

# Installation de MedShareEHR sur Ubuntu Server 18.04 LTS

## Installation ubuntu server (version 18.04 LTS)

Etapes non détaillées

- Télécharger ISO
- Graver ISO sur clef USB
- Booter PC sur clef USB
- Installation ubuntu Server sur PC depuis clef USB

Les outils suivants seront nécessaires pour les installations de modules MedShakeEHR

Installation des outils de compression zip et unzip

```
sudo apt install zip unzip
```

Installation de l'outil make

```
sudo apt install make
```

Terminer par une mise à jour pour appliquer les derniers patches de sécurité :

```
sudo apt update  
sudo apt upgrade
```

# Installation utilitaires divers

Différents outils sont utiles pour la gestion d'un serveur :

## Gestion De la Base de Données MySQL : phpmyadmin

```
sudo apt install phpmyadmin
```

A la fin de l'installation la console de gestion phpmyadmin est accessible via :

<https://<<adresse.ip.du.serveur>>/phpmyadmin>

Le login / password à utiliser pour phpmyadmin **n'est pas** le même que celui utilisé pour se loguer en console linux . il faut utiliser phpmyadmin/passwordspec avec passwordspec = password spécifié pendant installation phpmyadmin pour le user phpmyadmin.

## Installation de Webmin

(reste à faire)

## Installation de cockpit (surveillance de la consommation des ressources en temps réel)

```
sudo apt install cockpit
```

A la fin de l'installation la console de gestion cockpit est accessible via :

<https://<<adresse.ip.du.serveur>>:9090/>

Le login / password à utiliser pour cockpit est le même que celui utilisé pour se loguer en console linux

# Installation du Stack LAMP sur ubuntu Server

Référence de l'article : <https://doc.ubuntu-fr.org/lamp>

On peut déployer la pile LAMP sur n'importe quelle version ou [variante](#) d'Ubuntu. La variante [Ubuntu server](#) est particulièrement adaptée dans le cas d'un serveur "headless" (sans interface graphique) contrôlable à distance (par [ssh](#) notamment).

## Méthode recommandée : installation des paquets

Cette méthode consiste à [installer](#) les paquets nécessaires pour Apache, PHP et MySQL :

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

ou si vous préférez opter pour [MariaDB](#) au lieu de MySQL :

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

(attention le paquet php installe directement la dernière version de php)

La pile LAMP est alors installée. Cependant la plupart des scripts [PHP](#) ([CMS](#), forums, applications web en tout genre) utilisent des [modules de PHP](#) pour bénéficier de certaines fonctionnalités.

Voici comment installer les modules les plus courants :

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

Description des paquets :

- Le paquet [apache2](#) installe le serveur HTTP [Apache 2](#) (c'est une dépendance de [libapache2-mod-php](#)).
- Le paquet [php](#) méta-paquet permettant d'installer au moins un interpréteur [PHP](#) (aussi installé ici en dépendance de [libapache2-mod-php](#)).
- Le paquet [libapache2-mod-php](#) module d'Apache (on peut aussi utiliser PHP en ligne de commande ou indépendamment en FastCGI, ce qui le rend plus performant mais plus compliqué à mettre en place)
- Le paquet [mysql-server](#) installe le serveur de bases données [MySQL](#).
- Le paquet [mariadb-server](#) installe le serveur de base données [MariaDB](#).
- Le paquet [php-mysql](#) installe les module permettant d'utiliser MySQL ou MariaDB avec PHP.

Une fois les paquets installés, ouvrez un des liens suivants dans votre navigateur :

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

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

Selon la version installée, vous pouvez aussi avoir une page intitulée « Apache2 Ubuntu Default Page ».

# Configuration de l'installation Apache et MySQL

## Configuration d'Apache 2

La configuration et les détails à propos d'Apache 2 sont accessibles sur la [page dédiée à ce serveur HTTP](#).

## Configuration de MySQL

La configuration et les détails à propos du serveur de base de données sont accessibles sur la [page dédiée à MySQL](#) ou [celle dédiée à MariaDB](#).

Il est courant d'installer également [PHPMyAdmin](#) afin de gérer la base de donnée MySQL dans une interface Web écrite en PHP.

On peut aussi utiliser [Adminer](#) qui est plus simple à installer (il s'agit d'un simple fichier PHP) et propose à peu près les mêmes fonctionnalités.

## Configuration du démarrage automatique de LAMP

Par défaut, Apache et MySQL ou MariaDB démarrent automatiquement lorsque vous allumez votre ordinateur.

*Empêcher LAMP de démarrer automatiquement*

Pour empêcher cela :

```
sudo systemctl disable apache2
```

Et pour mysql :

```
sudo systemctl disable mysql
```

Vous pourrez donc ensuite les lancer manuellement.

Pour Apache :

```
sudo systemctl start apache2
```

et pour MySql

```
sudo systemctl start mysql
```

Les commandes disponibles sont *systemctl start*, *systemctl stop*, et *systemctl restart*.

*Réactiver le démarrage automatique*

Pour réactiver les services automatiquement au démarrage, c'est simple :

```
sudo systemctl enable apache2
```

```
sudo systemctl enable mysql
```

## Utilisation

La mise en place d'un site internet, [CMS](#), ou autre application web se fait en général en [créant un hôte virtuel](#) pour [Apache](#), et une [base de donnée MySQL](#).

### Création d'un hôte virtuel

Par exemple pour mettre en place une solution qu'on appellera ici *example* sur un nom de domaine *example.com*, en créera un fichier `/etc/apache2/sites-available/example.com.conf` contenant :

[example.com.conf](#)

```
<VirtualHost *:80>
  ServerName example.com
  ServerAlias www.example.com
  DocumentRoot "/var/www/example"
  <Directory "/var/www/example">
    Options FollowSymLinks
    AllowOverride all
    Require all granted
  </Directory>
  ErrorLog /var/log/apache2/error.example.com.log
  CustomLog /var/log/apache2/access.example.com.log combined
</VirtualHost>
```

On déploiera ici les fichiers du site dans le répertoire `/var/www/example`.

Puis on active l'hôte virtuel et on recharge la configuration d'Apache :

```
sudo a2ensite example.com
sudo systemctl reload apache2
```

## Configuration MySQL

### Accès root

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](#). Cette méthode ne permet pas de se connecter avec phpMyAdmin, mais il est [vivement déconseillé](#) de modifier ce comportement.

Si vous avez besoin d'un accès global à vos bases de données depuis un même compte, la solution conseillée est donc de créer un nouvel utilisateur et de lui attribuer tous les privilèges :

```
sudo mysql
```

Puis dans la console MySQL :

```
GRANT ALL ON *.* TO 'nom_utilisateur_choisi'@'localhost' IDENTIFIED BY '
mot_de_passe_solide' WITH GRANT OPTION;
FLUSH PRIVILEGES;
QUIT;
```

En remplaçant évidemment *nom\_utilisateur\_choisi* et *mot\_de\_passe\_solide* dans cette requête.

Ce seront ces informations qui devront être utilisées pendant la procédure d'installation de MedShakeEHR

# Installation du logiciel MedShakeEHR

Doc source : <http://www.logiciel-cabinet-medical.fr/documentation-technique/environnement-de-production-necessaire-a-medshakeehr.html>

## Préparation des répertoires pour l'installation du logiciel

```
sudo mkdir /home/ehr
sudo mkdir /home/ehr/public_html/
cd /home/ehr/public_html/
sudo wget https://raw.githubusercontent.com/MedShake/MedShakeEHR-base/master/installer/self-installer.php
sudo chmod -R 775 /home/ehr
sudo chown -R www-data:www-data /home/ehr
```

## Création d'un hôte virtuel dédié pour l'installation de MedShakeEHR

Doc source : <https://doc.ubuntu-fr.org/apache2>

Par exemple pour mettre en place un serveur web qu'on appellera ici *msehr* sur un nom de domaine *msehr*, on créera un fichier */etc/apache2/sites-available/msehr.conf* contenant :

```
<VirtualHost *:80>
  ServerName msehr
  ServerAlias ehr medshakeehr MedShakeEHR
  DocumentRoot "/home/ehr/public_html"
  <Directory "/home/ehr/public_html">
    Options FollowSymLinks
    AllowOverride all
    Require all granted
  </Directory>
  ErrorLog /var/log/apache2/error.msehr.log
  CustomLog /var/log/apache2/access.msehr.log combined
</VirtualHost>
```

Si le nom msehr n'existe pas sur le réseau (pas de serveur DNS local) il est aussi possible

- Soit de modifier le fichier */etc/hosts* du PC, pour le faire pointer sur le serveur avec un lien entre 192.168.0.100 et msehr. **C'est la configuration recommandée pour une installation simple**
- Soit de créer le host virtuel pour qu'il pointe directement sur l'adresse IP du serveur. Il faut alors remplacer *msehr* par *192.168.0.100* (si l'adresse du serveur est 192.168.0.100)

Suivant recommandation de l'éditeur du logiciel on déploiera ici :

- les fichiers d'installation dans le répertoire *"/home/ehr/public\_html"*.



- Les fichiers de production dans le répertoire *"/home/ehr"*

Puis on active l'hôte virtuel et on recharge la configuration d'Apache :

```
sudo a2ensite msehr
sudo systemctl reload apache2
```

**On doit activer le module apache mod\_rewrite pour que l'installateur MedShakeEHR fonctionne**

```
sudo a2enmod rewrite
sudo systemctl reload apache2
```

On lance alors le script d'installation de l'application sous le serveur Web . Pour cela accéder par le navigateur au fichier self-installer.php , à l'adresse :

- Soit <http://msehr/self-installer.php>.
- Soit <http://192.168.0.100/self-installer.php>.

Dans les écrans qui vont suivre :

- Indiquer /home/ehr comme répertoire d'installation.
- Dérouler ensuite le processus d'installation.

# Configurer les sauvegardes

Il semblerait que msehr fasse des sauvegardes tout seul, mais cela reste à prouver ...