



Mise en place d'une infrastructure Web complète
sous Linux (Apache, PHP, MariaDB, DNS, FTP et
WordPress)

NAOUEAL

Sommaire

1. Installation du serveur Web (Apache)

- 1.1 Installation du serveur
- 1.2 Vérification de l'adresse IP
- 1.3 Test d'accès au serveur

2. Installation et configuration de PHP

- 2.1 Préparation du système
- 2.2 Installation de PHP
- 2.3 Test de fonctionnement
- 2.4 Modules PHP et redémarrage Apache

3. Installation et gestion de MariaDB

- 3.1 Installation du serveur
- 3.2 Configuration et accès distant
- 3.3 Gestion des utilisateurs et commandes
- 3.4 Sauvegarde et import des bases

4. Installation et configuration de PhpMyAdmin

- 4.1 Téléchargement et installation
- 4.2 Configuration
- 4.3 Gestion des droits
- 4.4 Accès à l'interface

5. Mise en place d'un serveur DNS (BIND9)

- 5.1 Installation des paquets
- 5.2 Configuration de la zone directe
- 5.3 Configuration de la zone inverse
- 5.4 Résolution de noms

6. Installation et configuration d'un serveur FTP (vsftpd)

- 6.1 Installation et démarrage
- 6.2 Création d'un utilisateur FTP
- 6.3 Gestion des permissions

7. Mise en place du site WordPress

- 7.1 Configuration du serveur Web
- 7.2 Configuration DNS
- 7.3 Accès au site
- 7.4 Tests et validation

Mission 3 :

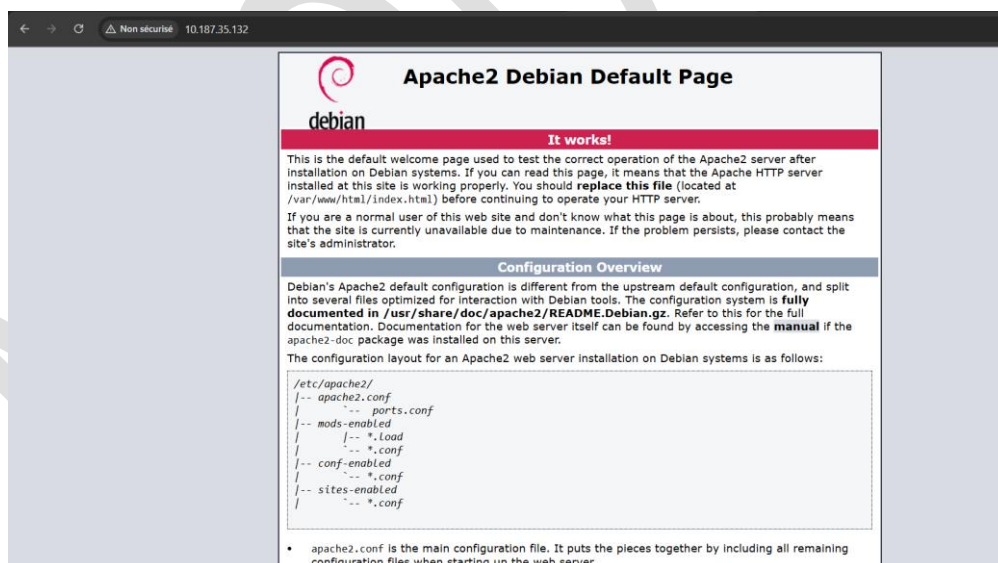
1. Installation du Serveur WEB.

On installe le serveur Web avec la commande suivante : `sudo apt install apache2 -y`

-Commande : `ip addr` (Affiche les adresse IP)

```
root@Apache:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0@if1639: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether bc:24:11:c4:8a:2c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.187.35.132/24 brd 10.187.35.255 scope global dynamic eth0
        valid_lft 13732sec preferred_lft 13732sec
    inet6 fe80::be24:11ff:fec4:8a2c/64 scope link
        valid_lft forever preferred_lft forever
```

-Puis on teste l'ip 10.187.35.132 , sur google :



2. INSTALLATION DE PHP

Avant d'installer PHP, il faut préparer le système et ajouter le dépôt officiel de PHP :

- Les paquets nécessaires et les outils de transport HTTPS :

```
apt -y install apt-transport-https lsb-release ca-certificates curl
```

- Import de la clé **GPG** du dépôt PHP :

```
wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
```

- Ajout du **dépôt PHP** à la liste des **sources** :

```
sudo bash -c 'echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list'
```

- **Mise à jour du système** :

```
apt update && sudo apt -y upgrade
```

Place maintenant à l'**installation de PHP** avec la commande ci-dessous :

```
apt -y install php
```

- **Vérification de l'installation** :

```
php -v
```

- Test de **PHP** avec un **serveur web**, on se positionnant dans le répertoire **web** :

```
cd /var/www/html
```

```
root@Apache:/var/www/html#
```

- Création d'un fichier de test **PHP** :

```
nano test.php
```

```
root@Apache:~# php -v
PHP 8.3.19 (cli) (built: Mar 13 2025 17:33:36) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.19, Copyright (c) Zend Technologies
with Zend OPcache v8.3.19, Copyright (c), by Zend Technologies
```

- Il faut **ajouter** le contenu suivant dans le fichier :

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

Vérification :

- -Puis Vérifier l'exécution dans le navigateur avec cette adresse :

<http://10.187.35.132/test.php>

| PHP Version 8.3.19 | |
|---|---|
| System | Linux Apache 6.5.11-7-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-7 (2023-12-05T09:44Z) x86_64 |
| Build Date | Mar 13 2025 17:33:36 |
| 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-ldap.ini, /etc/php8.3/apache2/conf.d/20-mbstring.ini, /etc/php8.3/apache2/conf.d/20-mcrypt.ini, /etc/php8.3/apache2/conf.d/20-mysqlnd.ini, /etc/php8.3/apache2/conf.d/20-openssl.ini, /etc/php8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php8.3/apache2/conf.d/20-posix.ini, /etc/php8.3/apache2/conf.d/20-readline.ini, /etc/php8.3/apache2/conf.d/20-shimop.ini, /etc/php8.3/apache2/conf.d/20-sockets.ini, /etc/php8.3/apache2/conf.d/20-sysvmsg.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 |

- Et enfin, installer les **modules PHP pour MariaDB** :

```
apt -y install libapache2-mod-php php-mysql
```

- Redémarrer **Apache** :

```
systemctl restart apache2
```

3. INSTALLATION DE LA BASE DE DONNEES SQL MARIADB

a. Installation du paquetage mariadb

```
apt -y install mariadb-server
```

- Créer un **compte administrateur** admin avec tous les **privilèges** :

```
sudo mariadb -u root -e "CREATE USER 'admin'@'%' IDENTIFIED BY 'P@$$word1'; GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION; FLUSH PRIVILEGES;"
```

- Test du **serveur MariaDB** :

```
ps aux | grep mysqld
```

- Vérifier sur quel **port** le serveur **écoute** :

```
ss -nlt
```

```
root@ServeurDebian:~# ss -nlt
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        [::1]:25                [::]:*                  *
```

Par défaut, MariaDB n'accepte que les connexions **locales**. Pour autoriser les connexions depuis d'autres machines :

- Modifier la configuration du **serveur MariaDB** :

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Chercher la **ligne** :

```
bind-address = 127.0.0.1
```

Remplacer par :

```
bind-address = 0.0.0.0
```

```
GNU nano 7.2                               50-server.cnf *
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#
#user                    = mysql
pid-file                 = /run/mysqld/mysqld.pid
basedir                  = /usr
#datadir                 = /var/lib/mysql
#tmpdir                  = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address             = 0.0.0.0

#
# * Fine Tuning
#
#key_buffer_size         = 128M

^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
```

Enregistrer le fichier et redémarrer MariaDB :

```
systemctl restart mariadb *
```

B. CONNEXION ET GESTION DE MARIADB

- Se connecter en tant que root

```
mariadb -u root
```

- Visualiser les **bases de données** existantes

```
SHOW DATABASES;
```

- Sélectionner une **base de données** :

```
USE <NomBaseDonnée>;
```

- Visualiser les **tables d'une base** :

```
SHOW TABLES;
```

- Visualiser les colonnes d'une table ;

```
SHOW COLUMNS FROM <NomTable>;
```

- Visualiser les comptes existants :

```
SELECT user, host FROM mysql.user;
```

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

MariaDB [(none)]> exit
```

- Définir un mot de passe pour le compte root

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('P$$word1');
FLUSH PRIVILEGES;
```

C. GESTION DES BASES DE DONNEES SQL AVEC MARIADB

- **Création** d'une base de données à partir d'un **script SQL** :

```
mariadb -u admin -p < geststages.sql
```

- **Sauvegarde** de la base de données dans un **fichier SQL** :

```
mysqldump -u admin -p geststages > sauvegardebdd.sql
```

4. INSTALLATION ET CONFIGURATION DE PHPMYADMIN

- **Téléchargement** du package :

wget <https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz>

- **Décompression** de l'archive

```
tar xvf phpMyAdmin-latest-all-languages.tar.gz
```

- **Création** du répertoire de destination :

```
mkdir /var/www/html/phpmyadmin
```

- Déplacement des fichiers

```
mv phpMyAdmin-*--all-languages/* /var/www/html/phpmyadmin
```

A. CONFIGURATION DE PHPMYADMIN :

- Copie du fichier de configuration

cp

```
/var/www/html/phpmyadmin/config.sample.inc.php/var/www/html/phpmyadmin/config.inc.php
```

p

- Création du dossier temporaire :

```
mkdir /var/www/html/phpmyadmin/tmp
```

- Modification du fichier de configuration :

```
nano /var/www/html/phpmyadmin/config.inc.php
```

- Ajouter ou modifier la ligne suivante :

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

Attribution des droits ;

```
nano /etc/apache2/conf-available/phpmyadmin.conf
```

Contenu :

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

<Directory /var/www/html/phpmyadmin/>
  AddDefaultCharset UTF-8
  <IfModule mod_authz_core.c>
    Require all granted
  </IfModule>
</Directory>

<Directory /var/www/html/phpmyadmin/setup/>
  <IfModule mod_authz_core.c>
    Require all granted
  </IfModule>
</Directory>
```

Activation de la configuration :

```
a2enconf phpmyadmin.conf
```

Redémarrage du serveur Apache :

```
systemctl restart apache2
```

Accès à phpMyAdmin :

- 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>



The screenshot shows the phpMyAdmin login interface. At the top, there is a logo for phpMyAdmin and the text 'Bienvenue dans phpMyAdmin'. Below this, there is a section for language selection with a dropdown menu currently set to 'Français - French'. Underneath, there is a 'Connexion' section with two input fields: 'Utilisateur' and 'Mot de passe'. A 'Connexion' button is located at the bottom right of the login section.

5. CREATION D'UN SERVEUR DNS

- Installation des paquets :

```
apt update  
apt install bind9 bind9-doc bind9utils dnsutils
```

Configuration de la zone directe

- Modification du fichier :

```
nano /etc/bind/named.conf.local
```

- Ajout de la zone :

```
zone "equipe5.m2l.local" {
    type master;
    file "/etc/bind/db.equipe5.m2l.local";
};
```

Création du fichier de zone directe :

nano /etc/bind/db.equipe5.m2l.local

Contenu :

```
$TTL 604800
@ IN SOA ns.equipe5.m2l.local. admin.equipe8.m2l.local. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
@ IN NS ns.equipe5.m2l.local.
ns IN A 192.168.228.5
wordpress IN A 192.168.228.17
```

Création de la zone inverse :

```
nano /etc/bind/db.192.168.228.rev
```

Contenu :

```
$TTL 604800
@ IN SOA ns.equipe5.m2l.local. admin.equipe8.m2l.local. (
    2
    604800
    86400
    2419200
```

604800)

```
@ IN NS ns.equipe5.m2l.local.
5 IN PTR ns.equipe5.m2l.local.
17 IN PTR wordpress.equipe5.m2l.local.
```

6. INSTALLATION ET CONFIGURATION D'UN SERVEUR FTP

A. INSTALLATION DE VSFTPD :

```
apt install vsftpd
```

- Sauvegarde du fichier de configuration :

```
cp /etc/vsftpd.conf /etc/vsftpd.conf_default
```

- Démarrage du service :

```
systemctl restart vsftpd
systemctl enable vsftpd
```

B. CREATION D'UN UTILISATEUR FTP

```
adduser ftpequipe5
```

- Création de l'arborescence :

```
mkdir -p /home/ftpequipe5/ftp/files
```

- Définition des permissions :

```
chown nobody:nogroup /home/ftpequipe5/ftp
```

```
chmod a-w /home/ftpequipe5/ftp
```

```
chown ftpequipe5 /home/ftpequipe5/ftp/files
```

Création d'un fichier de test :

```
echo "vsftpd fichier de test" | tee /home/ftpequipe5/ftp/files/test.txt
```

7. MISE EN PLACE DE WORDPRESS

A. CREATION D'UN HOTE VIRTUEL

Configuration d'apache pour heberger le site wordpress via un virtualhost.

B. CONFIGURATION DNS

Dans la zone DNS :

```
wordpress IN A 192.168.228.17
```

Nom de domaine utilisé :

wordpress.equipe5.m2l.local

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

