

Serveur LAMP et FTP

Serveur LAMP et FTP

Un serveur **FTP** (File Transfer Protocol) est un serveur qui permet d'échanger des fichiers entre un client et un serveur.

Un **serveur LAMP** est un ensemble de logiciels open source souvent utilisés ensemble pour héberger des sites web dynamiques et des applications web.

LAMP est l'acronyme de :

- **L**inux
- **A**pache
- **M**ariaDB ou **M**ySQL
- **P**hp

Au cours de ce TP nous allons mettre en place un serveur **LAMP** et un serveur **FTP** afin d'héberger un site web avec des droits d'accès différents selon les utilisateurs.

Sommaire

1. [Etape 1 : Préparation VM](#Etape 1 : Préparation VM)
 2. [Etape 2 : Serveur LAMP](#Etape 2 : Serveur LAMP)
 3. [Etape 3 : Serveur FTP](#Etape 3 : Serveur FTP)
 4. [Etape 4 : Fichier info.php](#Etape 4 : PHP)
 5. [Etape 5 : Pages HTML](#Etape 5 : Pages HTML)
 6. [Etape 6 : Pages HTML sécurisée](#Etape 6 : Pages HTML Sécurisée)
-

Etape 1 : Préparation VM

On utilise pour ce TP on utilise une VM Debian 12 que l'on va maintenant pouvoir préparer.

Sur la VM on installe les services suivants :

- **ssh** : Permet de se connecter à distance
- **htop** : Outil en ligne de commande similaire au gestionnaire des tâches Windows
- **Midnight Commander** : Un gestionnaire de fichiers en mode semi-graphique.

```
apt install ssh
apt install htop
apt install mc
```

On vérifie que l'on peut se connecter avec le ssh.

```
root@maxence-ThinkPad-P16s-Gen-2:/home/maxence# ssh sio@192.168.1.45
The authenticity of host '192.168.1.45 (192.168.1.45)' can't be established.
ED25519 key fingerprint is SHA256:77ce5f9a9B5oJydUwWdvdAyGFjlppGbUqIG30qjBaVY.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:12: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.45' (ED25519) to the list of known hosts.
sio@192.168.1.45's password:
Linux debian12 6.1.0-38-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.147-1 (2025-08-02) x86_64

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: Mon Sep  8 11:43:05 2025
sio@debian12:~$
```

On peut bien se connecter en ssh

Etape 2 : Serveur LAMP

On peut maintenant mettre en place le serveur **LAMP**.

 On installe les services apache2, MariaDB et Php :

```
apt install apache2  
apt install mariadb-server  
apt install php
```

 On termine maintenant la configuration de mariadb

```
mysql_secure_installation
```

 On nous demande alors d'entrer le mot de passe root pour la base de données, on choisit le mot de passe *Mot2pass*.

Etape 3 : Serveur FTP

On passe maintenant à l'installation et à la configuration du serveur FTP

Installation du serveur FTP :

On choisit d'utiliser **vsftpd** pour le serveur **FTP**.

```
apt install vsftpd
```

 On veut faire en sorte que 2 utilisateurs est accès au serveur FTP avec des autorisations différentes.

- Un utilisateur *élève* avec des droits uniquement en **lecture**.
- Un utilisateur *prof* avec des droits en **lecture, écriture, création de répertoires et suppression**.

Modification du fichier de configuration :

On commence par modifier le fichier de configuration :

```
sudo nano /etc/vsftpd.conf
```

On ajoute les lignes suivantes au document :

```
write_enable=YES  
chroot_local_user=YES  
allow_writeable_chroot=YES
```

Création des utilisateurs :

On va maintenant créer les deux utilisateurs, définir les mots de passes et créer leurs répertoires personnelles.

```
adduser eleve
```

mot de passe *azerty*

```
adduser prof
```

mot de passe *qwerty*

Configuration des permissions :

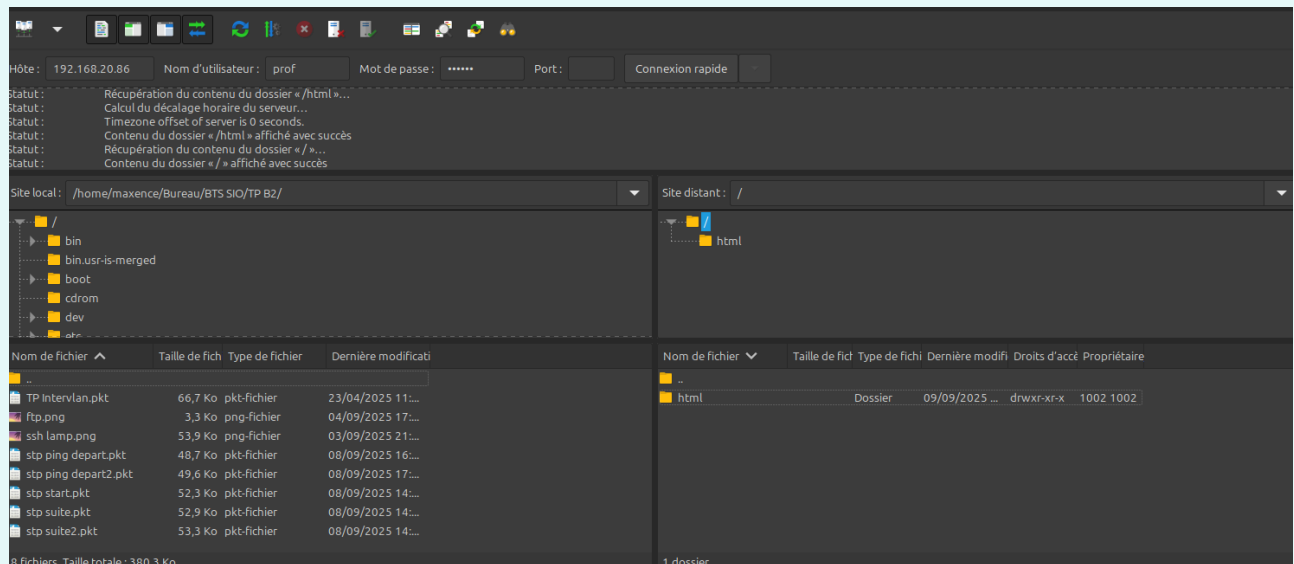
On va maintenant configurer les autorisations pour les deux utilisateurs

```
chmod -R 555 /home/eleve
```

```
usermod -d /var/www prof  
chown -R prof:prof /var/www
```

Vérification :

On vérifie que l'on peut bien se connecter avec les deux utilisateurs en utilisant Fillezilla et que les droits sont bien configurés.



Le prof a bien accès au dossier html mais ne peut pas accéder aux autres dossiers. Cela fonctionne donc bien.

Etape 4 : PHP

On va maintenant créer le fichier php pour vérifier que le serveur est fonctionnel puis on va phpmyadmin pour faciliter la gestion de la base de données.

Création fichier php :

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

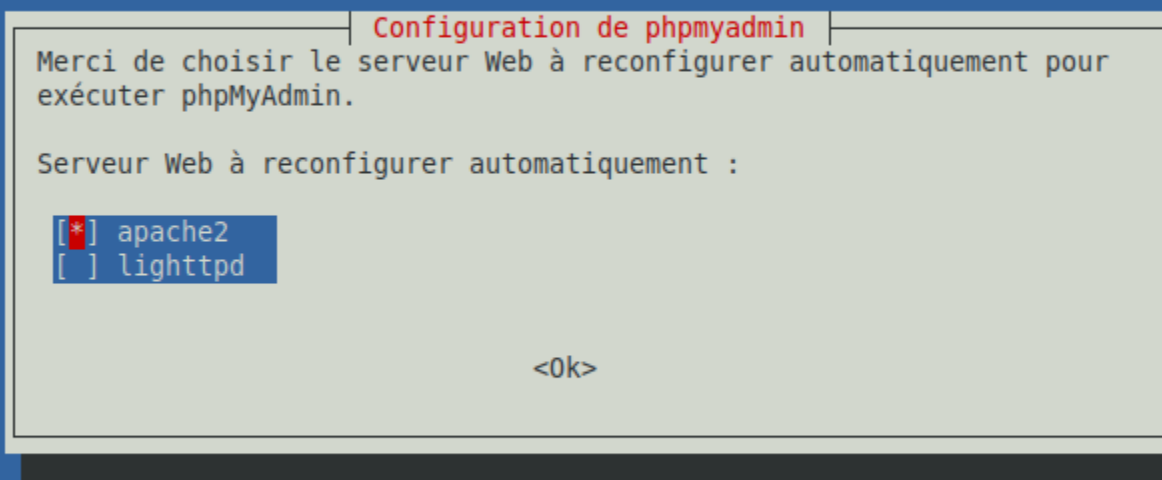
Contenu du fichier :

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

Installation phpmyadmin :

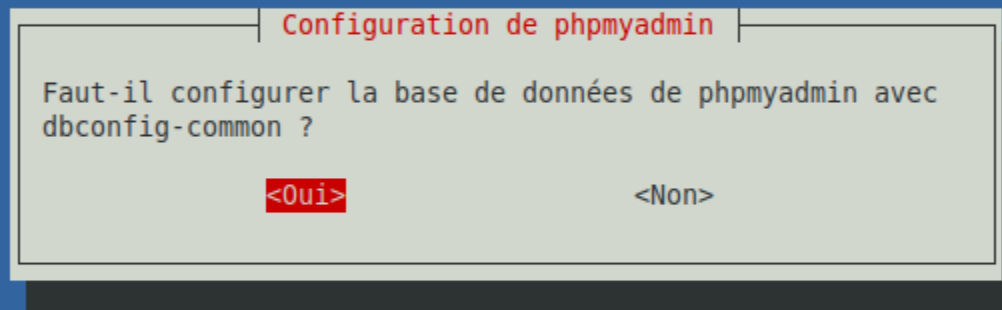
```
apt install phpmyadmin
```

Outil de configuration des paquets



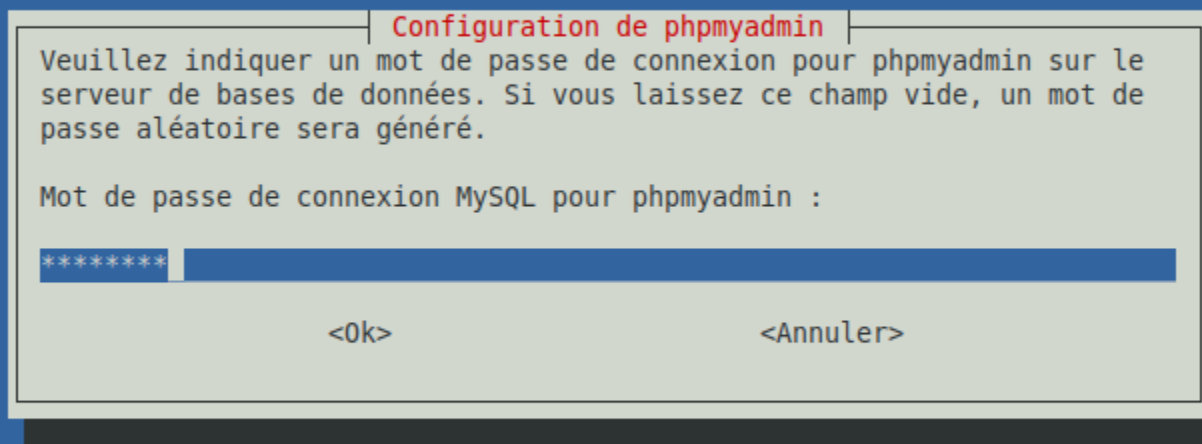
Lors de l'installation on sélectionne le serveur web apache2

Outil de configuration des paquets



On configure la base de données avec dbconfig-common

Outil de configuration des paquets



Et on définit le mot de passe pour phpmyadmin.

Etape 5 : Pages HTML

On va maintenant créer 2 pages html avec la seconde page accessible depuis la première page avec un lien.

Création page index.html :

```
nano /var/www/html/index.html
```

Contenu de la page :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Page d'accueil</title>
</head>
<body>
  <h1>Bienvenue sur mon site</h1>
  <p>Mon site web.</p>

  <nav>
    <ul>
      <li><a href="index.html">Accueil</a></li>
      <li><a href="page1.html">Page1</a></li>
      <li><a href="info.php">Info PHP</a></li>
      <li><a href="/phpmyadmin">phpMyAdmin</a></li>
    </ul>
  </nav>
</body>
</html>
```

Création page1.html :

```
nano /var/www/html/page1.html
```

Contenu de la page :


```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Page Sécurisée</title>
</head>
<body>
  <h1>Page Sécurisée</h1>
  <p>Cette page est protégée par mot de passe.</p>
  <p>Contenu confidentiel réservé aux utilisateurs autorisés.</p>

  <nav>
    <ul>
      <li><a href="index.html">Retour à l'accueil</a></li>
    </ul>
  </nav>

</body>
</html>
```

Etape 6 : Pages HTML Sécurisée

On veut maintenant sécurisé l'accès à la page1.html en demandant un mot de passe lors de chaque connexion. Pour cela on va utiliser .htaccess

Création fichier .htaccess

```
sudo nano /var/www/html/.htaccess
```

Contenu :

```
AuthType Basic
AuthName "Accès Restreint"
AuthUserFile /etc/apache2/.htpasswd
<Files "page1.html">
    Require valid-user
</Files>
```

Création fichier mot de passe :

```
htpasswd -c /etc/apache2/.htpasswd prof
```

htpasswd est un utilitaire pour créer et gérer les mots de passe pour l'authentification HTTP. Avec cette commande on crée le fichier .htpasswd avec l'utilisateur **prof** et son mot de passe hashé.

Activation module apache pour authentification :

On va maintenant activer le module auth_basic d'Apache.

```
a2enmod authn_core auth_basic authn_file
```

Configuration Apache2 :

```
nano /etc/apache2/apache2.conf
```

Contenu que l'on rajoute :

```
<Directory /var/www/>  
  Options Indexes FollowSymLinks  
  AllowOverride All  
  Require all granted  
</Directory>
```

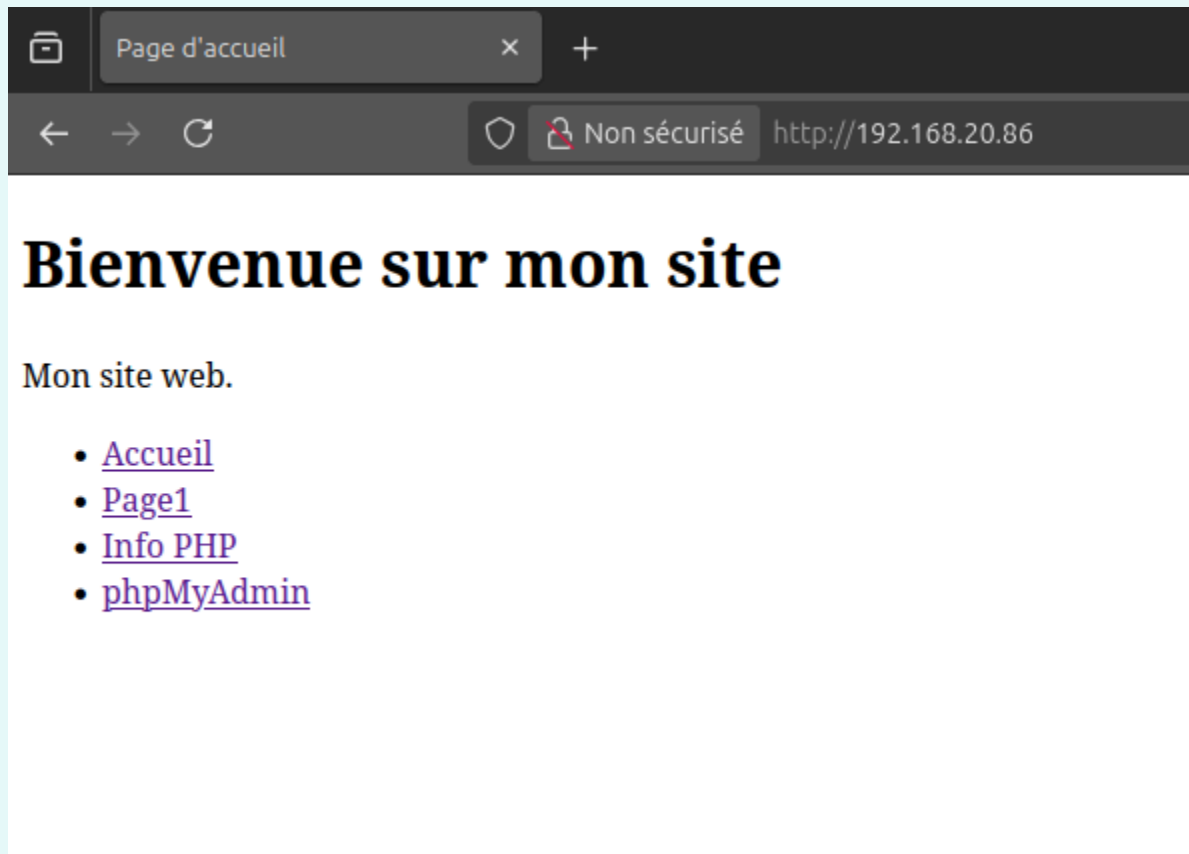
On redémarre Apache2 :

```
systemctl restart apache2
```

🔥 Vérification :

On vérifie maintenant que les deux pages html sont hébergé sur le serveur et que pour se connecter à la page1.html une authentification est demandé.

Page index.html :



Demande de mot de passe :

 http://192.168.20.86

Ce site vous demande de vous connecter.

Nom d'utilisateur

Mot de passe

Page page1.html :

  Non sécurisé http://192.168.20.86/page1.html

Page Sécurisée

Cette page est protégée par mot de passe.

Contenu confidentiel réservé aux utilisateurs autorisés.

- [Retour à l'accueil](#)