TP 1 SERVEURS

Installation ssh, htop, midnight commander (mc), lamp

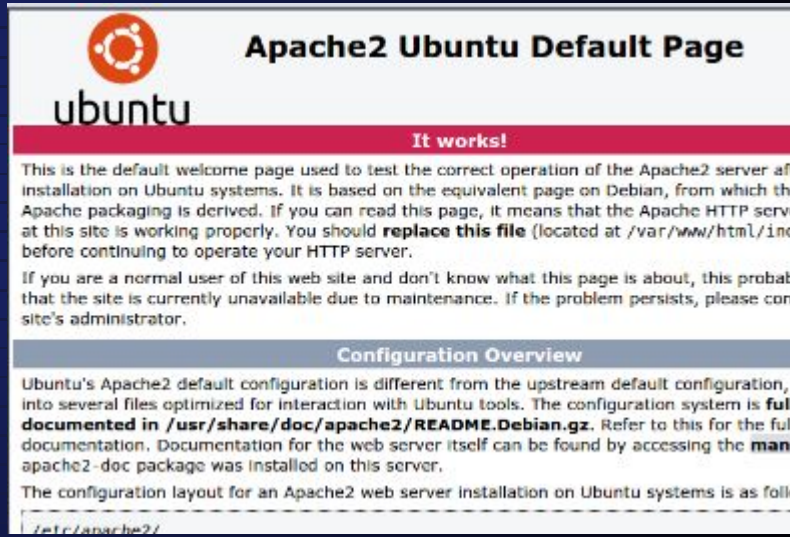
SSH: apt-get install openssh-server

htop: apt-get install htop (htop est un moniteur système pour les systèmes d'exploitation)

mc: apt-get install mc (MC est un gestionnaire de fichiers en mode texte qui affiche les fichiers présent par rapport à leur emplacement sur le disque)

apache2: apt-get install apache2

Et ainsi on aura la page apache2 html qui apparaîtra par défaut



installation ftp

Ici **on installe vsftpd** et on ajoute les user : **élève** qui ne pourra que lire et **prof** qui pourra tout faire et sera redirigé sur var/www :

apt install vsftpd

sudo adduser élève

sudo chmod 555/home/eleve

sudo adduser prof

on change son répertoire de connexion pour qu'il pointe directement vers /var/www :

sudo usermod -d /var/www prof

On donne les permissions nécessaires à cet utilisateur pour gérer le contenu de /var/www **:**

sudo chown prof:prof /var/www et sudo chmod -R 755 /var/www

puis on redémarre le service vsftpd

pour faire en sorte que l'utilisateur prof soit **bloquer dans /var/www** il faut aller dans **vsftpd.conf** et vérifier que la commande soit comme ca **chroot_local_user=YES** et **allow_writeable_chroot=YES** :

```
allow_writeable_chroot=yes
chroot_local_user=YES
```

sudo nano /etc/ssh/sshd_config et on va mettre **ChrootDirectory /var/www** qui empêchera d'accéder à d'autres répertoires du système.

```
Match User prof
        ChrootDirectory /var/www
```

installation ftp

On va ajouter les paramètres suivant dans le fichier **nano /etc/vsftpd.conf :**

pour activer les utilisateurs locaux pours qu'ils puissent se connecter.**local_enable=YES**

pour permettre l'écriture aux utilisateurs ayant les droits (prof) **write_enable=YES**

pour empêcher les utilisateurs de se déplacer hors de leur répertoire personnel **chroot_local_user=YES**

Pour activer le mode passif pour les connexions FTP :

pasv_enable=YES

pasv_min_port=10000

pasv_max_port=10100

Pour définir l'utilisateur (utilisateur prof)

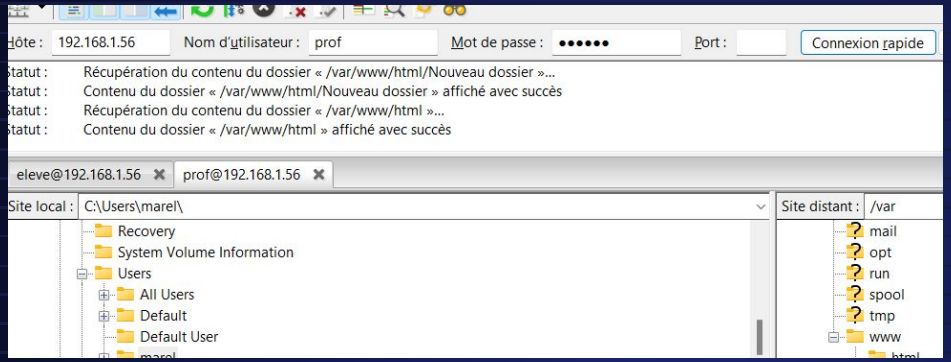
user_sub_token=$USER

local_root=/var/www

```
pasv_enable=YES
pasv_min_port=10000
pasv_max_port=10100
user_sub_token=$USER
local_root=/var/www/prof
write_enable=YES
chroot_local_user=YES
```

Se connecter

Avec filezilla client prof est redirigé vers var mais il peut descendre dans l'arborescence même après avoir fait les config cependant il peut écrire , supprimer, télécharger ...



On peut voir que l'élève ne peut pas écrire mais que lire

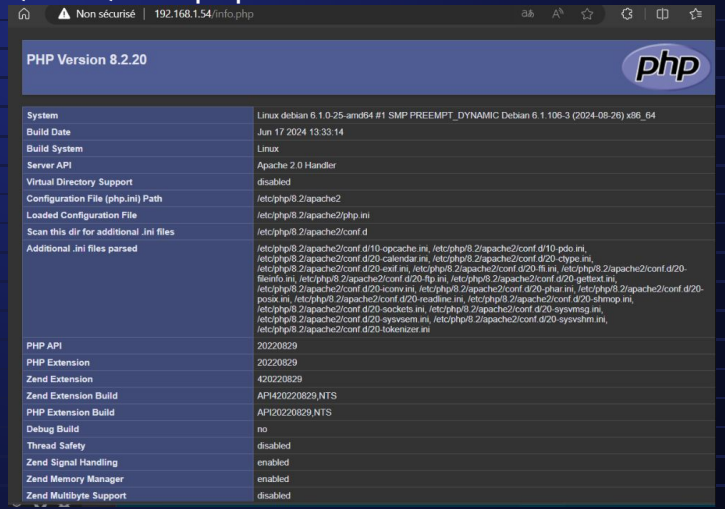
Creer info.php

Ca permet de vérifier que notre serveur est totalement opérationnel

D'abord faut l'installer avec ceci : **sudo apt install php libapache2-mod-php**

Et Créer le fichier dans nano /var/www/html/info.php

ajoutez ceci : <?php phpinfo(); ?>

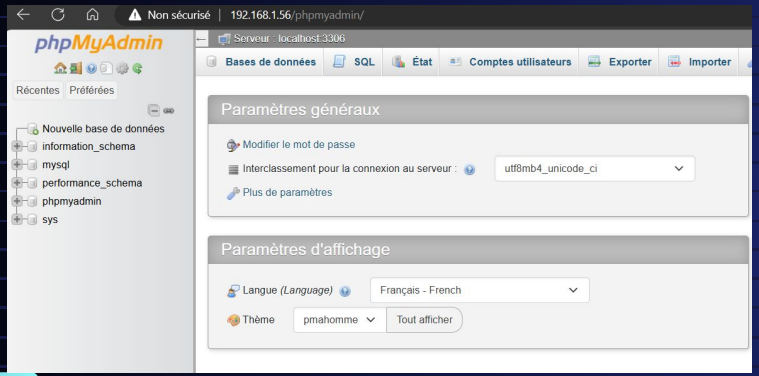


System	Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64
Build Date	Jun 17 2024 13:33:14
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829,NTS
PHP Extension Build	API20220829,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled

PHPMYADMIN

Premièrement j'ai installé le paquet mariaDB et après j'ai installé le paquet de phpmyadmin on met les mdp et oui pour dbconfig-common

et en mettant ip/phpmyadmin on met l'utilisateur et le mdp et voilà on arrive sur phpmyadmin



Créer page index.html et page1.html

index.html

```html
<!DOCTYPE html>
<html lang="fr">
<head>
                <meta charset="UTF-8">
                <meta name="viewport" content="width=device-width,initial-scale=1.0">
        <title>Index</title>
</head>
<body>
<h1>Page d'accueil</h1>
<p>Bienvenue sur la page d'accueil</p>
<p><a href="page1.html">page 2</a></p>
</body>
</html>
```

page1.html

```html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page 1</title>
</head>
<body>
    <a href="index.html"><h1>Page d'accueil Bonjour</h1></a>
</body>
</html>
```

ce qui donne ceci pour index.html

et ceci pour la page 1

Ces étapes vont **restreindre** l'accès à page1.html en demandant **un nom d'utilisateur et un mot de passe à chaque connexion**. Le fichier .htaccess contrôle l'accès et le fichier .htpasswd stocke les identifiants chiffrés.

1) Activer auth_basic

Si ce n'est pas déjà fait vous devez activer le module d'authentification **auth_basic** sur Apache.

sudo a2enmod auth_basic

sudo systemctl restart apache2

2) Créer un fichier .htpasswd pour stocker les identifiants

L'authentification basique repose sur un fichier .htpasswd qui contient les noms d'utilisateur et les mots de passe chiffrés. Pour créer ce fichier installez : **sudo apt-get install apache2-utils**

Créez un fichier .htpasswd dans un répertoire sécurisé (par exemple dans /var/www/ ou un autre répertoire que vous choisissez) et ajoutez un utilisateur. Par exemple ici pour créer un utilisateur prof

sudo htpasswd -c /var/www/.htpasswd prof

Vous serez invité à entrer et confirmer le mot de passe de l'utilisateur prof. Le fichier .htpasswd sera créé avec les informations d'utilisateur chiffrées.

3) Créer un fichier .htaccess pour protéger page1.html

Accédez au répertoire où se trouve votre fichier page1.html. Créez ou éditez un fichier .htaccess dans le répertoire /var/www/html

sudo nano /var/www/html/.htaccess

Ajoutez ceci :

<Files "page1.html">

AuthType Basic (Indique qu'on utilise une authentification basique.)

AuthName "Accès restreint" (C'est le message affiché lors de la demande de mot de passe.)

AuthUserFile /var/www/.htpasswd (Chemin vers le fichier .htpasswd contenant les identifiants.)

Require valid-user(Cela demande un utilisateur valide pour accéder à la page.)

</Files>