Installation d'un serveur web Debian9 - PHP 7.1 - MariaDB - Apache2 - CMS Joomla

Voici un mémo pour faire un serveur web de base pour le CMS Joomla.

Note : Cette installation est juste pratique, il faudra ajouter vos certificats, configurer correctement Debian et Apache pour pas lâcher trop d'informations.

Attention, le VHOST est à configurer en fonction de vos besoins, celui-ci est volontairement "léger".

Installation

Récupérer l'iso Debian9 net-install et faire l'installation de base (ssh et fichiers de bases)

Récup des dépôts pour php 7.1 :

```
apt-get install apt-transport-https lsb-release ca-certificates
wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" >
/etc/apt/sources.list.d/php.list
apt-get update
```

Installation PHP 7.1 et Apache 2

```
apt install --no-install-recommends php7.1 libapache2-mod-php7.1 php7.1-mysql php7.1-curl php7.1-
json php7.1-gd php7.1-mcrypt php7.1-msgpack php7.1-memcached php7.1-intl php7.1-sqlite3 php7.1-gmp
php7.1-geoip php7.1-mbstring php7.1-redis php7.1-xml php7.1-zip php7.1-cli php7.1-common
```

```
apt install apache2 apache2-utils unzip
```

** On active php pour Apache et le mode rewrite (ré écriture au vol des URL) :

```
a2dismod mpm_event
a2enmod php7.1
a2enmod rewrite
```

** On vérifie le fonctionnement en mettant une page php dans /var/www/html/ , (le dossier html est le dossier par défaut d'Apache). ** J'utilise nano mais vous pouvez choisir votre éditeur.

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

** On met ce code à l'intérieur :

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

** On enregistre et on ferme l'éditeur nano (Ctrl+O et Ctrl+X)

`systemctl restart apache2`

Pour vérifier le fonctionnement et la version, sur le navigateur **ip/test.php**

** Lorsque c'est vérifié, pour éviter une faille de sécurité, il est plus prudent de supprimer le fichier.
`rm /var/www/html/test.php`

**Passons à l'installation de la base de donnée :
`apt-get install mariadb-server mariadb-client`

** Sécuriser MariaDB :
`mysql_secure_installation`

** Comme demandé, tape le mot de passe root de Debian.
** Puis répondre "Y" (ou Entrée) à toutes les questions suivantes.

** Commence par donner un mot de passe à l'utilisateur root de MariaDB différent de l'utilisateur root de votre Debian.
** Cet utilisateur root de la base de données aura tous les droits d'accès, pour des raisons évidentes de sécurité, je vous recommande d'utiliser un mot de passe complexe !

** Maintenant, on met à jour de nouveau :
`apt update && apt upgrade`

Installation terminé

Création des DOSSIERS

** Il faut charger les systèmes CMS (depuis un wget ou via WinSCP) et placer les fichiers dans les dossiers correspondants.
`cd /var/www/`
`wget https://github.com/AFUJ/joomla-cms-fr/archive/master.zip`
`unzip master.zip`
`mv joomla-cms-fr-master test`
(** pour renommer le dossier dézippé en *test*)

** Remise en place des droits, maintenant, on ré-attribue le propriétaire et le groupe www-data aux dossiers dans /var/www/ :
`cd /var/www/`
`chown -R www-data: *`

** Lorsque le site aura besoin "d'écrire", il faut lui permettre de le faire avec l'utilisateur / groupe www-data.

** Créer les VHosts (exemple pour Joomla!) - Ne pas oublier le port d'écoute !!
** !! Attention, Apache ne prend pas en compte les fichiers de configuration qui ne se termine pas par "**.conf" !!
** !! Attention : le VHOST est volontairement simple, à ne pas utiliser en dehors du réseau local

`nano /etc/apache2/sites-available/biomediqa.conf`

```
<VirtualHost *:80>
ServerAdmin contact@test.com
ServerName test
DocumentRoot /var/www/test/
</VirtualHost>
```

** Mise en place du site pour Apache
`a2ensite test`

** Une fois fait, il faut mettre à jour le serveur DNS pour lier l'IP et le vhost (si c'est en local, le fichier host de votre distribution fera l'affaire)

`x.x.x.x test`

Création de la Base de Données

** Mise en place de l'utilisateur et de la base de donnée pour test:
`mysql -u root -p`

--CREATE DATABASE nom_db;
--GRANT ALL PRIVILEGES ON nom_db.* TO "user_db"@"localhost" IDENTIFIED BY 'mot_de_passe';
--FLUSH PRIVILEGES;

`CREATE DATABASE test_db;`
`GRANT ALL PRIVILEGES ON test_db.* TO "testuser"@"localhost" IDENTIFIED BY '123456789';`
`FLUSH PRIVILEGES;`
`exit`

** Lorsque c'est fait pour chaque site, et vu que nous sommes au début, on peut reboot.
** Ensuite se connecter sur le nom de domaine (http://test)

Commandes utiles pour manager la base de données en ligne de commande

** Pour lister les tables de données :
`show databases;`

** Supprimer une bdd
`mysql> drop database exemple;`

** Création d'un utilisateur
Première action, création d'un utilisateur avec un mot de passe. Deux façons de faire sont possibles en fonction de la façon dont vous voulez attribuer le mot de passe au moment de la création de l'utilisateur.

** Lister les utilisateurs
`select user,host from mysql.user;`

- -- Mot de passe en clair dans la requête
  `CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';`
- -- Mot de passe passé sous un algorithme de Hash
  `SELECT PASSWORD('password');` -- Création du Hash du mot de passe
  `CREATE USER 'user'@'localhost' IDENTIFIED BY PASSWORD '*2470C0C06DEE42FD1618BB9900DFG1E6Y89F4';`

** Renommer un utilisateur
`RENAME USER 'user'@'localhost' TO 'user2'@'localhost';`

** Changer un mot de passe
`SET PASSWORD FOR 'user'@'localhost' = PASSWORD('newpassword');`

** Attribution de privilèges
`GRANT SELECT, INSERT, UPDATE, DELETE ON `database`.* TO 'user'@'localhost';`

** Si vous voulez attribuer tout les droits sur une base de données à un utilisateur, il vous suffit d'effectuer la requête suivante :
`GRANT ALL ON `database`.* TO 'user'@'localhost';`
`FLUSH PRIVILEGES;`

** Révocation de privilèges
`REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'user'@'localhost';`

** Voir les priviléges d'un utilisateur sur une base
`show grants for "utilisateur"@"localhost";`

** Suppression Utilisateur
`DROP USER 'user'@'localhost';`

Revision #1
Created 2019-10-31 12:03:50 UTC by Cécile

Updated 2019-10-31 12:04:21 UTC by Cécile