

Apprentissage | Installer une plateforme LAMPP

1] Le principe

L'installation d'une plate-forme LAMPP sur **Linux** se concrétise par l'installation du service **Apache**, du système de gestion de base de données **MySQL** (en fait ici d'une version plus OpenSource avec **MariaDB**) et du langage **PHP** :



Avec comme client outil **PhpMyAdmin** pour la gestion de base de données :



- **Les logiciels utilisés ou plate-forme LAMPP**
 - **Apache** pour le serveur Web (version 2.4.41 ou +)
 - **MariaDB** comme système de gestion de base de données relationnelles (version 10.3.22 ou +)
 - **PHP** pour le langage (version 7.4.3 ou +)
 - **PHPMyAdmin** pour l'administration des bases de données (version 4.9.5 ou +)

2] Première phase d'installation

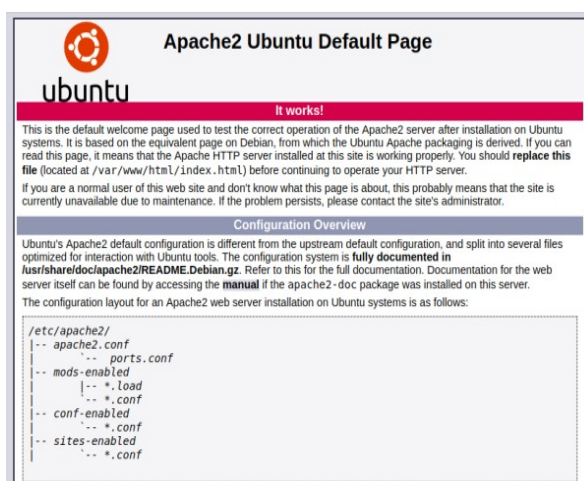
T2.1 | Dans une console et en **root**, procédez à l'installation des logiciels principaux (remplacez `aptitude` par `apt` ou `apt-get` si vous n'avez pas installé l'outil) :

```
# aptitude install apache2 php php-cli mariadb-server php-mysql
```

T2.2 | Vérifiez la bonne installation en relançant les services :

```
# service apache2 restart
# service mysql restart
```

La configuration est déjà opérationnelle comme le montre dans un navigateur Web l'URL <http://localhost> :



T2.3 | Lancez la configuration du service **MySQL** en validant par défaut les questions avec juste l'entrée d'un mot de passe pour le root par :

```
# mysql_secure_installation
```

T2.4 | Ensuite, créez un nouvel administrateur (l'exemple ici avec comme nom `sio` pour la démonstration) :

```
# mysql -u root -p
mysql> CREATE USER 'sio'@'localhost' IDENTIFIED BY 'sio';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'sio'@'localhost' WITH GRANT OPTION;
mysql> FLUSH PRIVILEGES;
mysql> exit
```

3] Deuxième phase d'installation

T3.1 | Installez PHPMyAdmin et le reste des dépendances :

```
# aptitude install phpmyadmin php-intl
```

T3.2 | A la demande de configuration de PhpMyAdmin, **cochez la case du serveur Web Apache2** (barre d'espaces) avant de valider

T3.3 | Validez par **Oui** (option par défaut) la demande de configuration de PHPMyAdmin avec `dbconfig-common`

T3.4 | Indiquez plusieurs fois le mot de passe (ici `sio`) qui peut être le même que celui de MariaDB

T3.5 | Relancez Apache :

```
# service apache2 restart
```

L'utilitaire se lance par le navigateur à la page <http://localhost/phpmyadmin> :



4] Configurations

- **Configurations de base**

Ces configurations pour le développement sont à faire en mode console avec comme éditeur `nano` ou `vim`.

Fichier `/etc/apache2/apache2.conf` :

T4.1 | Ajoutez après la directive `ServerRoot` (ligne 69) la ligne suivante :

```
ServerName "localhost"
```

Modules à activer :

T4.2 | Activez le module permettant la réécriture d'URLs, utile pour les frameworks PHP :

```
# a2enmod rewrite
```

Fichier `/etc/php/7.2/cli/php.ini` :

T4.3 | Modifiez les variables (lignes) suivantes :

```
max_execution_time = 300
...
display_errors = On
...
date.timezone = Europe/Paris
```

Respectivement vers les lignes 385, 479 et 941. Pensez à enlever le point-virgule de commentaire dans la date.

Fichier `/etc/php/7.2/apache2/php.ini` :

T4.4 | Modifiez les variables (lignes) suivantes :

```
max_execution_time = 300
...
memory_limit = 256M
...
display_errors = On
...
date.timezone = Europe/Paris
```

Respectivement vers les lignes 385, 406, 479 et 941. Pensez à enlever le point-virgule de commentaire dans la date.

T4.5 | Relancez à nouveau Apache pour la prise en compte des modifications :

```
# service apache2 restart
```

- **Configurations particulières**

T4.6 | Remplacez le contenu du fichier `/etc/apache2/sites-enabled/000-default.conf` par :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/html/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Il est plus pratique de travailler sous Linux dans le répertoire de l'utilisateur. Pour cela, il existe un module d'Apache permettant de « déporter » des pages Web dans un répertoire nommé `public_html` créé dans l'espace de l'utilisateur.

T4.7.1 | Activez le module permettant la mise en place des pages Web dans le répertoire utilisateur :

```
# a2enmod userdir
```

Par défaut, l'utilisation de PHP est interdite dans le répertoire de l'utilisateur par souci de sécurité... ce qui est plus que gênant pour le développement !

Fichier `/etc/apache2/mods-enabled/php7.2.conf` (correspondant à la version de PHP) :

T4.7.2 | Commentez (ou effacez) toutes les lignes à partir de (et inclus) jusqu'à la fin :

```
...  
# Running PHP scripts in user directories is disabled by default  
# tout à commenter...
```

T4.7.3 | Créez en tant qu'utilisateur un répertoire nommé `public_html` à la racine du répertoire personnel**T4.7.4** | Relancez à nouveau Apache pour la prise en compte des modifications :

```
# service apache2 restart
```

Les pages Web seront visibles dans un navigateur à la page <http://localhost/~sio/> (si évidemment votre utilisateur s'appelle ainsi... et au départ c'est vide).