

# Documentation Technique MISSION 3

## AP2 SISR : Contexte M2L

### Plan:

#### 1) Installation d'un environnement Web

- a. Installation du serveur Web ApacheB
- b. Installation de PHP

#### 2) Installation de la base de données SQL MariaDB

- a. Installation de MariaDB
- b. Test du serveur MySQL
- c. Accès à distance

#### 3) Connexion à MariaDB

#### 4) Gestion des bases de données SQL avec MariaDB

- a. Création d'une base de données à partir d'un script SQL
- b. Sauvegarde d'une base de données dans un fichier de script SQL

#### 5) Installation et configuration de PhpMyAdmin

#### 6) Création d'un serveur DNS

#### 7) Installation et configuration d'un serveur FTP

- a. Installation de vsftpd
- b. Création de l'utilisateur FTP
- c. Sécuriser le serveur FTP
- d. Configuration de vsftpd
- e. Connexions cryptées

## **8) Installation et configuration d'une machine Linux pour le test**

### **9) Wordpress**

- a. Création d'un host virtuel pour le site wordpress
- b. Création d'un nom de domaine pour le site wordpress
- c. Problème rencontré
- d. Connexion à la base de données wordpress
- e. Présentation de notre site wordpress  
(<http://wordpress.equipe8.m2l.local/>)

Create: LXC Container

General Template Disks CPU Memory Network DNS **Confirm**

Key ↑	Value
cores	1
features	nesting=1
hostname	ServeurDebian
memory	512
nameserver	10.187.88.5
net0	name=eth0,bridge=vibr134,firewall=1,ip6=dhcp,ip=dhcp
nodename	siohyp1
ostemplate	NFS-NAS:vztmpl/debian-11-standard_11.7-1_amd64.tar.zst
pool	SIO2023SISREquipe08
rootfs	NFS-NAS:8
searchdomain	0870019y.lan
ssh-public-keys	
start	1
swap	512

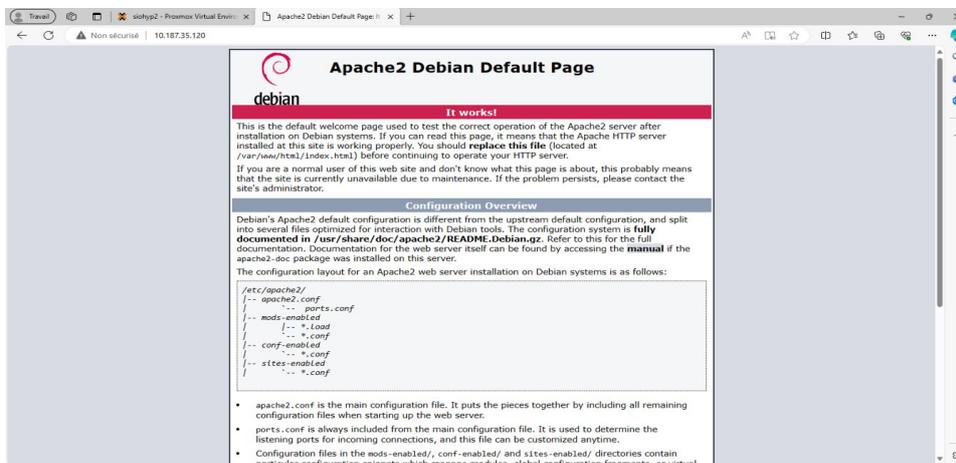
Start after created

Advanced  **Back** **Finish**

## 1. Installation d'un environnement Web

### A. Installation du serveur Web

Pour installer le serveur Web, nous avons installé le paquetage apache2 avec la commande suivante : `sudo apt install apache2 -y`



## B. Installation de PHP

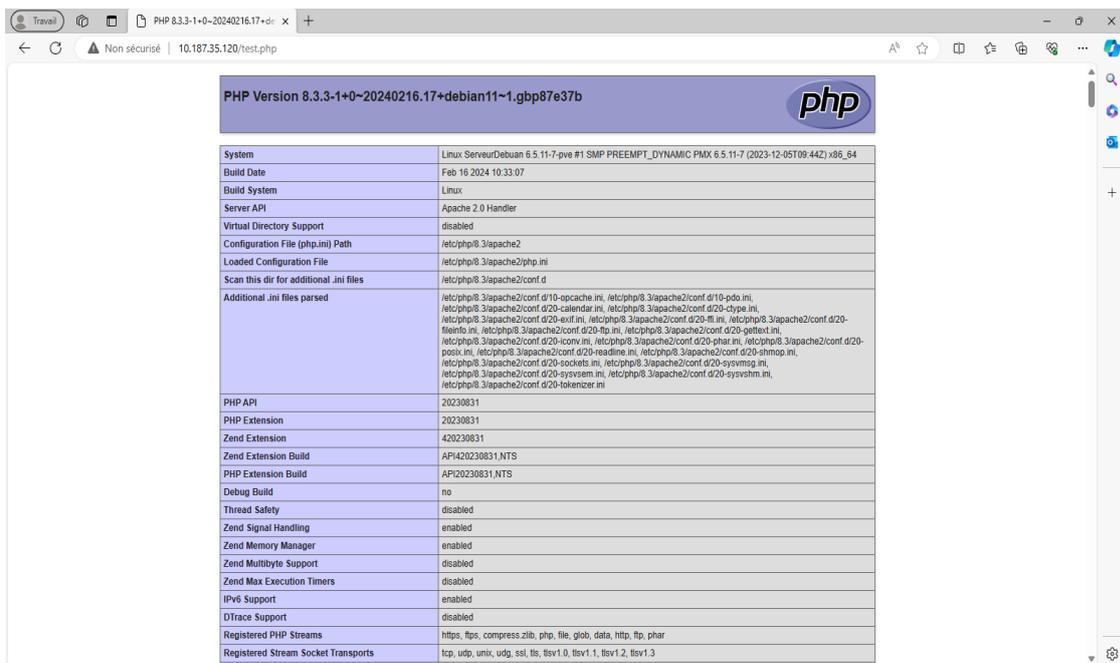
Installation des prérequis avec l'installation de paquets issus de dépôts personnalisés, nous avons utilisé la commande suivante : **sudo apt -y install apt-transport-https lsb-release ca-certificates curl**

L'import des clés de signature des paquets sury.org/php est fait avec la commande suivante : **sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg**

On a ajout du dépôt dans les sources d'installation avec la commande suivante : **sudo bash -c 'echo «deb https://packages.sury.org/php/ \$(lsb\_release -sc) main» > /etc/apt/sources.list.d/php.list'**

On a ensuite fait la mise à jour de la liste des paquets et celle de l'installation si besoin avec la commande suivante : **sudo apt update && sudo apt -y upgrade**

Nous avons installé le packaging PHP avec la commande suivante : **sudo apt -y install php**



PHP Version 8.3.3-1+0-20240216.17+debian11-1.gb87e37b	
System	Linux ServeurDebian 6.5.11-7-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-7 (2023-12-05T09:44Z) x86_64
Build Date	Feb 16 2024 10:33:07
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php8.3/apache2
Loaded Configuration File	/etc/php8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php8.3/apache2/conf.d
Additional .ini files parsed	/etc/php8.3/apache2/conf.d/10-opcache.ini, /etc/php8.3/apache2/conf.d/10-pdo.ini, /etc/php8.3/apache2/conf.d/20-calendar.ini, /etc/php8.3/apache2/conf.d/20-ctype.ini, /etc/php8.3/apache2/conf.d/20-exif.ini, /etc/php8.3/apache2/conf.d/20-ffi.ini, /etc/php8.3/apache2/conf.d/20-fileinfo.ini, /etc/php8.3/apache2/conf.d/20-ftp.ini, /etc/php8.3/apache2/conf.d/20-gettext.ini, /etc/php8.3/apache2/conf.d/20-iconv.ini, /etc/php8.3/apache2/conf.d/20-imagick.ini, /etc/php8.3/apache2/conf.d/20-imagick.ini, /etc/php8.3/apache2/conf.d/20-ldap.ini, /etc/php8.3/apache2/conf.d/20-ldap.ini, /etc/php8.3/apache2/conf.d/20-mbstring.ini, /etc/php8.3/apache2/conf.d/20-memcached.ini, /etc/php8.3/apache2/conf.d/20-msgpack.ini, /etc/php8.3/apache2/conf.d/20-redis.ini, /etc/php8.3/apache2/conf.d/20-soap.ini, /etc/php8.3/apache2/conf.d/20-sockets.ini, /etc/php8.3/apache2/conf.d/20-sysvsem.ini, /etc/php8.3/apache2/conf.d/20-sysvshm.ini, /etc/php8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831.NTS
PHP Extension Build	API20230831.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3

## - Installation des paquetages PHP pour MariaDB

Pour que des applications PHP puissent utiliser la base de données MariaDB, nous avons installé les paquetages suivants qui doivent être installés avec la commande suivante : `sudo apt -y install libapache2-mod-php php-mysql`

Pour afficher la version du serveur Apache2 et visualiser les versions des logiciels nous avons utilisé la commande suivante : `sudo apache2 -V`

Pour connaître la version du moteur de scripts PHP nous avons utilisé la commande : `php -v`

## 2. Installation de la base de données SQL MariaDB

### a. Installation du paquetage MariaDB

Nous avons installé le paquetage MariaDB avec la commande suivante : `sudo apt -y install mariadb-server`.

Ensuite, on a créé un compte admin pour l'administration de MariaDB avec comme mot de passe `P@$word1` en utilisant la commande suivante : `sudo mariadb -u root -e "CREATE USER admin@'%'; GRANT ALL PRIVILEGES ON *.* to admin@%' IDENTIFIED BY 'P@$word1' WITH GRANT OPTION; FLUSH PRIVILEGES;"` .

Ce compte admin dispose maintenant de tous les privilèges (ALL PRIVILEGES) sur toutes les bases de données (ON \*.\*). Après la définition nous l'avons activé avec la commande `SQL FLUSH PRIVILEGES`.

### b. Test du serveur MySQL

Pour visualiser le processus mariadb on a utilisé la commande : `ps aux | grep mysqld`

Le processus serveur s'appelle `mysql`

Le serveur SQL MariaDB est en écoute sur le port TCP 3306 mais uniquement en local c'est-à-dire depuis l'adresse IP 127.0.0.1 comme le montre la commande suivante : `ss -nlt`

```

root@ServeurDebian:~# ss -nl
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
LISTEN    0            80          0.0.0.0:3306             0.0.0.0:*
LISTEN    0           4096        0.0.0.0:5355            0.0.0.0:*
LISTEN    0           4096        127.0.0.53%lo:53        0.0.0.0:*
LISTEN    0            100        127.0.0.1:25            0.0.0.0:*
LISTEN    0            511        *:*:80                  *:*
LISTEN    0           4096        *:*:22                  *:*
LISTEN    0           4096        [::]:5355               [::]:*
LISTEN    0            100        [::]:25                 [::]:*

```

### c. Pour permettre l'accès à distance au serveur SQL MariaDB

Par défaut, MariaDB n'écoute et n'autorise que les connexions depuis l'hôte local (adresse IP 127.0.0.1) c'est-à-dire le serveur sur lequel MariaDb est installé. Tout accès distant au serveur est refusé par défaut.

Pour permettre l'accès depuis un autre hôte ou serveur à MariaDB, nous avons modifier la configuration de MariaDB.

On a donc édité le fichier `t/etc/mysql/mariadb.conf.d/50-server.cnf` avec l'éditeur nano et modifier la ligne suivante:

**Bind-address** = **127.0.0.1**

Pour obtenir la ligne suivante :

**Bind-address** = **0.0.0.0**

Ensuite on a enregistré le fichier modifié et on a redémarré le service MariaDb avec la commande : `sudo systemctl restart mariadb`

### 3. Connexion à MariaDB:

Pour se connecter à mariaDB avec le compte root sans mot de passe, on a utilisé la commande suivante : `sudo mariadb -u.`

On visualise les bases de données existantes avec la commande `show databases;`

Pour choisir une base de donnée, on utilise la commande `use <NomBaseDonnée>`

Pour visualiser toutes tables existantes dans la BDD mysql, on utilise la commande `show tables;`

Pour visualiser les colonnes d'une table, on utilise la commande `show columns from <NomTable>;`

Pour visualiser les comptes existants, on utilise la commande : `select user, password from user;`

```
MariaDB [(none)]> select user, password from user;
+-----+-----+
| user | password |
+-----+-----+
| root |          |
| admin | *07843C5608DD48788D41F23F44E511A0886F443B |
+-----+-----+
2 rows in set (0.000 sec)

MariaDB [(none)]> exit
```

Nous avons définis le mot de passe du compte root avec la commande `MariaDB [mysql]> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('P@$word1');` `MariaDB [mysql]> FLUSH PRIVILEGES ;`

## 4. Gestion des bases de données SQL avec MariaDB

### a. Création d'une base de données à partir d'un script SQL

Pour cela, nous avons procédé de la manière suivante:

- Nous avons téléchargé sur le serveur Debian qui héberge la base de données MariaDB, le script SQL de création de la base de donnée.
- On a ouvert une session sur le serveur Debian et accédé au dossier où se trouve le script.
- Ensuite nous avons exécuté la commande suivante pour créer la base de données à partir du script :

Commande : `sudo mariadb -u admin -p < geststages.sql`

**Explication de la commande :**

La commande sudo permet d'avoir une élévation de privilège pour exécuter la commande de lancement du client SQL mariadb ;

— le paramètre -u permet d'indiquer le compte de connexion à utiliser ;

— le paramètre -p permet la demande de saisie du mot de passe qu'il faudra alors indiquer à l'exécution de la commande ;

— le caractère < est un pipe qui permet de lire le contenu du fichier situé à droite (geststages.sql) pour l'injecter dans le client MariaDB situé à gauche. En résumé cela permet de demander au client MariaDB d'exécuter toutes les commandes SQL contenues dans le fichier de script.

### b. Sauvegarde de la base de données dans un fichier de script SQL

Pour sauvegarder la base de données MariaDB dans un script SQL qui contiendra alors les commandes SQL de création de la base, de création des tables et des données, nous avons utilisé la commande suivante :

**COMMANDE :** `sudo mysqldump -u admin -p geststages> sauvegardebdd.sql`

Explication de la commande :

- la commande `sudo` permet d’avoir une élévation de privilège pour exécuter la commande de lancement du client SQL mariadb ;
- le paramètre `-u` permet d’indiquer le compte de connexion à utiliser ;
- le paramètre `-p` permet la demande de saisie du mot de passe qu’il faudra alors indiquer à l’exécution de la commande ;
- le caractère `>` est un pipe qui permet de rediriger le résultat de la commande située à gauche dans le fichier de script situé à droite. En résumé cela permet de demander de sauvegarder toute la base sous forme de commandes SQL dans le fichier de script `savegardebdd.sql`

## 5. Installation et configuration de PhpMyAdmin

- Téléchargement du package phpMyAdmin avec la commande : `wget https://www.phpmyadmin.net/downloads/phpMyadmin-latest-all-languages.tar.gz`

Wget est un programme en ligne de commande de téléchargement de fichiers depuis le Web. Il supporte les protocoles, HTTP, HTTPS et FTP

- Décompression de l’archive avec la commande suivante : `sudo tar xvf phpMyAdmin-latest-all-languages.tar.gz`
- Création du dossier suivant sous la racine du server web apache 2 avec la commande ci-contre : `mkdir /var/www/html/phpMyadmin`
- Déplacement du fichier décompressé vers le dossier créé ci-dessus avec la commande : `mv phpMyAdmin-*-all-languages/ /var/www/html/phpmyadmin`
- Création d’un fichier de configuration en copiant celui déjà présent dans le dossier phpMyAdmin avec la commande ci-contre : `sudo cp phpmyadmin/config.sample.inc.php phpmyadmin/config.inc.php`
- Création d’un dossier temporaire nommé “tmp” pour le stockage, s’il n’existe pas avec la commande suivante : `sudo mkdir /var/www/html/phpmyadmin/tmp`
- Modification de la configuration du phpMyAdmin avec la commande : `sudo nano /var/html/phpmyadmin/config.inc.php`

Modification du dossier temporaire à la ligne suivante indiquée :

```
$cfg['TempDir'] = '/var/www/html/phpmyadmin/tmp';
```

- Donner l'accès à l'utilisateur Apache pour lire les fichiers avec la commande ci-contre : **sudo chown -R www-data:www-data /var/www/html/phpmyadmin**
- Création d'un fichier de configuration Apache pour phpMyAdmin avec la commande : **sudo nano /etc/apache2/conf-available/phpmyadmin.conf**

```
Alias /phpmyadmin /var/www/html/phpmyadmin
```

```
<Directory /var/www/html/phpmyadmin/>
```

```
AddDefaultCharset UTF-8
```

```
<IfModule mod_authz_core.c>
```

```
<RequireAny>
```

```
Require all granted
```

```
</RequireAny>
```

```
</IfModule>
```

```
</Directory>
```

```
<Directory /var/www/html/phpmyadmin/setup/>
```

```
<IfModule mod_authz_core.c>
```

```
<RequireAny>
```

```
Require all granted
```

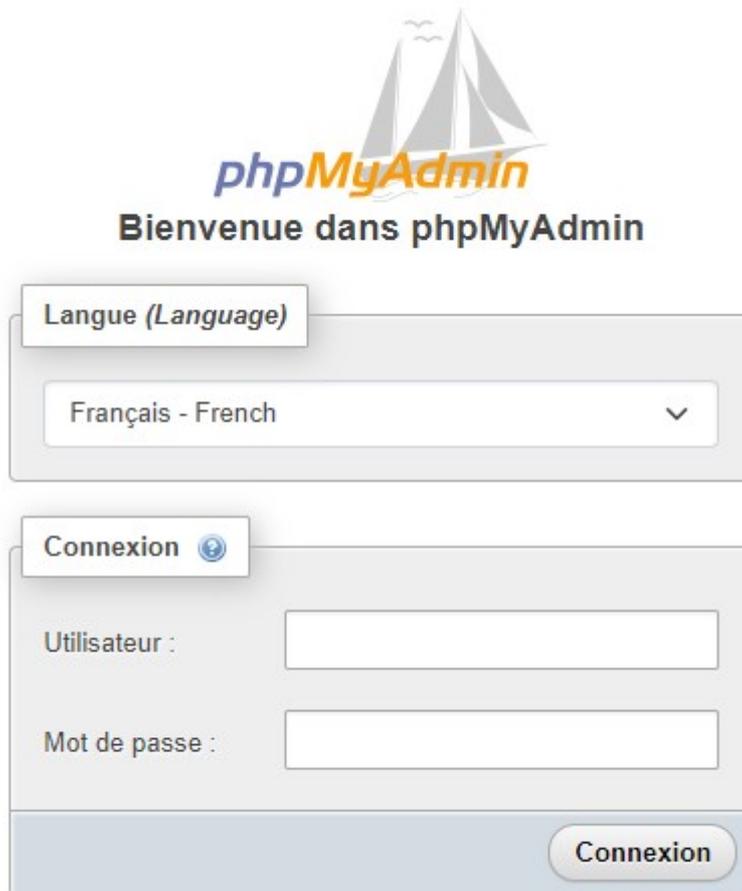
```
</RequireAny>
```

```
</IfModule>
```

```
</Directory>
```

- Activation de la nouvelle configuration avec la commande suivante : **sudo a2enconf phpmyadmin.conf**
- On redémarre le serveur Web Apache.

- A l'aide de notre navigateur, on accède à l'interface Web de phpMyadmin avec l'adresse ip suivante : <http://192.168.228.17/phpmyadmin>



phpMyAdmin

Bienvenue dans phpMyAdmin

Langue (Language)

Français - French

Connexion

Utilisateur :

Mot de passe :

Connexion

## 6. Création d'un serveur DNS

Adresse ip web : [192.168.228.17](http://192.168.228.17)

Adresse ip DNS : [192.168.228.5](http://192.168.228.5)

- Installation des paquets avec les commandes suivantes : [apt update](#)

[Apt install bind9 bind9-doc bind9utils dnsutils](#)

- Modification du fichier de configuration `named.conf.local` pour créer notre zone :

```
//  
// Do any local configuration here  
//  
zone "equipe8.m2l.local" {  
    type master;  
    file "/etc/bind/db.equipe8.m2l.local";  
};  
  
zone "228.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192.168.228.rev";  
};  
  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
include "/etc/bind/zones.rfc1918";
```

- **Création d'un fichier pour la zone equipe8.m2l.local (db.equipe8.m2l.local)**

```
GNU nano 5.4 db.equipe8.m2l.local  
$TTL 86400  
@      IN      SOA    ns.equipe8.m2l.local. hostmaster.equipe8.m2l.local. (  
        2 ;serial  
        86400 ; refresh  
        21600 ; retry  
        3600000 ; expire  
        3600 ) ; negative caching ttl  
  
@      IN      NS     ns.equipe8.m2l.local.  
ns     IN      A      192.168.228.5  
servftp IN    A      192.168.228.13  
ftp    IN      CNAME  servftp  
www    IN      A      192.168.228.17  
  
[ Read 13 lines ]  
^G Help      ^C Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U U  
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E R
```

- **Création d'un fichier de configuration pour la zone inverse (db.192.168.228.rev)**

```
;  
; BIND reverse data file for local loopback interface  
;  
$TTL      604800  
@         IN      SOA      equipe8.m21.local. hostmaster.equipe8.m21.local. (  
          1          ; Serial  
          604800     ; Refresh  
          86400     ; Retry  
          2419200   ; Expire  
          604800 )   ; Negative Cache TTL  
;  
@         IN      NS      equipe8.m21.local.  
100      IN      PTR     equipe8.m21.local.
```

## 7. Installation et configuration d'un serveur FTP

### a. Installation de vsftpd

Pour l'installation de vsftpd nous avons utilisé la commande : `sudo apt install vsftpd`

Une fois installé, nous avons fait une copie du fichier de configuration d'origine, en cas de problème.

Commande utilisée : `sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default`

```
root@ServeurDebian:/etc# cp vsftpd.conf vsftpd.conf_default
```

On a utilisé les commandes : `systemctl restart vsftpd` `systemctl enable vsftpd` pour démarrer le service.

Nous confirmons qu'il fonctionne :

```
root@ServeurDebian:/etc# systemctl restart vsftpd  
root@ServeurDebian:/etc# systemctl enable vsftpd  
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
```

### b. Création de l'utilisateur FTP

Pour la création d'un utilisateur nous avons utilisés les commandes suivantes :

```
root@ServeurDebian:/etc# sudo useradd -m ftpequipe8  
root@ServeurDebian:/etc# sudo passwd btssioequipe8
```

Ensuite nous définirons la propriété du dossier avec les commandes :

```
root@ServeurDebian:/home/ftpequipe8# sudo mkdir ftp  
root@ServeurDebian:/home/ftpequipe8# sudo chown nobody:nogroup ftp
```

Maintenant nous supprimons les autorisations d'écriture de ce dossier avec la commande ci-dessous :

```
root@ServeurDebian:/home/ftpequipe8# sudo chmod a-w ftp
```

Nous continuons à créer le répertoire du conteneur de fichiers et nous attribuerons la propriété avec les commandes suivantes :

```
Sudo mkdir /home/ftpequipe8/ftp/files
```

```
Sudo chown nombre-ftpequipe8 /home/ftpequipe8/ftp/files
```

Nous allons créer un fichier de test dans le dossier des fichiers via la commande :

```
Echo "vsftpd archivo de ejemplo" | sudo tee /home/ftpequipe8/ftp/files/ejemplo.txt
```

### c. Sécuriser le serveur FTP

Dans cette étape **ouvrons les ports 20 et 21 pour le trafic FTP**. Les ports 40000-50000 seront réservés pour la plage de ports passifs qui sera finalement définie dans le fichier de configuration et le port 990 sera utilisé lorsque TLS est activé. Exécutez ce qui suit pour le faire :

```
sudo ufw allow 20/tcp; sudo ufw allow 21/tcp; sudo ufw allow 990/tcp; sudo ufw allow 40000:50000/tcp
```

### d. Configuration de vsftpd

Puisque nous voulons que les utilisateurs puissent télécharger des fichiers, nous allons **modifier le fichier de configuration vsftpd**:

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

Dans le fichier, nous allons trouver les entrées suivantes et les décommenter:

```
anonymous_enable=NO
```

```
write_enable=YES
```

```
local_enable=YES
```

```
GNU nano 5.4 /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
local_umask=022
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
```

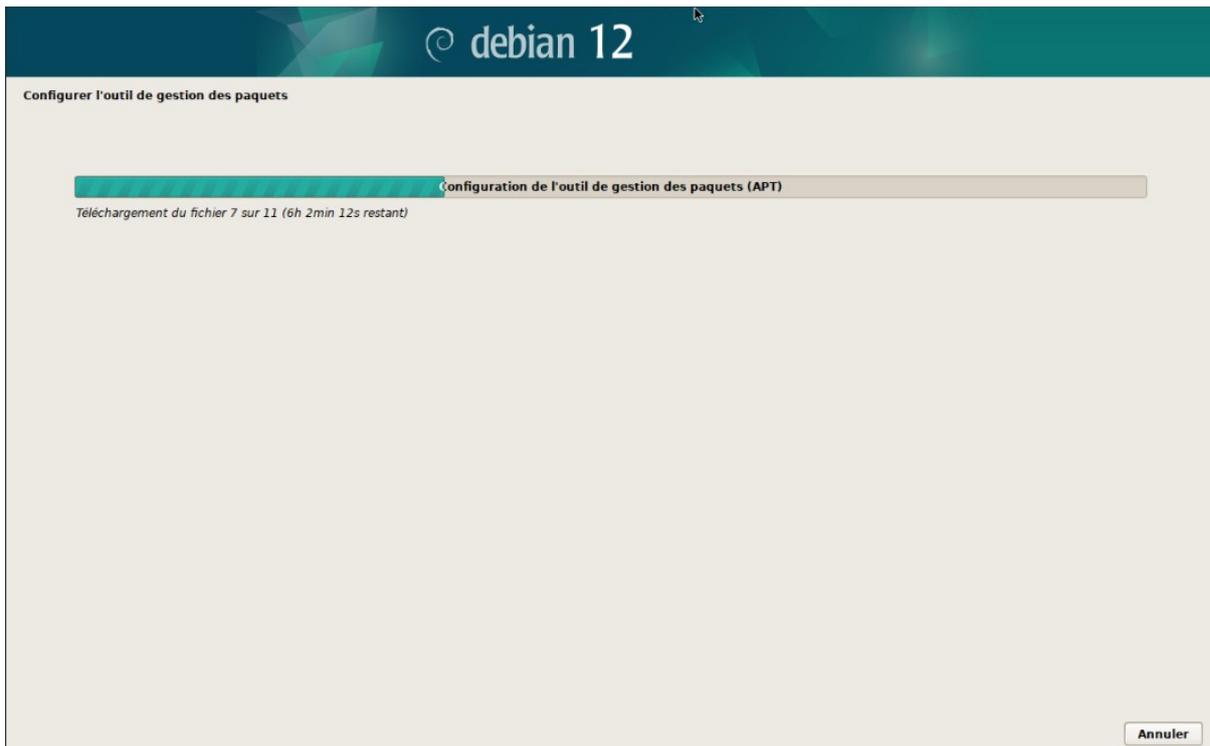
## e. Connexions cryptées

Nous utiliserons TTL / SSL pour assurer la sécurité. **Nous devons créer le certificat SSL et l'utiliser pour protéger le serveur FTP.** Nous allons faire cela avec la commande :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

## 8. Installation et configuration d'une machine Linux

On a installé et configuré une machine Linux pour effectuer des tests.



## 9. Wordpress

### a. Création d'un host virtuel pour le site wordpress

```
GNU nano 5.4                               wordpress.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin www.webmaster@localhost
DocumentRoot /var/www/wordpress
ServerName wordpress.equipe8.m2l.local

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log

[ Read 32 lines ]
^G Help      ^C Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line  M-E Redo
```

### b. Création d'un nom de domaine pour le site wordpress

```

GNU nano 5.4                                named.conf.local
//
// Do any local configuration here
//
zone "equipe8.m2l.local" {
    type master;
    file "/etc/bind/db.equipe8.m2l.local";
};

zone "228.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.228.rev";
};

// Consider adding the 1918 zones here, if they are not used in your
// organization
include "/etc/bind/zones.rfc1918";

[ Read 17 lines ]
^G Help      ^C Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Loca

```

- **equipe8.m2l.local** est le nom du domaine de notre zone

```

GNU nano 5.4                                db.equipe8.m2l.local
$TTL 86400
@      IN      SOA      ns.equipe8.m2l.local. hostmaster.equipe8.m2l.local. (
        2 ;serial
        86400 ; refresh
        21600 ; retry
        3600000 ; expire
        3600 ) ; negative caching ttl

@      IN      NS       ns.equipe8.m2l.local.
ns     IN      A        192.168.228.5
servftp IN     A        192.168.228.13
ftp    IN      CNAME    servftp
www    IN      A        192.168.228.17
wordpress IN   A        192.168.228.17

[ Read 14 lines ]
^G Help      ^C Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo

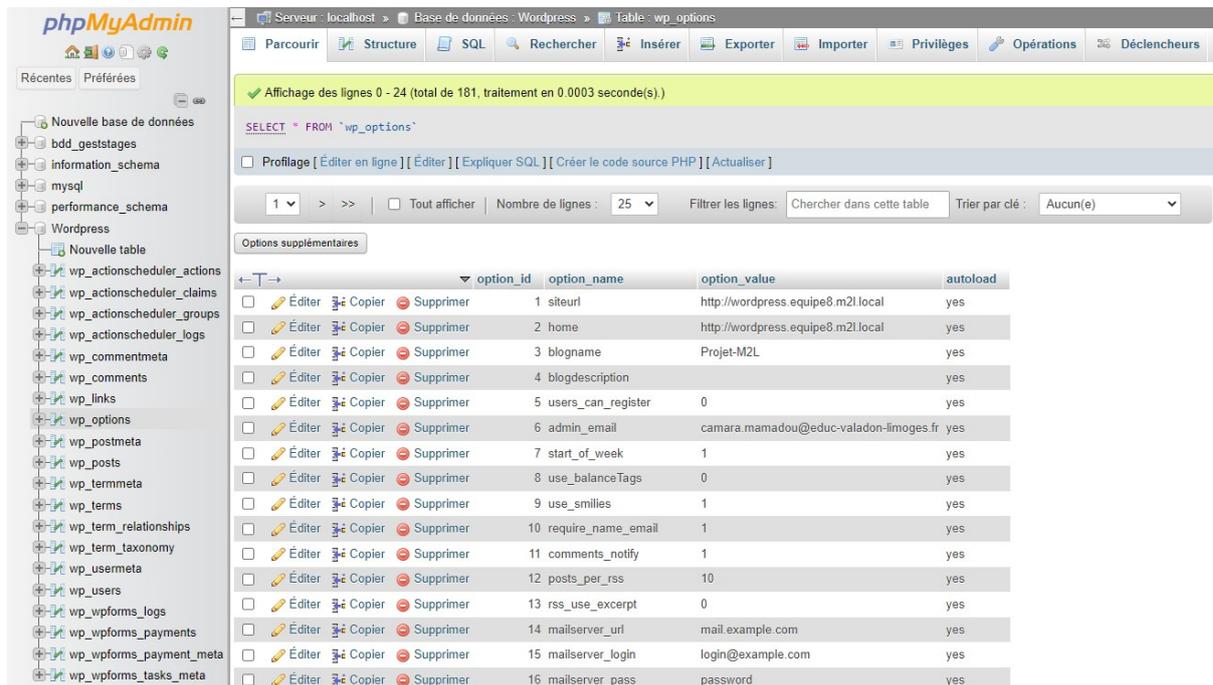
```

- **wordpress IN A 192.168.228.17** : Met en correspondance l'adresse du serveur web et le nom du serveur wordpress

### c. Problème rencontré

On a dû modifier l'ancien nom de domaine (wordpress.local) par le nouveau nom de domaine (equipe8.m2l.local). Car le site devait être nommé comme wordpress.**equipe8.m2l.local**.

Après la modification du nom de domaine, on n'arrivait pas nous connecter à la base de données du site wordpress



The screenshot shows the phpMyAdmin interface for a WordPress database. The table 'wp\_options' is selected, and the SQL query 'SELECT \* FROM `wp\_options`' is executed. The results show 16 rows of options. The first two rows, 'siteurl' and 'home', have values that include the old domain name 'http://wordpress.equipe8.m2l.local'. The other options include 'blogname', 'blogdescription', 'users\_can\_register', 'admin\_email', 'start\_of\_week', 'use\_balanceTags', 'use\_smilies', 'require\_name\_email', 'comments\_notify', 'posts\_per\_rss', 'rss\_use\_excerpt', 'mailserver\_url', 'mailserver\_login', and 'mailserver\_pass'.

option_id	option_name	option_value	autoload
1	siteurl	http://wordpress.equipe8.m2l.local	yes
2	home	http://wordpress.equipe8.m2l.local	yes
3	blogname	Projet-M2L	yes
4	blogdescription		yes
5	users_can_register	0	yes
6	admin_email	camara.mamadou@educ-valadon-limoges.fr	yes
7	start_of_week	1	yes
8	use_balanceTags	0	yes
9	use_smilies	1	yes
10	require_name_email	1	yes
11	comments_notify	1	yes
12	posts_per_rss	10	yes
13	rss_use_excerpt	0	yes
14	mailserver_url	mail.example.com	yes
15	mailserver_login	login@example.com	yes
16	mailserver_pass	password	yes

On l'a modifié via notre base de données directement, dans la base de données de wordpress, puis dans la table wp\_options, on a édité les 2 premières lignes.

Pour cela nous avons dû chercher sur internet pour trouver la solution.

Le site internet : <https://www.zenhosting.tn/blog/comment-changer-le-nom-de-domaine-de-votre-site-wordpress/>

#### d. Connexion à la base de données wordpress

Pour pouvoir se connecter à la base de données Wordpress, on a configuré le fichier wp-config.php

```
GNU nano 5.4 wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/documentation/article/editing-wp-config-php/
 *
 * @package WordPress
 */
/** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'Wordpress' );
/** Database username */
define( 'DB_USER', 'admin' );
/** Database password */
define( 'DB_PASSWORD', 'P@$$word1' );
/** Database hostname */
define( 'DB_HOST', 'localhost' );
/** Database charset to use in creating database tables. */
```

- **Wordpress** : est le nom de la base de données
- **admin** : est le nom d'utilisateur
- **P@\$\$word1** : est le mot de passe

**e. Présentation de notre site wordpress**  
[\(http://wordpress.equipe8.m2l.local/\)](http://wordpress.equipe8.m2l.local/)

Pour accéder à notre site wordpress nous devons nous connectez sur le wifi **BTSSIO-M2L**.

