

TP Serveurs

B2



Installation de SSH

- Au début, on tape les commandes **sudo apt upgrade** puis **sudo apt update**, pour mettre à jour notre machine.
- En ligne de commande, on passe en super utilisateur avec **su**. Ensuite on tape la commande **apt install ssh**.

```
root@debian:/home# apt install ssh_
```

- On vérifie que notre ssh est bien installé avec **which ssh**

```
root@debian:/home# which ssh  
/usr/bin/ssh  
root@debian:/home# _
```

Installation du paquet htop

On tape la commande **apt install htop**

```
root@debian:/home# apt install htop
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

Pour vérifier que notre paquet htop est bien installé, on tape **htop** dans le terminal.

On voit que cela nous affiche les taches en cours, l'utilisation du CPU, de la mémoire et bien d'autres.

```
CPU[]
Mem[|||||||||||||||||||||||||]
Swp[

0.7%] Tasks: 17, 1 thr, 55 kthr; 1 running
118M/961M] Load average: 0.01 0.04 0.03
0K/975M] Uptime: 00:08:37

Main
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
924 root 20 0 8040 4120 3156 R 0.7 0.4 0:00.21 htop
1 root 20 0 100M 13264 9152 S 0.0 1.3 0:01.75 /sbin/init
209 root 20 0 49400 15204 12084 S 0.0 1.5 0:00.14 /lib/systemd/systemd-journald
231 root 20 0 26184 6004 4652 S 0.0 0.6 0:00.28 /lib/systemd/systemd-udev
257 systemd-ti 20 0 90080 6636 5756 S 0.0 0.7 0:00.09 /lib/systemd/systemd-timesyncd
397 root 20 0 5868 3600 2768 S 0.0 0.4 0:00.01 dhclient -4 -v -i -pf /run/dhclient.enp0s3.pid -lf /var/lib/dhcp/dhclient.enp0s3.leases -I -d
466 systemd-ti 20 0 90080 6636 5756 S 0.0 0.7 0:00.00 /lib/systemd/systemd-timesyncd
468 root 20 0 6608 2836 2584 S 0.0 0.3 0:00.00 /usr/sbin/cron -f
470 messagebus 20 0 9136 4948 4372 S 0.0 0.5 0:00.14 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --
472 root 20 0 25200 7664 6676 S 0.0 0.8 0:00.10 /lib/systemd/systemd-logind
473 root 20 0 16532 5852 4988 S 0.0 0.6 0:00.01 /sbin/wpa_supplicant -u -s -O DIR=/run/wpa_supplicant GROUP=netdev
475 root 20 0 6112 4092 3560 S 0.0 0.4 0:00.11 /bin/login -p --
505 user 20 0 18872 10516 8952 S 0.0 1.1 0:00.09 /lib/systemd/systemd --user
506 user 20 0 100M 2952 0 S 0.0 0.3 0:00.00 (sd-pam)
512 user 20 0 7972 4640 3292 S 0.0 0.5 0:00.02 -bash
543 root 20 0 9004 3808 3404 S 0.0 0.4 0:00.08 su
546 root 20 0 7196 3952 3412 S 0.0 0.4 0:00.02 bash
808 root 20 0 15432 8780 7540 S 0.0 0.9 0:00.01 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Installation du paquet midnight

On tape la commande **apt install mc**

Puis on vérifie l'installation de notre paquet avec la commande mc

La commande mc saisie précédemment nous renvoie cela

```
root@debian:/home# apt install mc
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```



The screenshot shows the Midnight Commander (mc) interface with two panes. The left pane shows the directory listing for /home, and the right pane shows the directory listing for /user. Both panes display a table with columns for Nom, Taille, and Date de Modifi. The /home pane shows a directory listing for /home with entries for .n, .., and /user. The /user pane shows a directory listing for /user with entries for .n, .., and /user. The interface includes a menu bar at the top with options like Fichier, Commande, Options, and Droite. At the bottom, there is a status bar with the text 'Astuce: Recherche de fichier: travaillez avec les fichiers trouvés en utilisant la fonction "Mettre en panneau"' and a keyboard shortcut menu.

Nom	Taille	Date de Modifi
.n	REP-SUP	4 sept. 09:17
..		
/user	4096	4 sept. 09:21

Nom	Taille	Date de Modifi
.n	REP-SUP	4 sept. 09:17
..		
/user	4096	4 sept. 09:21

Installation de LAMP



Installation de notre serveur Apache

Avec la commande **apt install apache2** .

```
root@debian:/home# apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

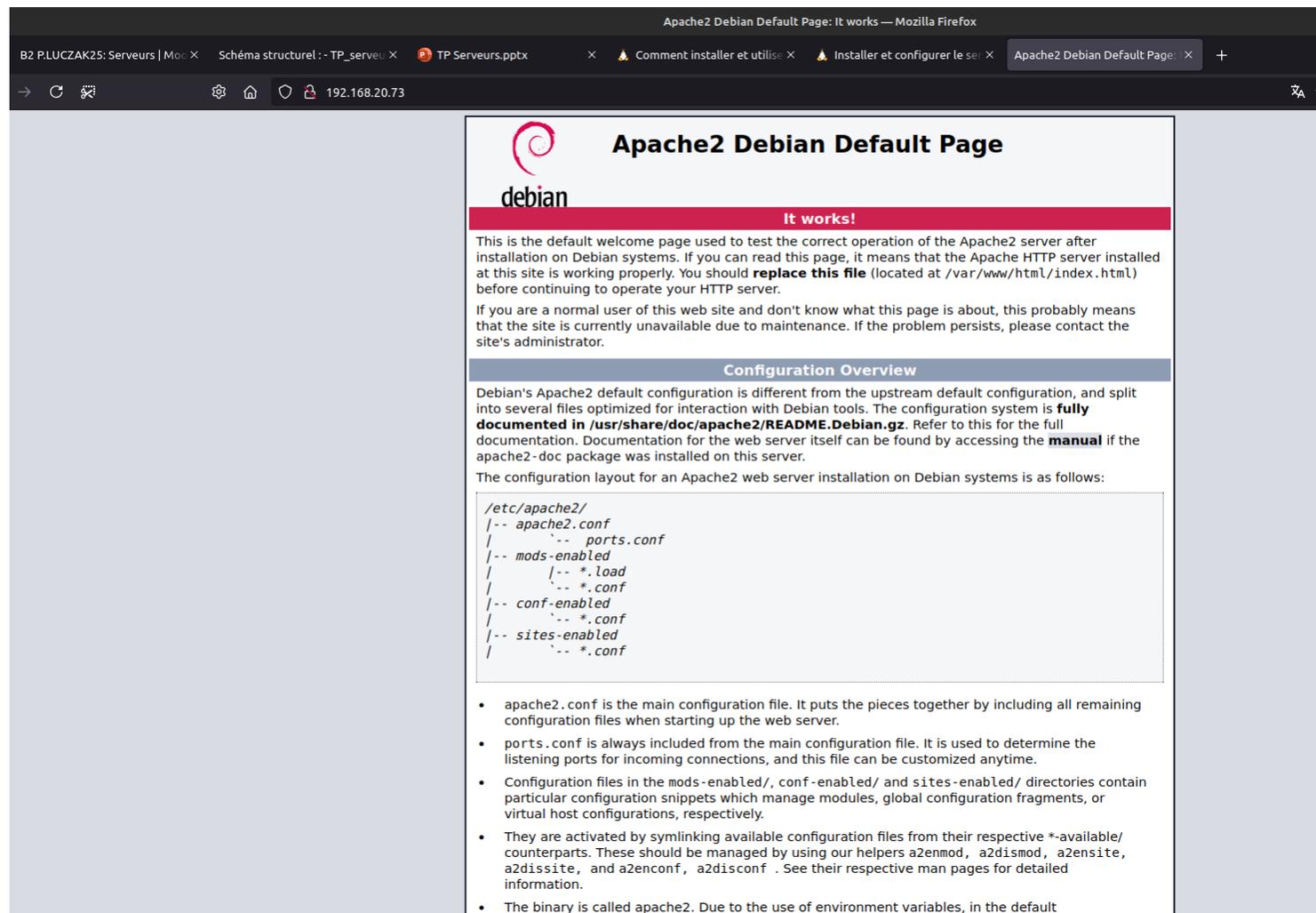
Ensuite nous allons démarrer notre serveur apache avec les commandes suivantes

```
root@debian12:~# systemctl start apache2
root@debian12:~# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@debian12:~# systemctl status apache2
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Tue 2024-09-10 08:44:33 CEST; 50s ago
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 1422 (apache2)
    Tasks: 55 (limit: 2306)
  Memory: 9.1M
    CPU: 39ms
  CGroup: /system.slice/apache2.service
          └─1422 /usr/sbin/apache2 -k start
             └─1424 /usr/sbin/apache2 -k start
                └─1425 /usr/sbin/apache2 -k start

sept. 10 08:44:33 debian12 systemd[1]: Starting apache2.service - The Apache HTTP Server...
sept. 10 08:44:33 debian12 apachectl[1421]: AH00558: apache2: Could not reliably determine the server's
sept. 10 08:44:33 debian12 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

Vérification Apache

Pour vérifier que notre serveur Apache est bien accessible, nous allons récupérer l'IP de notre machine avec la commande `ip a` puis on tape l'adresse dans notre navigateur.



Apache2 Debian Default Page: It works — Mozilla Firefox

B2 P.LUCZAK25: Serveurs | Moc x Schéma structurel :- TP_serveu x TP Serveurs.pptx x Comment installer et utilis x Installer et configurer le se x Apache2 Debian Default Page: x +

→ ↻ 🔍 🏠 📄 192.168.20.73 🌐 ⭐

Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default

Installation des modules PHP

Pour cela nous allons saisir la commande suivante afin d'installer tous les modules php nécessaires au fonctionnement de notre serveur.

```
root@debian:/home# apt-get install -y php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait
```

Installation des modules SQL

Nous allons saisir la commande suivante pour installer les modules SQL :

Ensuite nous allons exécuter le script suivant afin de sécuriser notre base de données.

```
root@debian:/home# apt-get install -y mariadb-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

```
root@debian:/home# mariadb-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
haven't set the root password yet, you should just press enter here.

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

Installation du FTP

Nous allons donc installer notre client FTP avec la commande suivante :

```
root@debian:/# apt install proftpd
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

Ensuite nous allons démarrer notre FTP avec la commande **systemctl start proftpd** puis **systemctl enable proftpd** afin qu'il soit actif à chaque démarrage de la machine.

```
root@debian:/# systemctl status proftpd
• proftpd.service - ProFTPD FTP Server
   Loaded: loaded (/lib/systemd/system/proftpd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-07 18:06:57 CEST; 1min 53s ago
     Docs: man:proftpd(8)
  Main PID: 1142 (proftpd)
    Tasks: 1 (limit: 1108)
   Memory: 1.9M
      CPU: 52ms
   CGroup: /system.slice/proftpd.service
           └─1142 "proftpd: (accepting connections)"

sept. 07 18:06:57 debian systemd[1]: Starting proftpd.service - ProFTPD FTP Server..
sept. 07 18:06:57 debian proftpd[1140]: Checking syntax of configuration file
sept. 07 18:06:57 debian systemd[1]: proftpd.service: Can't open PID file /run/proftpd
sept. 07 18:06:57 debian systemd[1]: Started proftpd.service - ProFTPD FTP Server.
```

Création des utilisateurs et gestion des droits

On crée nos utilisateurs avec la commande **useradd** **[nom du user]**

Puis on ajoute nos utilisateurs dans leurs groupes respectifs avec la commande suivante

Nous allons nous connecter sur chaque user pour fixer les droits

Les droits pour le user prof sont lecture et écriture donc nous utiliserons un **chmod -R 770**

Les droits du user élève sont lecture uniquement alors la commande sera la suivante **chmod -R 775**

```
root@debian:~# usermod -aG profs prof
root@debian:~# usermod -aG eleve eleve
root@debian:~# _
```

```
root@debian12:~# chown -R prof: /var/www
root@debian12:~# chmod -R 770 /var/www
root@debian12:~#
```

```
root@debian12:/var/www# chmod -R 775 /var/www
root@debian12:/var/www# _
```

Racine dossier Prof

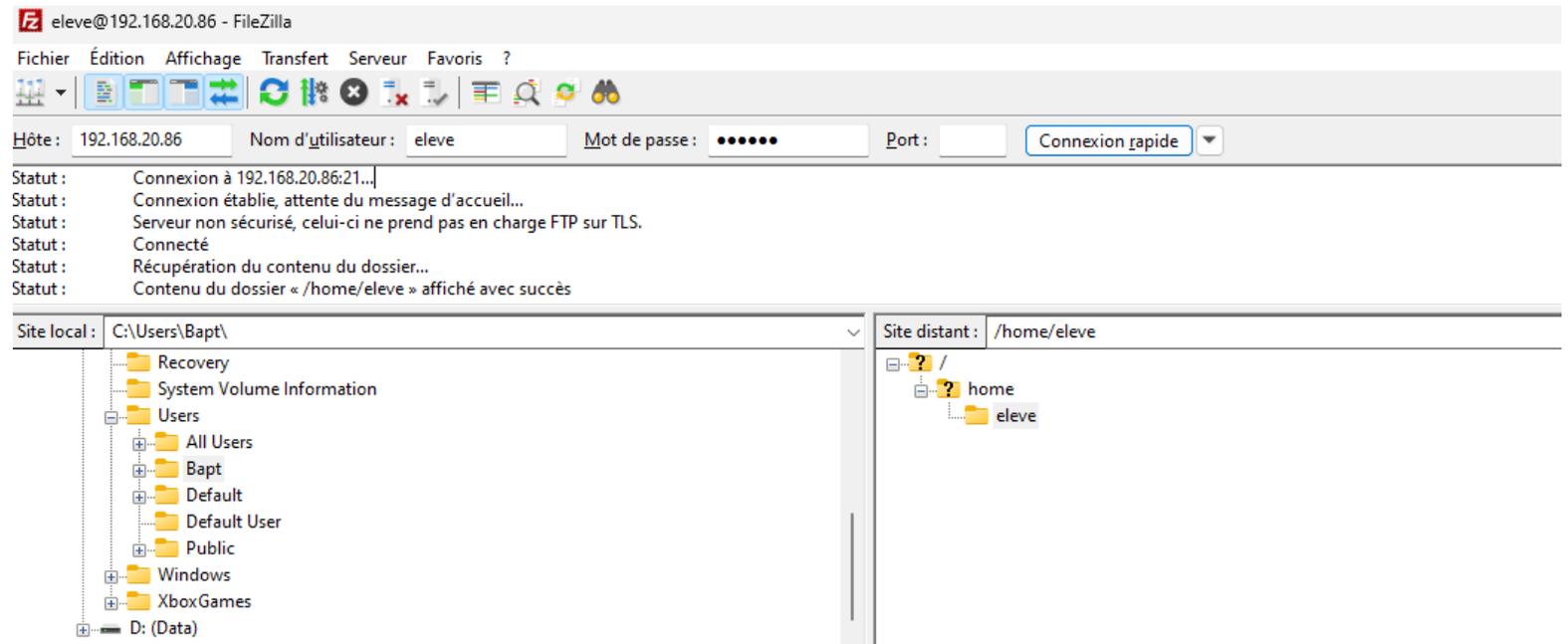
On se rend dans `/etc/passwd` avec nano, puis on modifie la racine de l'utilisateur prof par **`/var/www`**

```
ftp:x:104:65534::/srv/ftp:/usr/sbin/nologin
prof:x:1001:1001:,,,:/home/prof:/var/www
eleve:x:1002:1002:,,,:/home/eleve:/bin/bash
```

Puis on vérifie cela sur filezilla en se connectant en FTP.

Vérification

Sur filezilla, nous voyons donc que nous pouvons avoir accès à notre FTP



Fichier phpinfo.php

Dans le répertoire `/var/www`, on supprime le fichier `index.html` avec la commande `rm -r index.html`.

Création du fichier `phpinfo.php` avec `touch phpinfo`

Puis nous allons écrire dedans avec la commande `nano phpinfo.php`

```
root@debian12:/var/www/html# ls -la
.  ..  phpinfo.php
root@debian12:/var/www/html#
```

```
GNU nano 7.2
<? php
phpinfo();
?>
```



PHP Version 8.2.20	
System	Linux debian12 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-mysqld.ini, /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/15-xml.ini, /etc/php/8.2/apache2/conf.d/20-bz2.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-curl.ini, /etc/php/8.2/apache2/conf.d/20-dom.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-gd.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-mbstring.ini, /etc/php/8.2/apache2/conf.d/20-mcrypt.ini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.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-simplexml.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, /etc/php/8.2/apache2/conf.d/20-xmlreader.ini, /etc/php/8.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.2/apache2/conf.d/20-zip.ini, /etc/php/8.2/apache2/conf.d/20-zlib.ini

Installation des modules PHP

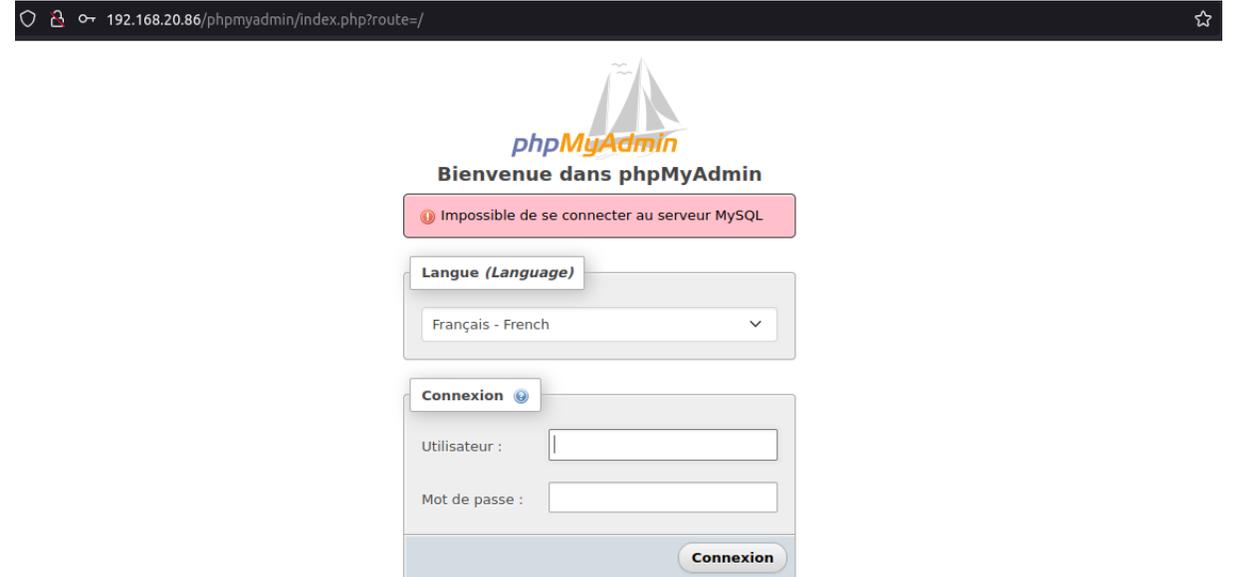
Ensuite nous allons donc installer différents modules php afin de pouvoir administrer comme nous le souhaitons notre serveur.

```
root@debian12:/# cd ..  
root@debian12:/# apt install phpmyadmin apache2 php-zip php-gd php-json php-curl libapache2-mod-php  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait
```

Accès à phpmyadmin

Une fois les modules installés, on tape notre **ip/phpmyadmin/index.html**

Impossible de se connecter puisqu'il faut créer un user dans notre base de données **mariadb**



Création d'utilisateurs

On tape **mariadb** dans le shell pour accéder à notre base de données mariadb, ensuite on rentre la requête suivante pour créer notre user.

```
root@debian12:~# mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

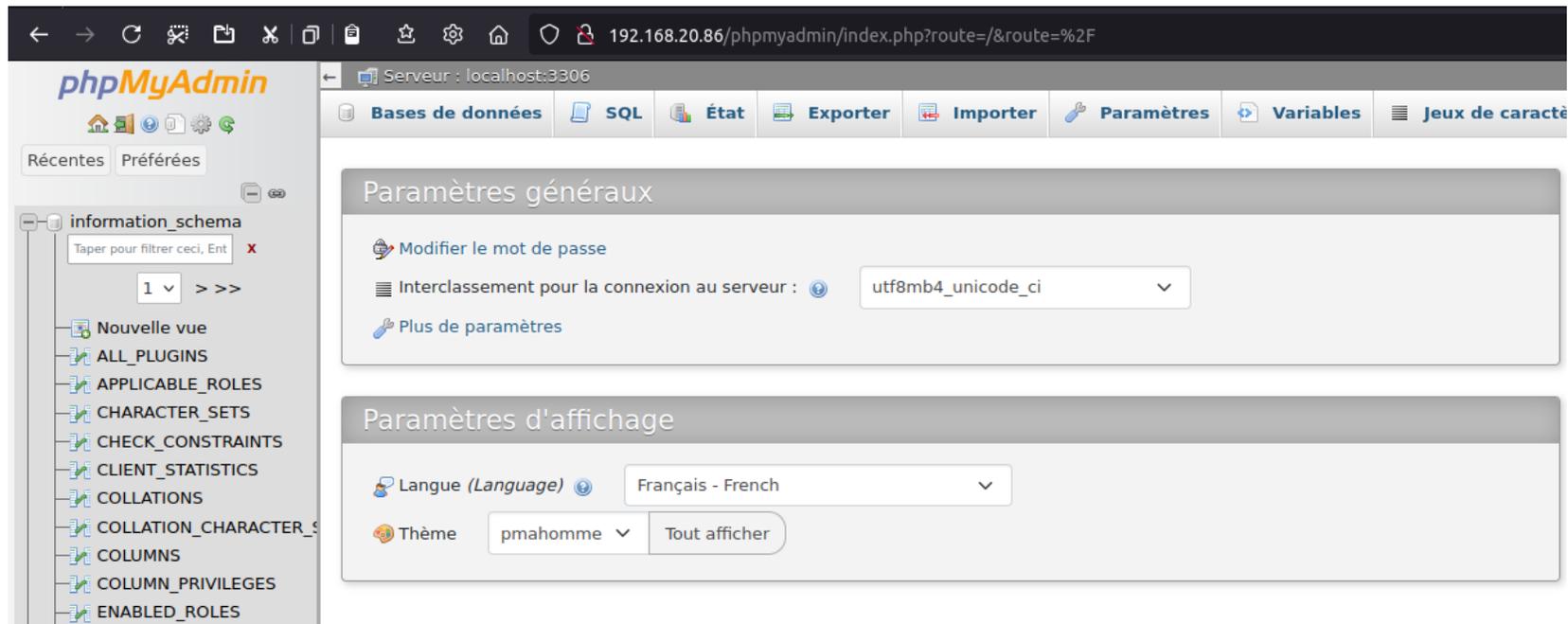
MariaDB [(none)]> CREATE USER 'bapt'@'%' IDENTIFIED BY 'bapt13';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> _
```

Vérification sur phpmyadmin

On se rend sur notre page phpmyadmin consultée précédemment.

On rentre le login/mdp créer avec la requête entrée juste au-dessus et nous avons donc accès.



Création des pages html

Création des pages html dans le répertoire `/var/www` avec la commande **touch page1.html** et **touch index.html**.

```
GNU nano 7.2 index.html
<!DOCTYPE html>
<html lang='fr'>
<head>
  <title>Index.html</title>
</head>
<body>
  <header>
    <h1>Bonjour c'est la page html</h1>
  </header>
  <a href="page1.html">Aller à la page suivante</a>
</body>
</html>
```

```
GNU nano 7.2
<!DOCTYPE html>
<html lang="fr">
<title>page1.html</title>
<body>
<h1>Bienvenue sur la page 1</h1>
</body>
</html>
```

Sécurisation de la page html

Pour sécuriser notre page html, nous allons activer le module d'authentification d'apache2.

Avec la commande suivante **a2enmod authn_core**

```
root@debian12:/var/www/html# cd  
root@debian12:~# a2enmod authn_core  
Module authn_core already enabled  
root@debian12:~# _
```

Création d'un fichier **.htpasswd** *nom d'utilisateur* dans le répertoire /var/www

```
root@debian12:~# cd /etc/apache2  
root@debian12:/etc/apache2# ls -a  
. .. apache2.conf conf-available conf-enabled envvars magic mods-available  
root@debian12:/etc/apache2# touch .htpasswd bapt  
root@debian12:/etc/apache2# ls -a  
. .. apache2.conf bapt conf-available conf-enabled envvars .htpasswd magic
```

Sécurisation de la page html

On se rend dans notre répertoire **`/var/www`**

Dans lequel nous allons créer notre fichier `.htaccess` avec la commande **`touch .htaccess`**

Puis on saisit les informations suivantes :

```
GNU nano 7.2
<Files "page1.html">
  AuthType Basic
  AuthName "Espace Protégé"
  AuthUserFile /etc/apache2/.htpasswd
  Require valid-user
</Files
```

Sécurisation de la page html

On se rend dans notre répertoire avec nano **/etc/apache2/.htaccess**, puis on rentre les commandes ci-dessus.

Elles nous permettent de mettre la couche de sécurité pour notre page.

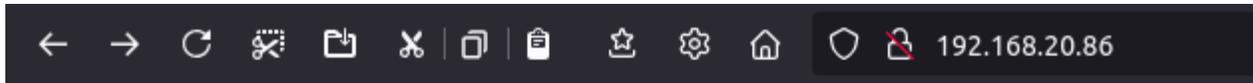
Ensuite on se rend avec nano dans **/etc/apache2/sites-available/000-default.conf**, puis on y ajoute les informations suivantes afin que cela soit dirigé correctement.

```
GNU nano 7.2
<Files "page1.html">
    AuthType Basic
    AuthName "Espace Protégé"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Files
```

```
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    <Directory /var/www/html>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Vérification



Bonjour c'est la page html

[Aller à la page suivante](#)

