

Compte rendu et Guide d'installation du serveur web sur Raspberry Pi

Ce guide détaillé vous aide à installer un serveur web complet sur un Raspberry Pi OS vierge, comprenant Apache, PHP, MariaDB, et un serveur FTP. À chaque étape, nous indiquerons clairement quand basculer entre l'interface de votre Raspberry Pi et celle de votre ordinateur.

1. Mise à jour du système

Dans le terminal de votre Raspberry Pi, veuillez exécuter ces commandes pour mettre à jour votre système :

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

2. Installation d'Apache

Dans le terminal de votre Raspberry Pi, commencez l'installation à l'aide de la commande :

```
sudo apt install apache2 -y
```

3. Vérification d'Apache

Sur votre ordinateur, ouvrez un navigateur web et saisissez :

```
http://votre_adresse_ip
```

Si Apache est installé correctement, une page par défaut d'Apache s'affichera.

4. Installation de PHP

Dans le terminal du Raspberry Pi, installez php avec la commande :

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

Vérifiez l'installation de php :

Tout d'abord créez un fichier `info.php` dans le dossier `/var/www/html/` avec le contenu suivant :

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

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

Sur votre ordinateur :

- Ouvrez un navigateur web et saisissez `http://votre_adresse_ip/info.php`. Vous devriez voir une page d'information PHP si tout est configuré correctement.

Il est possible que vous ayez constaté que l'utilisation de `sudo` est requise pour disposer des permissions nécessaires à la modification des fichiers dans le répertoire `/var/www/html/`. Pour remédier à cela, exécutez les commandes suivantes :

```
sudo usermod -aG www-data [utilisateur]
sudo chown -R www-data:www-data /var/www/html
sudo chmod -R 775 /var/www/html
```

Après ces modifications, une déconnexion puis reconnexion de la session de l'utilisateur est nécessaire pour appliquer les changements.

5. Installation de MariaDB

Dans votre terminal, veuillez lancer cette commande :

```
sudo apt install mariadb-server mariadb-client -y
```

Afin de **sécuriser** votre installation, faites la commande :

```
sudo mysql_secure_installation
```

Vous serez amené à répondre certaines questions :

- Entrer le mot de passe root actuel : Appuyez sur Entrée (si aucun mot de passe n'est défini par défaut).
- Passer à l'authentification par socket Unix : Saisissez n.
- Définir le mot de passe root : Saisissez Y, puis configurez un mot de passe sécurisé, par exemple : `RoiDuFort94`.
- Supprimer les utilisateurs anonymes : Saisissez Y.
- Interdire la connexion root à distance : Saisissez Y.
- Supprimer la base de données de test : Saisissez Y.
- Recharger les tables de privilèges : Saisissez Y.

6. Installation de l'extension PHP pour MariaDB et Connexion à MariaDB

Pour permettre à PHP d'interagir avec MariaDB, installez l'extension appropriée, puis redémarrez le serveur Apache :

```
sudo apt install php-mysql -y  
sudo systemctl restart apache2
```

Test de la Connexion entre PHP et MariaDB

a. Connexion à MariaDB avec l'utilisateur root :

```
sudo mariadb -u root -p
```

Entrez le mot de passe root que vous avez configuré précédemment (par exemple, `RoiDuFort94` si vous avez suivi les instructions).

b. Création d'une base de données pour le test :

Exécutez les commandes SQL suivantes :

```
CREATE DATABASE test_db;  
EXIT;
```

c. Création d'un fichier de test PHP :

Créez un fichier `test_db.php` et insérez le code suivant :

```
sudo vi /var/www/  
html/test_db.php  
  
<?php  
$servername = "localhost";  
$username = "root";  
$password = "votre_mot_de_passe";  
$dbname = "test_db";  
  
// Créer une connexion  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Vérifier la connexion  
if ($conn->connect_error) {  
    die("Échec de la connexion : " . $conn->connect_error);  
}
```

```
}  
echo "Connexion réussie à la base de données!";  
$conn->close();  
?>
```

Remplacez `votre_mot_de_passe` par le mot de passe root de MariaDB.

d. Test de la connexion :

Dans votre navigateur, accédez à : [http://\[ip_du_serveur\]/test_db.php](http://[ip_du_serveur]/test_db.php)

7. Installation de vsftpd (Serveur FTP)

Dans le terminal Raspberry Pi, installez :

```
sudo apt install vsftpd
```

Ensuite vérifiez l'installation du serveur FTP :

```
vsftpd -v
```

Configuration Professionnelle du Serveur FTP avec vsftpd

1. Vérification de l'installation de vsftpd

Une fois installé, vérifiez la version de `vsftpd` pour s'assurer qu'il fonctionne correctement.

2. Configuration du Serveur FTP

L'objectif est de configurer `vsftpd` pour désactiver les connexions anonymes et rediriger le FTP vers le répertoire `/var/www/html/`. Voici les étapes :

- **Modification du Fichier de Configuration :**

Ouvrez le fichier `/etc/vsftpd.conf` avec l'éditeur `vi` :

```
sudo vi /etc/vsftpd.conf
```

- Paramètres à Ajuster :

- Désactiver les connexions anonymes :

Assurez-vous que la ligne suivante est définie sur `NO` et n'est pas commentée :

```
anonymous_enable=NO
```

- Activer l'accès pour les utilisateurs locaux :

Vérifiez que la ligne suivante est décommentée :

```
local_enable=YES
```

- Autoriser les opérations d'écriture :

Assurez-vous que cette ligne est définie sur `YES` :

```
write_enable=YES
```

- Définir le répertoire racine FTP :

Ajoutez ces lignes pour chrooter les utilisateurs locaux dans ``/var/www/html/`` :

```
user_sub_token=$USER  
local_root=/var/www/html/
```

- Autoriser l'écriture dans un répertoire chrooté :

Ajoutez la ligne suivante pour éviter les erreurs liées aux permissions :

```
allow_writeable_chroot=YES
```

- Application des Changements :

Une fois les modifications apportées, redémarrez le service pour appliquer la configuration :

```
sudo systemctl restart vsftpd
```

À ce stade, le serveur FTP est sécurisé et les connexions sont correctement redirigées vers le répertoire ``/var/www/html/``, qui contient les fichiers publics de votre serveur web Apache.

8. Connexion FileZilla

Sur votre ordinateur , utilisez un client FTP comme FileZilla pour vous connecter à [votre_adresse_ip](#) avec vos identifiants.

Connexion au Serveur FTP via un Client FTP

Pour accéder au serveur FTP à distance, un client FTP est nécessaire, tel que FileZilla.

- Installation de FileZilla (sous Debian) :
- Passez en mode administrateur avec :

```
su -
```

- Entrez le mot de passe administrateur puis installez FileZilla :

```
apt install filezilla
```

- Quittez le mode administrateur :

```
exit
```

Cette configuration permet désormais une connexion sécurisée au serveur FTP, facilitant la gestion des fichiers du serveur web à distance.