

# Installation et configuration d'un serveur LAMP

---

## Pourquoi utiliser LAMP ?

Un serveur L.A.M.P est un serveur qui s'appuie sur quatre principaux composants : Linux, Apache, Mysql, PHP. On a vu dans le TP1 Apache, qui est le serveur Web, MySQL le système de gestion de bases de données, PHP le moteur de scripts, Linux étant le système d'exploitation.

## Installer PHP

Comme toujours, on va mettre a jour les paquets avec la commande :

```
apt update
```

Puis on installe le paquet Apache pour avoir la dernière version, et l'activer pour qu'il démarre avec Debian au démarrage :

```
apt install apache2 -y  
systemctl enable apache2
```

On a ensuite accès à la page d'accueil d'Apache comme dans le TP1 en se rendant sur l'adresse IP de notre serveur.

Certains modules sont indispensables pour faire tourner un site internet, comme le module rewrite utilisé pour la réécriture d'URL, pour qu'ils soient plus lisibles.

```
a2enmod rewrite
```

A2enmod est la commande pour activer un mod, a2dismod est celle qui désactive un module. On en profite pour ajouter quelques mods comme « deflate » « headers » « ssl », pour respectivement gérer la compression, les headers HTTP, et les certificats SSL.

```
a2enmod deflate  
a2enmod headers  
a2enmod ssl
```

On redémarrera toujours apache après avoir activé ou désactivé un module, ou après n'importe quelle modification de configuration.

```
restart apache2
```

On peut trouver les fichiers de configuration de apache2 au chemin suivant :

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

Ce fichier peut servir en premier temps à configurer Apache pour ne pas afficher le numéro de version sur les pages d'erreur. On peut aussi déclarer des hôtes virtuels, mais je ne vais pas m'attarder là-dessus, nous l'avons vu au premier TP.

## Installer PHP

PHP soutient apache comme une extension, pour gérer les scripts « .php ». On va commencer par installer le paquet php lui-même.

```
apt install php -y
```

Cette commande va installer quelques modules indispensables de php qui permettent l'intégration avec apache. On va quand même installer quelques paquets supplémentaires pour avoir un php bien complet.

```
apt install php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml  
php-pear php-bcmath
```

On va vérifier que notre moteur PHP est bien actif, nous allons créer un fichier « phpinfo.php » à la racine du site

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

Dans ce fichier on mettra :

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

On pourra y accéder en ajoutant après l'IP du site /phpinfo.php, page sur laquelle on pourra voir toutes les informations de configuration de PHP et Apache. C'est une page sensible donc il ne faut pas qu'elle soit accessible à tout le monde. On la crée et utilise quand c'est nécessaire.

## Installer MySQL/MariaDB

On installe après tout ça notre base de données. On utilise le fork MariaDB, qui a l'avantage d'être open-source. Toujours la même commande pour installer :

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

Installation terminée, on va exécuter une commande pour sécuriser notre installation au minimum.

### `mariadb-secure-installation`

MariaDB va nous permettre de choisir un mot de passe pour le compte « root », empêcher les connexions distantes sur notre instance, empêcher les connexions anonymes et supprimer la base de test, après un petit interrogatoire.

On se connectera à MariaDB en utilisant la commande suivante :

```
mariadb -u root -p
```

Puis on aura accès à notre console MariaDB dans laquelle on pourra saisir des requêtes SQL comme par exemple, afficher la liste des bases de données.

```
SHOW DATABASES ;
```

On utilisera cette console quand on développera nos application sur le serveur LAMP, comme WordPress, NextCloud par exemple etc...

On peut aussi déployer PhpMyAdmin sur notre serveur pour administrer MariaDB sur l'interface graphique Web. Comme apache, on redemarre toujours MariaDB après avoir effectué des modifications de configuration.

```
systemctl restart mariadb
```

## Conclusion

Notre serveur LAMP est installé et fonctionnel. La suite de la configuration dépend de l'utilisation qu'on fera du serveur.