

# UBUNTU LAMP : INSTALLATION LAMP SUR UBUNTU



## 1. Présentation

**LAMP** est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites web. L'acronyme original se réfère aux logiciels suivants :

1. **L**inux, le système d'exploitation (GNU/Linux) ;
2. **A**pache, le serveur Web ;
3. **M**ySQL ou MariaDB, le serveur de base de données ;
4. **P**HP, langage de programmation interprété qui permet de générer du contenu web dynamiquement.

## Installation d'Apache PHP et MySQL

Installation d'Apache, PHP et MySQL :

```
sudo apt install apache2 php libapache2-mod-php default-mysql-server php-mysql
```

Si vous avez un message d'erreur ouvrez le fichier `/etc/apt/sources.list`, supprimez les lignes présentes, et remplacez les par l'exemple contenant **contrib non-free** présent [ici](#).

Mettez à jour vos dépôts :

```
sudo apt update
```

et relancez la commande d'installation précédente.

Installation des modules PHP les plus courants :

```
sudo apt install php-curl php-gd php-intl php-json php-mbstring  
php-xml php-zip
```

En cas d'erreur à l'installation voir ici.

Une fois les paquets installés, vérifier l'installation soit :

- <http://127.0.0.1/>
- <http://localhost>

Si le message « **It works!** » s'affiche, votre serveur **LAMP** est correctement installé.

Si ce n'est pas le cas, vérifiez qu'apache est lancé:

## Étape 2: Installation de phpMyAdmin

phpMyAdmin est une interface web en PHP pour administrer à distance les SGBD MySQL et MariaDB.

Il permet d'administrer les éléments suivants :

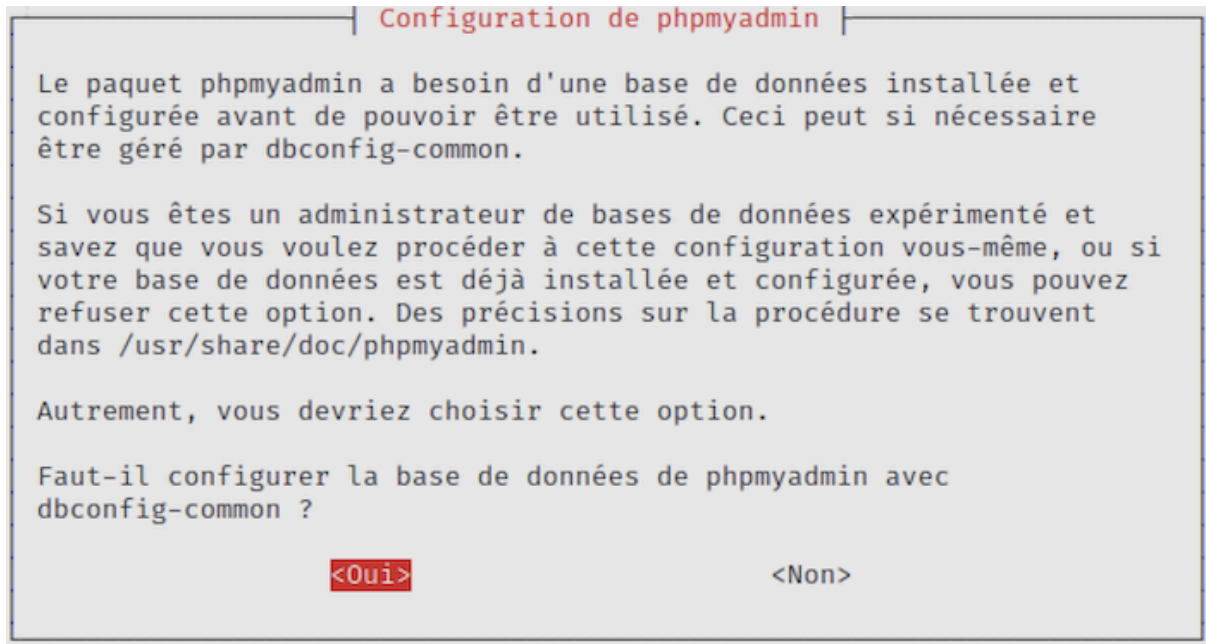
- les bases de données,
- les tables et leurs champs (ajout, suppression, définition du type),
- les index, les clés primaires et étrangères,
- les utilisateurs de la base et leurs permissions,
- importer ou exporter les données dans divers formats (CSV, XML, PDF, OpenDocument, Word, Excel et LaTeX).

Avant de pouvoir installer **phpmyadmin** vérifier que **mysql** est correctement lancé

Pour l'installer il suffit de saisir :

```
sudo apt install phpmyadmin
```

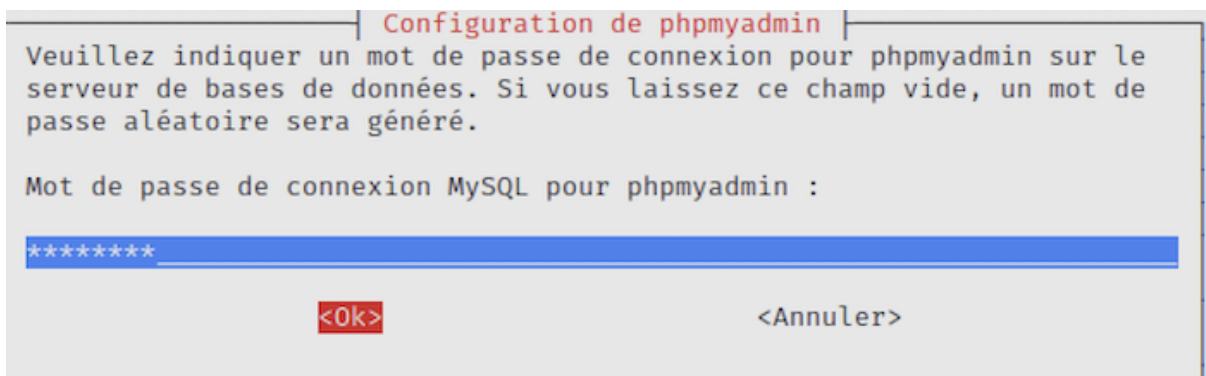
Répondez **Oui** à la première question avec la touche **tabulation** :



The screenshot shows a terminal window with the title "Configuration de phpmyadmin". The text inside reads: "Le paquet phpmyadmin a besoin d'une base de données installée et configurée avant de pouvoir être utilisé. Ceci peut si nécessaire être géré par dbconfig-common." followed by a paragraph explaining that if you are an experienced database administrator, you can skip this configuration. It then asks: "Faut-il configurer la base de données de phpmyadmin avec dbconfig-common ?". At the bottom, there are two options: "<Oui>" (highlighted in red) and "<Non>".

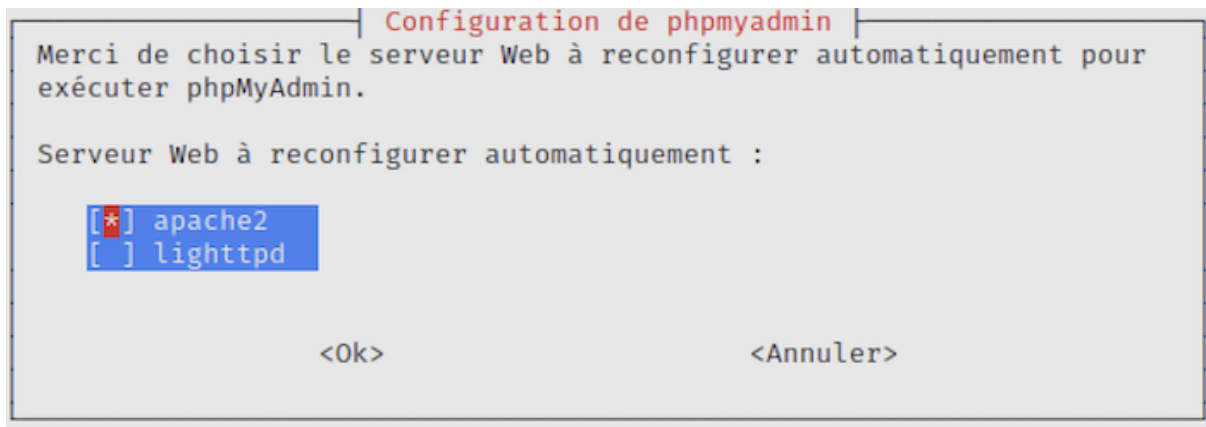
Puis validez par la touche **entrée**.

Définir ensuite un mot de passe pour l'utilisateur MySQL **phpmyadmin**, vous pouvez mettre votre mot de passe root :



The screenshot shows a terminal window with the title "Configuration de phpmyadmin". The text inside reads: "Veuillez indiquer un mot de passe de connexion pour phpmyadmin sur le serveur de bases de données. Si vous laissez ce champ vide, un mot de passe aléatoire sera généré." followed by the prompt "Mot de passe de connexion MySQL pour phpmyadmin :". Below the prompt is a blue input field containing "\*\*\*\*\*". At the bottom, there are two options: "<Ok>" (highlighted in red) and "<Annuler>".

Choisir le serveur web à configurer automatiquement (utiliser les **flèches** du clavier pour se déplacer et la **barre d'espace** pour sélectionner/désélectionner, la touche **tab** vous permettra de choisir **OK**). **Attention le surlignage rouge n'est pas une sélection, il faut qu'une étoile \* s'affiche entre les crochets.**



phpMyAdmin devrait être accessible à l'adresse <http://localhost/phpmyadmin>.

## En cas de problèmes

Si vous ne pouvez pas accéder à l'url, c'est que vous avez probablement oublié d'appuyer sur espace pour sélectionner apache2, au moment de l'installation de phpMyAdmin.

Vous devez donc purger l'installation de phpMyAdmin :

```
sudo apt purge phpmyadmin
```

Puis relancer l'installation :

```
sudo apt purge phpmyadmin
```

## Connexion à phpMyAdmin

Avec MySQL depuis Bionic 18.04, et MariaDB depuis Xenial 16.04, l'authentification de l'utilisateur root de MySQL se fait au moyen du plugin auth\_socket, donc avec sudo.

La solution conseillée est donc de créer un nouvel utilisateur et de lui attribuer tous les privilèges :

```
sudo mysql
```

Puis dans la console MySQL (Attention, pensez à remplacer l'**utilisateur** par votre nom d'utilisateur et **mot\_de\_passe** par votre mot de passe).

Création de l'utilisateur :

```
CREATE USER 'utilisateur'@'%' IDENTIFIED BY 'mot_de_passe' ;
```

Attribution des droits :

```
GRANT ALL PRIVILEGES ON *.* TO 'utilisateur'@'%' WITH GRANT OPTION ;
```

Puis :

```
FLUSH PRIVILEGES;
```

Et enfin :

```
QUIT;
```

Cette solution peut être utile sur un serveur de développement mais elle rend plus dangereuse une attaque par force brute sur la page de connexion de phpMyAdmin. En production il est indispensable de protéger l'accès à phpMyAdmin grâce à l'outil fail2ban, et à des mesures de restriction d'accès par IP par exemple.

## Commandes à connaître:

Vérifier si apache et mysql sont lancés

```
sudo systemctl status apache2  
sudo systemctl status mysql
```

Lancer apache et mysql au démarrage de l'OS

```
sudo systemctl enable apache2  
sudo systemctl enable mysql
```

Empêcher apache et mysql de démarrer automatiquement

```
sudo systemctl disable apache2  
sudo systemctl disable mysql
```

Démarrer apache et mysql

```
sudo systemctl start apache2  
sudo systemctl start mysql
```

## Stopper apache et mysql

```
sudo systemctl stop apache2  
sudo systemctl stop mysql
```

## Redémarrer apache et mysql notamment après une modification

```
sudo systemctl restart apache2  
sudo systemctl restart mysql
```

## 6. Problème de verrou à l'installation

Il se peut que vous ne puissiez pas obtenir lancer les commandes d'installation car l'application **apt** est déjà lancée, souvent pour réaliser des mises à jour.

Pour contourner ce problème vous pouvez tuer le processus :

```
sudo killall apt apt-get
```

Supprimer les fichiers **lock** en fonction du message affiché :

```
sudo rm /var/lib/apt/lists/lock  
sudo rm /var/cache/apt/archives/lock  
sudo rm /var/lib/dpkg/lock  
sudo rm /var/lib/dpkg/lock-frontend
```

Puis reconfigurer les paquets **si besoin** :

```
sudo dpkg --configure -a
```

## 7. Droits d'accès au dossier html

Afin de pouvoir modifier facilement les fichiers présents dans **/var/www/html**, nous allons :

ajouter notre utilisateur au groupe **www-data** :

```
sudo chown -R $USER:www-data /var/www/html
```

devenir propriétaire du dossier avec le groupe **www-data** :

```
sudo chown -R $USER:www-data /var/www/html
```

pour terminer, nous donnons les droits de lecture, d'écriture et d'exécution au propriétaire et au groupe (**www-data** dans notre cas). Les autres utilisateurs auront un accès en lecture et en exécution :

```
sudo chmod -R 775 /var/www/html
```