

P2022 : Sunshare
Ayaad AHAMADA

Installation et configuration (MariaDB-PhpMyAdmin-Apache)

Table des matières

1 - INSTALLATION DU SERVEUR APACHE2.....	2
1.1 - L'INSTALLATION.....	2
1.2 - VÉRIFICATION DU FONCTIONNEMENT D'APACHE.....	3
1.3 - VÉRIFICATION DU FONCTIONNEMENT D'APACHE – SI PAS D'INTERFACE GRAPHIQUE.....	3
1.4 - CONFIGURATION DU VIRTUAL HOST.....	4
2 - INSTALLATION DE PHP SUR LA RPI.....	5
2.1 - L'INSTALLATION.....	5
2.2 - VÉRIFICATION FONCTIONNEMENT PHP.....	5
3 - INSTALLER LE GESTIONNAIRE DE BASE DE DONNÉES « MARIADB».....	6
3.1 - MISE À JOUR DU SYSTÈME.....	6
3.2 - L'INSTALLATION.....	6
3.3 - ACCÈS AVEC ROOT.....	7
3.4 - CONNEXION AVEC ROOT.....	7
3.5 - CRÉER UN NOUVEL UTILISATEUR MARIADB.....	7
3.6 - TEST DE CONNEXION.....	8
4 - PHPMYADMIN, GÉRER FACILEMENT SES BASES DE DONNÉES.....	9
4.1 - L'INSTALLATION.....	9
4.2 - VÉRIFICATION DE L'INSTALLATION.....	9

1 - Installation du serveur Apache2

1.1 - L'installation

1^{er} étape mise à jour système : Avant d'installer le serveur, assurons nous d'avoir une machine bien à jour. Pour ce faire nous devons **posséder les droits administrateur**, soit en étant connecté en root, soit via la commande **sudo**.

Comme ainsi :

```
sudo apt update  
sudo apt upgrade  
sudo apt update
```

2^{ème} étape installation d'apache : une fois la Raspberry Pi à jour, nous allons installer le serveur Apache.

Comme ainsi :

```
sudo apt install apache2
```

3^{ème} Donner des droits au dossier d'apache : Cela vous permettra de facilement administrer les sites.

Pour cela, lancez les commandes suivantes :

```
sudo chown -R pi:www-data /var/www/html/  
sudo chmod -R 770 /var/www/html/
```

Signification :

chown : (**change owner**, changer le propriétaire) permet de changer le propriétaire du fichier.

Chmod : (**change mode**, changer les permissions) permet de modifier les permissions sur un fichier.

-R 770 : Donne tous les droits au propriétaire, les droits de lecture et exécution au groupe et aucuns droits aux autres...

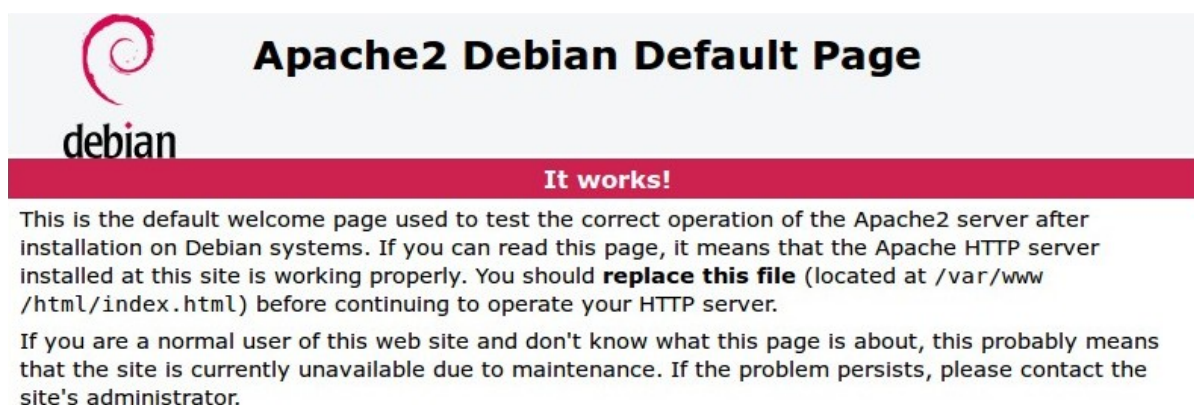
/var/www/html/ : Apache utilise le répertoire /var/www/html comme racine pour votre site.

1.2 - Vérification du fonctionnement d'apache

Une fois l'installation terminée, **tester qu'Apache fonctionne correctement** en nous rendant sur l'adresse de la Raspberry. Pour cela, il faut tenter d'accéder à la Raspberry depuis le port 80 (ce port n'étant pas encore ouvert depuis l'extérieur, **il faudra le faire depuis la Raspberry elle-même**).

Ouvrir le navigateur web de la Raspberry, et d'aller à l'adresse « <http://127.0.0.1> ». Vous devriez alors obtenir une page avec un message du genre « *It works!* ».

Vous devriez obtenir une page qui se présentera ainsi :



1.3 - Vérification du fonctionnement d'apache – si pas d'interface graphique

Si jamais vous ne possédez **pas d'interface graphique sur votre Raspberry**, ou que vous **utilisez le SSH** pour vous connecter à votre Raspberry, vous pouvez utiliser la commande suivante :

```
wget -O test_apache.html http://127.0.0.1
```

Cela va enregistrer le **code HTML** de la page dans le fichier « **test_apache.html** » dans le répertoire courant.

Vous pouvez maintenant lire le fichier avec la commande suivante :

```
cat ./test_apache.html
```

Si vous voyez marqué à un endroit dans le code « *It works!* », c'est qu'Apache fonctionne.

1.4 - Configuration du virtual host

Avec Apache, chaque site ou application web correspond en principe à un hôte virtuel (**VirtualHost** en anglais). Chaque hôte virtuel est défini par un fichier de configuration indépendant, qu'on trouve ou qu'on crée dans le répertoire `/etc/apache2/sites-available/`

Le premier *VirtualHost* est défini dans le fichier `/etc/apache2/sites-available/000-default.conf`.

Si jamais vous n'arrivez pas à accéder à votre page d'accueil, en tapant l'adresse ip de votre Rpi. Modifier le fichier conf en saisissant la commande suivante :

```
Sudo nano /etc/apache2/sites-available/000-default.conf
```

Ensuite, ajouter la ligne **DirectoryIndex** avec le chemin qui mène à votre page d'accueil. Celui-ci va déterminer quel fichier Apache doit traiter par défaut, lorsque l'utilisateur a saisi son @ip.

```
<VirtualHost *:80>

#Adresse électronique que le serveur inclut dans les messages
#d'erreur envoyés au client
ServerAdmin ayaad.ahamadaa0@gmail.com

#le chemin absolu vers l'emplacement local
DocumentRoot /var/www/html

DirectoryIndex /PHP/page_accueil.php

</VirtualHost>
```

2 - Installation de PHP sur la Rpi

Avec l'installation fait précédemment, vous pouvez **dès à présent** utiliser votre Raspberry pour faire un site **en HTML, CSS et JavaScript pur**, en interne.

Cependant, vous voudrez sans doute rapidement **permettre des interactions entre le site et l'utilisateur**. Par exemple permettre à l'utilisateur de s'inscrire, etc. Pour cela, vous allez avoir besoin de PHP.

2.1 - L'installation

La ligne de commande à saisir :

```
sudo apt install php php-mbstring
```

2.2 - Vérification fonctionnement PHP

1^{er} étape : vous allez en premier lieu **supprimer le fichier « index.html »** dans le répertoire « /var/www/html ».

```
wget rm /var/www/html/index.html
```

2^{ème} étape : puis **créez un fichier « index.php »** dans ce répertoire, avec cette ligne de commande

```
echo "<?php phpinfo(); ?>" > /var/www/html/index.php
```

3^e étape, **accéder** maintenant à la page que vous venez de créer (avec votre navigateur ou en ligne de commande).


Sur votre navigateur :

```
http://127.0.0.1/index.php
```

Ou

```
http://@ip de votre Rpi/index.php
```

Vous devriez avoir un résultat proche de cette image :

PHP Version 7.3.27-1-deb10u1	
	
System	Linux rpi4-20210210 5.9.0-0-bpo.5-arm64 #1 SMP Debian 5.9.15-1-bpo10+1 (2020-12-31) aarch64
Build Date	Feb 13 2021 16:31:40
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqld.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-curl.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gd.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-intl.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mbstring.ini, /etc/php/7.3/apache2/conf.d/20-mysql.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini, /etc/php/7.3/apache2/conf.d/20-zip.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.NTS
PHP Extension Build	API20180731.NTS
Debug Build	no

3 - Installer le gestionnaire de base de données « MariaDB »

Maintenant que nous avons mis en place PHP, vous allez sans doute vouloir stocker des informations pour les utiliser dans vos sites. Pour cela, on utilise le plus souvent des bases de données.

Nous allons donc mettre en place un SGBD (Système de Gestion de Bases de Données), à savoir MariaDB.

3.1 - Mise à jour du système:

Si cela n'est pas encore fait, voir chapitre « 1.1 L'installation », afin de mettre à jour votre raspberry.

3.2 - L'installation

1^{er} étape : Installer MariaDB avec cette commande :

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

```
pi@raspberrypi:~$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libreadline5 libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl libzlib1g
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server-10.3 mariadb-server-core-10.3 mysql-common socat
Suggested packages:
  gawk-doc libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl
  libwww-perl mailx mariadb-test tinyc
The following NEW packages will be installed:
  galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libreadline5 libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl libzlib1g
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server mariadb-server-10.3 mariadb-server-core-10.3
  mysql-common socat
0 upgraded, 33 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.5 MB of archives.
After this operation, 150 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

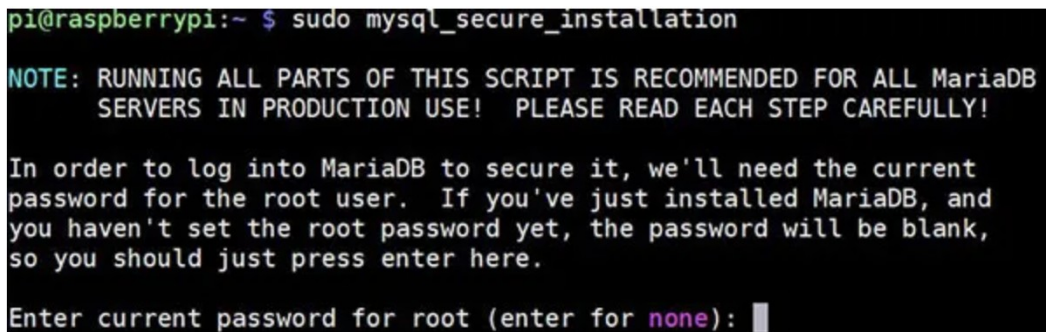
2^e étape : Tapez « Y » et Entrée pour continuer

3.3 - Accès avec root

Définir le mot de passe pour l'utilisateur root et commencer à utiliser MariaDB :

```
sudo mysql_secure_installation
```

Un message comme celui-ci va apparaître :



```
pi@raspberrypi:~ $ 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
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
```

Appuyez sur entrer pour continuer (vous n'avez pas besoin de mot de passe par défaut)

1. Tapez ensuite « Y » pour définir un nouveau mot de passe, et entrez le mot de passe de votre choix.
2. Maintenant, appuyez trois fois sur « Y » pour :
 - Supprimer les utilisateurs anonymes
 - Interdire la connexion à distance à la racine
3. Supprimer la base de données des tests
4. Et enfin, appuyez à nouveau sur « Y » pour recharger les privilèges

MariaDB est prêt à être utilisé avec pour connexion root.

3.4 - Connexion avec root

Utilisez la commande mysql pour votre première connexion :

```
mysql -uroot -p
```

3.5 - Créer un nouvel utilisateur MariaDB

Connectez-vous à la ligne de commande **MySQL** :

```
sudo mysql -uroot -p
```

Créer une nouvelle base de données :

```
CREATE DATABASE <dbname>;
```

Dans cette étape et les suivantes, remplacez toutes les variables entre <...> par le nom que vous voulez utiliser.

Créer le nouvel utilisateur :

```
CREATE USER '<username>'@'localhost' IDENTIFIED BY '<password>';
```

Ensuite, **permettez à ce nouvel utilisateur de tout faire sur la base de données** que vous venez de créer :

```
GRANT ALL PRIVILEGES ON <dbname>.* TO '<username>'@'localhost';
```

Enfin, **rechargez les autorisations** avec :

```
FLUSH PRIVILEGES;
```

3.6 - Test de connexion

Avec cette commande, vérifier que tout fonctionne bien, :

```
sudo mysql -u<username> -p
```

Et utilisez le mot de passe que vous venez de créer. Vous êtes maintenant connecté en console sur votre base de données

4 - PhpMyAdmin, gérer facilement ses bases de données

Vous pourriez vouloir **une interface un peu plus simple** pour administrer vos bases de données qu'une simple console MySQL. Pour cela, vous pouvez installer PHPMyAdmin.

4.1 - L'installation

L'installation de PHPMyAdmin se fait très simplement, via le gestionnaire de paquets, en utilisant la commande suivante :

```
sudo apt install phpmyadmin
```

PHPMyAdmin vous posera plusieurs questions concernant ses réglages.

Comme nous avons déjà configuré la base de données, choisissez **no** à la question concernant l'utilisation de `dbconfig-common`. Ensuite choisissez d'utiliser PHPMyAdmin pour **un serveur Apache**. Pour le mot de passe root, il s'agit de **celui que vous aviez utilisé pour MySQL**.

Activer l'extension **mysqli** si cela n'est pas encore fait.

```
sudo phpenmod mysqli  
sudo /etc/init.d/apache2 restart
```

4.2 - Vérification de l'installation

Sur votre navigateur utiliser l'adresse de votre Raspberry suivi de `/phpmyadmin`. Par exemple, en local ce sera «<http://127.0.0.1/phpmyadmin>».

Si vous avez une erreur, cela peut venir du fait que PHPMyAdmin se soit installé dans un autre dossier. Dans ce cas, essayez la commande

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```