

HÉBERGEMENT PI 5

Axel ROUX

CFA Saint-Félix Lasalle

Matériel utilisé :

- Raspberry PI 5 avec une alimentation de 45 W
-

Installer le serveur Web Apache

Connexion au PI 5 directement, mise à jour des packets :

sudo apt update && sudo apt upgrade -y

Redémarrage :

sudo reboot

Installons le serveur web Apache :

sudo apt-get install apache2 -y

Saisir l'adresse ip dans la barre d'adresse du navigateur. page d'accueil Apache par défaut « **It Works!** ».

Si vous ne connaissez pas l'IP, utilisez la commande ip address pour l'afficher (elle se trouve dans la section de l'interface **eth0**).

Utilisez la commande ip address pour trouver l'adresse IP de votre Raspberry Pi (dans la section **eth0**) :

ip address

Activer mod_rewrite

sudo nano /etc/apache2/apache2.conf

Faites défiler jusqu'à la section :

<Directory /var/www/>

et remplacez *AllowOverride None* par *AllowOverride All* :

<Directory /var/www/>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

```
</Directory>
```

Ensuite, enregistrer (**Ctrl + O, Entrée**) et quitter Nano (**Ctrl + X**).

On active le module apache2 :

```
sudo a2enmod rewrite
```

```
sudo systemctl restart apache2
```

Installer PHP

Installons PHP et le module PHP pour Apache :

```
sudo apt-get install php libapache2-mod-php -y
```

Vérifions que PHP fonctionne en créant un fichier info.php contenant du code affichant les extensions PHP activées :

```
sudo nano /var/www/html/info.php
```

Copiez-collez ce code dans le fichier :

```
<?php  
phpinfo();  
?>
```

Enregistrez (**Ctrl + O, Entrée**) puis quittez (**Ctrl + X**).

Dans le navigateur, on entre l'adresse IP du Pi 5 suivie de /info.php, ici :

192.168.0.2/info.php

La page Php confirme que ça fonctionne.

Une fois le test terminé, on supprime info.php pour éviter d'exposer des informations sensibles :

```
sudo rm -rf /var/www/html/info.php
```

Installer le serveur MySQL (MariaDB)

Installons le serveur de base de données (MariaDB) et le module PHP pour MySQL :

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

on lance ensuite le script de sécurisation pour renforcer la base :

sudo mysql_secure_installation

Voici les réponses recommandées aux questions du script :

1. **Enter current password for root (enter for none):** → Entrée
2. **Switch to unix_socket authentication [Y/n]:** → N
3. **Change the root password? [Y/n]:** → N
4. **Remove anonymous users? [Y/n]:** → Y
5. **Disallow root login remotely? [Y/n]:** → Y
6. **Remove test database and access to it? [Y/n]:** → Y
7. **Reload privilege tables now? [Y/n]:** → Y