removing user 'pi' ...
Warning: group 'pi' has no more members.
Done.
yannick@raspberrypi:~ $
```

## 8.2 Installation fail2ban

---

L'installation de **fail2ban** va permettre de bloquer l'utilisateur après 5 essais infructueux au moment de rentrer le mot de passe de l'utilisateur. Pour installer ce package, utilisez la commande :

**sudo apt install fail2ban**



```
yannick@raspberrypi: ~
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

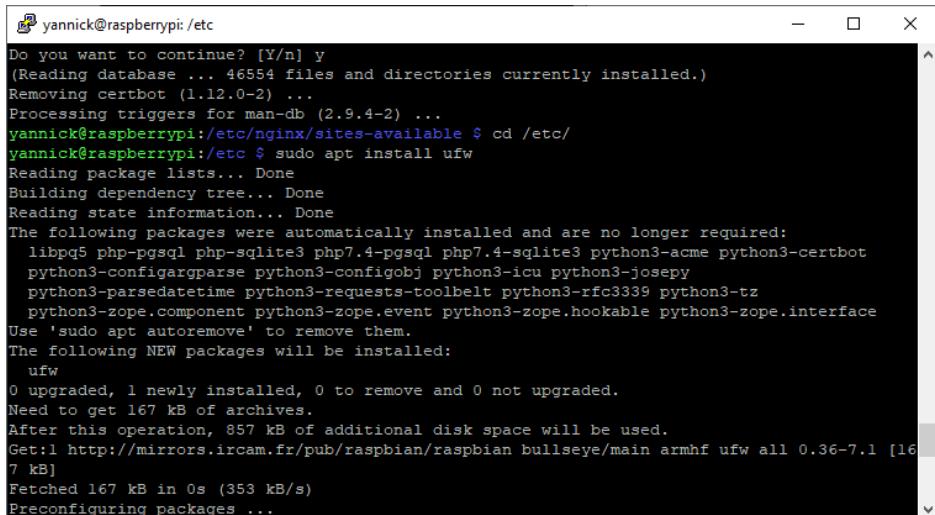
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar 26 10:36:09 2024
yannick@raspberrypi:~ $ sudo apt install fail2ban
[sudo] password for yannick:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libpq5 php-pgsql php-sqlite3 php7.4-pgsql php7.4-sqlite3
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-pyinotify python3-systemd whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify python3-systemd whois
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 586 kB of archives.
After this operation, 2,761 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

## 8.3 Installation et configuration du pare-feu

---

L'utilisation d'un pare-feu est primordiale pour sécuriser sa machine. Pour installer le pare-feu, utiliser la commande :

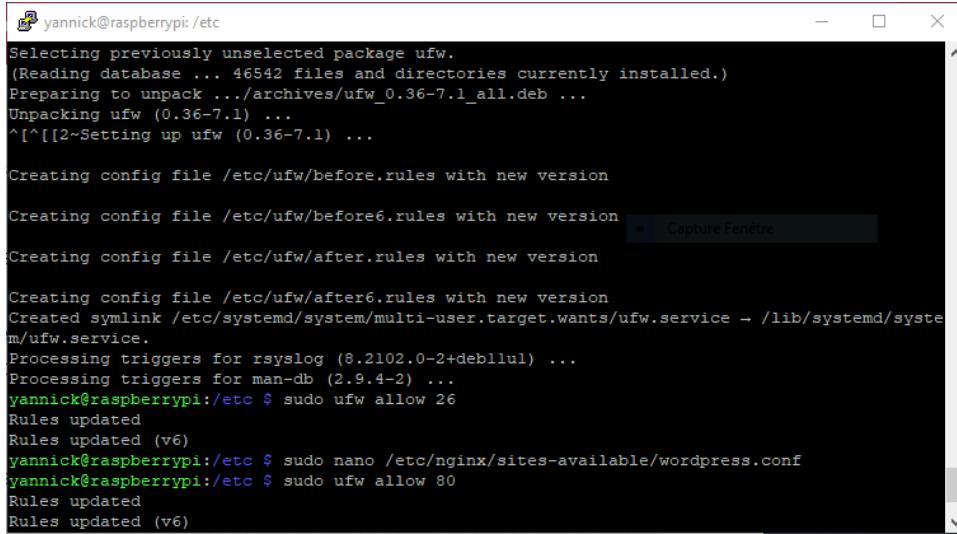
**sudo apt install ufw**



```
yannick@raspberrypi: /etc
Do you want to continue? [Y/n] y
(Reading database ... 46554 files and directories currently installed.)
Removing certbot (1.12.0-2) ...
Processing triggers for man-db (2.9.4-2) ...
yannick@raspberrypi:/etc/nginx/sites-available $ cd /etc/
yannick@raspberrypi:/etc $ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libpq5 php-pgsql php-sqlite3 php7.4-pgsql php7.4-sqlite3 python3-acme python3-certbot
  python3-configargparse python3-configobj python3-icu python3-josepy
  python3-parsedatetime python3-requests-toolbelt python3-rfc3339 python3-tz
  python3-zope.component python3-zope.event python3-zope.hookable python3-zope.interface
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ufw
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 167 kB of archives.
After this operation, 857 kB of additional disk space will be used.
Get:1 http://mirrors.ircam.fr/pub/raspbian/raspbian/bullseye/main armhf ufw all 0.36-7.1 [167 kB]
Fetched 167 kB in 0s (353 kB/s)
Preconfiguring packages ...
```

Configurez les ports que vous voulez laisser accessible, ici le port de WordPress (80) et du SSH (22 par défaut).

## **sudo ufw allow <port>**

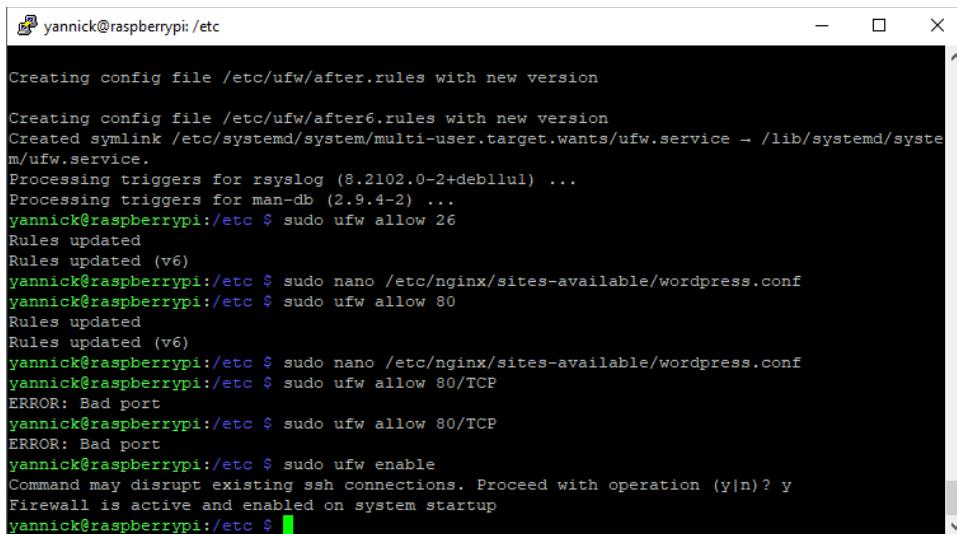


```
yannick@raspberrypi: /etc
Selecting previously unselected package ufw.
(Reading database ... 46542 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36-7.1_all.deb ...
Unpacking ufw (0.36-7.1) ...
^[[2~[Setting up ufw (0.36-7.1) ...

Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
Created symlink /etc/systemd/system/multi-user.target.wants/ufw.service → /lib/systemd/system/ufw.service.
Processing triggers for rsyslog (8.2102.0-2+deb11u1) ...
Processing triggers for man-db (2.9.4-2) ...
yannick@raspberrypi:/etc $ sudo ufw allow 26
Rules updated
Rules updated (v6)
yannick@raspberrypi:/etc $ sudo nano /etc/nginx/sites-available/wordpress.conf
yannick@raspberrypi:/etc $ sudo ufw allow 80
Rules updated
Rules updated (v6)
```

Activez le pare-feu pour le rendre fonctionnel.

## **sudo ufw enable**



```
yannick@raspberrypi: /etc
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
Created symlink /etc/systemd/system/multi-user.target.wants/ufw.service → /lib/systemd/system/ufw.service.
Processing triggers for rsyslog (8.2102.0-2+deb11u1) ...
Processing triggers for man-db (2.9.4-2) ...
yannick@raspberrypi:/etc $ sudo ufw allow 26
Rules updated
Rules updated (v6)
yannick@raspberrypi:/etc $ sudo nano /etc/nginx/sites-available/wordpress.conf
yannick@raspberrypi:/etc $ sudo ufw allow 80
Rules updated
Rules updated (v6)
yannick@raspberrypi:/etc $ sudo nano /etc/nginx/sites-available/wordpress.conf
yannick@raspberrypi:/etc $ sudo ufw allow 80/TCP
ERROR: Bad port
yannick@raspberrypi:/etc $ sudo ufw allow 80/TCP
ERROR: Bad port
yannick@raspberrypi:/etc $ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
yannick@raspberrypi:/etc $
```

## **8.4 Changer le port par défaut du SSH**

---

Allez dans le fichier de configuration SSH :

## **sudo nano /etc/ssh/sshd\_config**

Trouvez la ligne :

## #Port 22

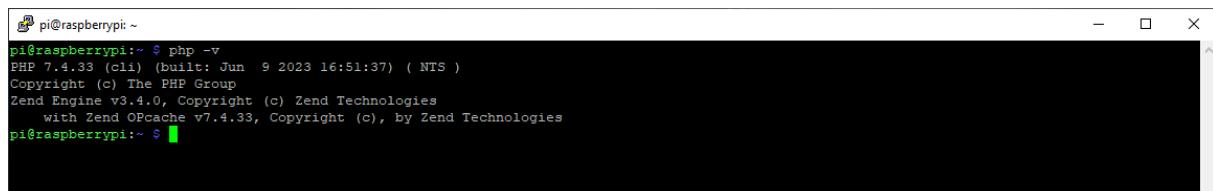
Remplacez par un port non utilisé comme 26 par exemple :

## Port 26

## Test

---

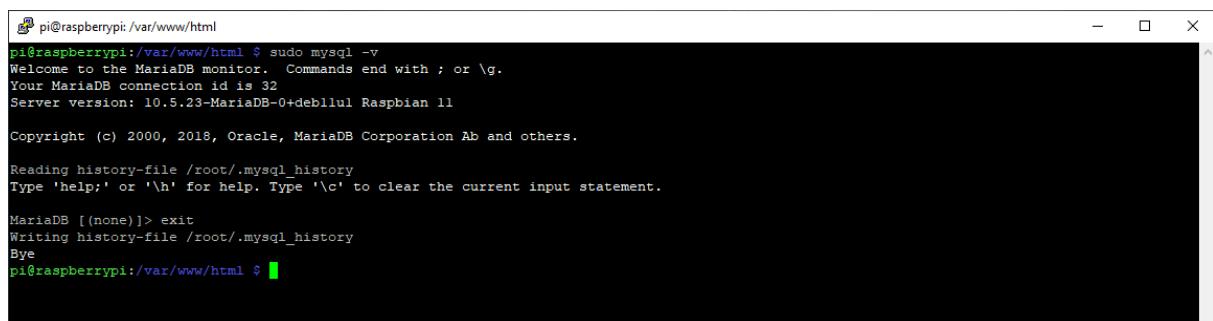
La commande "**php -v**" permet d'afficher la version installée de PHP et donc de vérifier que ce dernier est correctement installé :



```
pi@raspberrypi:~ $ php -v
PHP 7.4.33 (cli) (built: Jun 9 2023 16:51:37) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies
pi@raspberrypi:~ $
```

De même pour Maria DB qui peut être testé en ouvrant une requête "sql" avec la commande :

## **sudo mysql -v**



```
pi@raspberrypi:/var/www/html $ sudo mysql -v
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.5.23-MariaDB-0+deb11u1 Raspbian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Reading history-file /root/.mysql_history
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Writing history-file /root/.mysql_history
Bye
pi@raspberrypi:/var/www/html $
```

Pour tester la solidité de notre machine au flux de requête, on va utiliser le programme suivant :



```

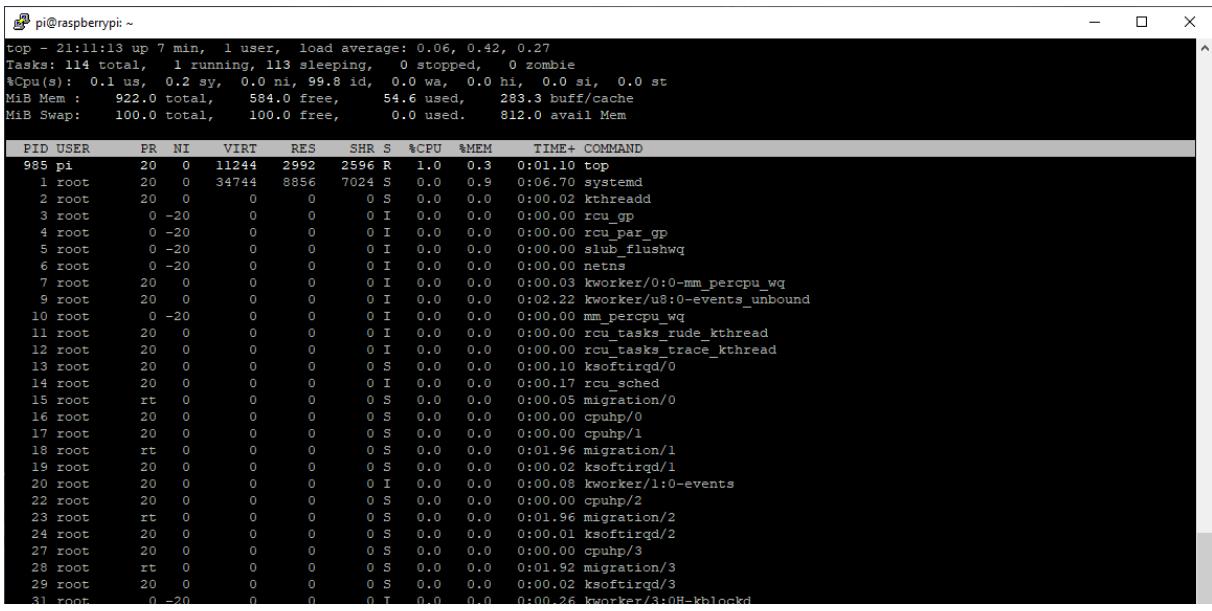
*new 1.bat - Bloc-notes
Fichier Edition Format Affichage Aide
:loop
ping 10.103.253.221 -l 65500 -w 1 -n 1
goto :loop

```

The screenshot shows a Windows Notepad window titled "new 1.bat - Bloc-notes". The content of the file is a batch script. It starts with a label ":loop", followed by a ping command that sends 65500 bytes every second to the IP address 10.103.253.221, and then loops back to the start. The Notepad window has standard menu options like Fichier, Edition, Format, Affichage, and Aide at the top. At the bottom, there are status bars for Line (Ln 3, Col 20), Zoom (100%), Encoding (Windows (CRLF)), and Character Encoding (UTF-8).

Le programme va **ping** la machine en boucle permettant de tester sa solidité.

### Gestionnaire des tâches au démarrage :



```

pi@raspberrypi: ~
top - 21:11:13 up 7 min, 1 user, load average: 0.06, 0.42, 0.27
Tasks: 114 total, 1 running, 113 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 0.2 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 922.0 total, 584.0 free, 54.6 used, 283.3 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used. 812.0 avail Mem

PID USER      PR  NI    VIRT    RES   SHR S %CPU %MEM TIME+ COMMAND
985 pi        20   0 11244 2992 2596 R  1.0  0.3 0:01.10 top
 1 root       20   0 34744 8856 7024 S  0.0  0.9 0:06.70 systemd
 2 root       20   0     0  0 S  0.0  0.0 0:00.02 kthreadd
 3 root      0 -20     0  0 I  0.0  0.0 0:00.00 rcu_gp
 4 root      0 -20     0  0 I  0.0  0.0 0:00.00 rcu_par_gp
 5 root      0 -20     0  0 I  0.0  0.0 0:00.00 slub_flushwq
 6 root      0 -20     0  0 I  0.0  0.0 0:00.00 netns
 7 root      20   0     0  0 I  0.0  0.0 0:00.03 kworker/0:0-mm_percpu_wq
 9 root      20   0     0  0 I  0.0  0.0 0:02.22 kworker/u8:0-events_unbound
10 root     0 -20     0  0 I  0.0  0.0 0:00.00 mm_percpu_wq
11 root     20   0     0  0 I  0.0  0.0 0:00.00 rcu_tasks_rude_kthread
12 root     20   0     0  0 I  0.0  0.0 0:00.00 rcu_tasks_trace_kthread
13 root     20   0     0  0 S  0.0  0.0 0:00.10 ksoftirqd/0
14 root     20   0     0  0 I  0.0  0.0 0:00.17 rcu_sched
15 root     rt   0     0  0 S  0.0  0.0 0:00.05 migration/0
16 root     20   0     0  0 S  0.0  0.0 0:00.00 cpuhp/0
17 root     20   0     0  0 S  0.0  0.0 0:00.00 cpuhp/1
18 root     rt   0     0  0 S  0.0  0.0 0:01.96 migration/1
19 root     20   0     0  0 S  0.0  0.0 0:00.02 ksoftirqd/1
20 root     20   0     0  0 I  0.0  0.0 0:00.08 kworker/1:0-events
22 root     20   0     0  0 S  0.0  0.0 0:00.00 cpuhp/2
23 root     rt   0     0  0 S  0.0  0.0 0:01.96 migration/2
24 root     20   0     0  0 S  0.0  0.0 0:00.01 ksoftirqd/2
27 root     20   0     0  0 S  0.0  0.0 0:00.00 cpuhp/3
28 root     rt   0     0  0 S  0.0  0.0 0:01.92 migration/3
29 root     20   0     0  0 S  0.0  0.0 0:00.02 ksoftirqd/3
31 root     0 -20     0  0 T  0.0  0.0 0:00.26 kworker/3:0H-kblockd

```

The screenshot shows a terminal window on a Raspberry Pi with the command "top" running. The output provides a real-time view of system tasks and processes. It includes statistics for CPU usage, memory, and swap space, along with a detailed list of processes. The processes listed include the top command itself, various kernel threads (ksoftirqd, rcu\_tasks, rcu\_sched, migration, kworker), and system daemons (systemd, kthreadd). The terminal window has a black background with white text and standard Linux-style command-line interface elements.

Gestionnaire des tâches après installation et pendant que le serveur tourne avec une personne unique connectée :

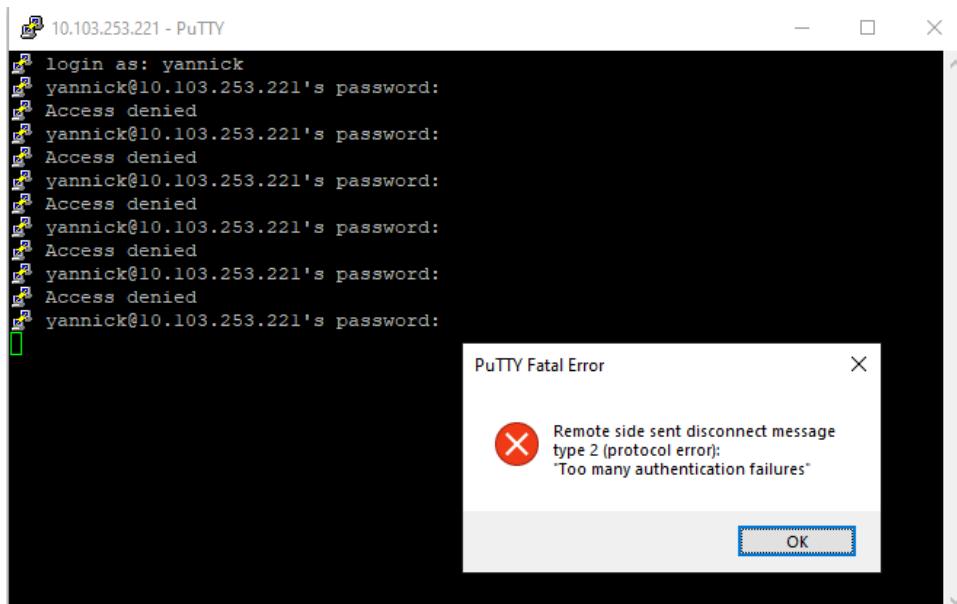
```

pi@raspberrypi: /var/www/html
top - 22:20:31 up 1:16, 1 user,  load average: 0.02, 0.07, 0.08
Tasks: 130 total, 1 running, 129 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.1 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 922.0 total, 159.0 free, 118.5 used, 644.4 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used. 718.6 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
13908 pi 20 0 11232 3152 2596 R 1.3 0.3 0:00.13 top
  1 root 20 0 34744 8968 7024 S 0.0 0.9 0:25.33 systemd
  2 root 20 0 0 0 0 S 0.0 0.0 0:00.03 kthreadd
  3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
  4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
  5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 slub_flushwq
  6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
  10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
  11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
  12 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
  13 root 20 0 0 0 0 S 0.0 0.0 0:00.53 ksoftirqd/0
  14 root 20 0 0 0 0 I 0.0 0.0 0:01.18 rcu_sched
  15 root rt 0 0 0 0 S 0.0 0.0 0:00.05 migration/0
  16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
  17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
  18 root rt 0 0 0 0 S 0.0 0.0 0:01.97 migration/1
  19 root 20 0 0 0 0 S 0.0 0.0 0:00.10 ksoftirqd/1
  22 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/2
  23 root rt 0 0 0 0 S 0.0 0.0 0:01.96 migration/2
  24 root 20 0 0 0 0 S 0.0 0.0 0:00.07 ksoftirqd/2
  27 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/3
  28 root rt 0 0 0 0 S 0.0 0.0 0:01.92 migration/3
  29 root 20 0 0 0 0 S 0.0 0.0 0:00.07 ksoftirqd/3
  31 root 0 -20 0 0 0 I 0.0 0.0 0:00.61 kworker/3:0H-kblockd
  32 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
  33 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 inet_frag_wq
  35 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kaudited
  36 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
  37 root 20 0 0 0 0 I 0.0 0.0 0:02.19 kworker/u8:1-events_unbound
  38 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom_reaper
  39 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback
  40 root 20 0 0 0 0 S 0.0 0.0 0:00.32 kcompactd0
  41 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd
  42 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd
  43 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 blkcg_punt_bio
  44 root -51 0 0 0 0 S 0.0 0.0 0:00.00 watchdogd

```

Pour tester que **fail2ban** fonctionne correctement, trompez-vous 5 fois de mot de passe au moment de la connexion et la 6e fois vous devenez obtenir ceci :



## Glossary

---

- **Raspberry Pi:** A small, single-board computer developed by the Raspberry Pi Foundation. It is used for various applications, including education, IoT (Internet of Things) projects, and as a low-cost computing platform.
- **Linux Distribution:** A version of the Linux operating system that includes the Linux kernel and additional software packages tailored for specific use cases or user preferences. Examples include Ubuntu, Debian, and Fedora.
- **Nginx:** An open-source web server software widely used for hosting websites and web applications. It is known for its stability, security, and flexibility.
- **MySQL:** An open-source relational database management system (RDBMS) used for storing and managing data in structured format. It is used as the database backend for web applications, including WordPress.
- **PHP:** A server-side scripting language used for developing dynamic web pages and web applications. It is used in conjunction with web servers like Apache to process and generate dynamic content.
- **WordPress:** A popular open-source content management system (CMS) used for creating and managing websites and blogs. It provides a user-friendly interface for content management and supports a wide range of plugins and themes for customization.
- **sudo:** A command that allows users to execute commands with administrative privileges.
- **passwd:** A command used to change a user's password.
- **ls:** A command that allows you to list the files and folders present in the folder you are currently in.
- **cd:** A command used to browse directories.