Guide rapide sur l'installation d'un serveur LAMP sur Ubuntu Server 22.04 ou 24.04 destiné à l'hébergement d'un site sous WordPress

Pour installer un serveur web afin d'héberger un site WordPress fonctionnant sous Linux (Ubuntu Server 22.04 ou 24.04), utilisant Apache 2, PHP 8, MySQL (ou le fork MariaDB) et Redis (utile pour les dernières versions de WordPress), suivez ce guide.

Remarques :

1. Ce guide ne constitue pas réellement un tutoriel pas-à-pas mais plutôt une liste de choses à faire, et les commandes clés à utiliser. Les fichiers de configurations des différents paquets sont à adapter à vos exigences.
2. Ce guide a été rédigé en m'appuyant sur un VPS fonctionnant sur Ubuntu Server dans sa version 22.04.4. Il est possible et même probable qu'il faille adapter certaines étapes pour réussir la mise en place de cette pile logicielle sur la version 24.04 d'Ubuntu Server.
3. A ce stade de l'évolution d'Ubuntu et à la date du 17 mai 2024, je vous recommande pour le moment de rester sur la version 22.04 puisque Ubuntu 24.04 est disponible depuis relativement peu de temps. Je vous conseille en effet d'attendre la version 24.04.1 au minimum avant de mettre en production cette version sur votre ou vos serveurs. La version 24.04.1 devrait être disponible dans le courant de l'été 2024.
4. Toutes les manipulations suivantes seront faites à distance via SSH (pas d'accès physique au serveur).
5. Je pars du principe que vous avez un utilisateur autre que root mais que vous possédez les droits pour utiliser la commande sudo (que vous êtes sudoer).

Pré-requis

1. Vous venez d'installer Ubuntu Server sur votre serveur physique ou virtuel et vous n'avez donc rien fait de plus pour le moment sur votre système.
2. Vous avez coché l'installation de OpenSSH lors de l'installation du système.
3. Vous savez comment vous connecter via SSH à votre serveur.
4. Votre serveur est connecté à Internet et la connexion est parfaitement fonctionnelle.
5. Vous possédez un nom de domaine et ce dernier pointe sur le serveur qui hébergera votre site à l'issue de ce guide.
6. Vous avez correctement ouvert les ports 80 et 443 sur votre routeur et configuré votre reverse proxy (connexions domestiques uniquement, pré-requis potentiellement optionnel).

Étape 1 : Connexion via SSH et mise à jour de la machine

En premier, connectez-vous à votre système via SSH.
Une fois que vous êtes connecté, mettons à jour la machine. En effet, les images disques téléchargées sur le site d'Ubuntu ne contiennent pas toujours les toutes dernières versions des paquets.
Pour ce faire, utilisez la commande suivante :

```
$ sudo apt update && sudo apt upgrade -y && sudo apt autoremove --purge -y
```

Étape 2 : Installation des paquets constituants la pile logicielle

Nous allons maintenant installer Apache 2, PHP 8, MariaDB et Redis et quelques utilitaires permettant par exemple de gérer les bases de données et de faire la connexion entre PHP et le serveur de bases de données.

```
$ sudo apt install apache2 php mysql-server phpmyadmin libapache2-mod-php php-xml
php-gd php-curl php-mysql php-redis redis-server
```

Étape 3 : Configuration d'Apache

Éditer le fichier suivant pour le faire correspondre à vos besoins : /etc/apache2/apache2.conf
Par exemple, avec nano.

```
$ sudo nano /etc/apache2/apache2.conf
```

Éditez notamment le bloc <Directory /var/www/> et plus précisément la ligne `AllowOverride None` qu'il faudra remplacer par `AllowOverride All` .

Si vous souhaitez utiliser les hôtes virtuels afin d'héberger plusieurs sites sur un même serveur accessible via des noms de domaine différents, suivez les étapes ci-dessous :

Dupliquer le fichier `/etc/apache2/sites-available/000-default.conf` et nommez le fichier par le nom de domaine du site qu'il servira. Pour l'exemple, disons toto.conf.

Modifiez ce fichier en renseignant le nom de domaine du site concerné (ne pas oublier de dé-commenter la ligne) et son chemin physique sur le serveur, par exemple `/var/www/site-de-toto`.

Il est ensuite nécessaire d'activer l'hôte virtuel créé avec la commande suivante :

```
$ sudo a2ensite toto.conf
```

Profitez-en pour activer le mod rewrite d'Apache, qui est nécessaire, avec la commande suivante :

```
$ sudo a2enmod rewrite
```

Étape 4 : Configuration de PHP

Avec PHP il est nécessaire d'éditer le fichier suivant : /etc/php/8.1/apache2/php.ini (Ubuntu 22.04) ou /etc/php/8.3/apache2/php.ini (Ubuntu 24.04)

Ubuntu 22.04(.4) :

```
$ sudo nano /etc/php/8.1/apache2/php.ini
```

Ubuntu 24.04 :

```
$ sudo nano /etc/php/8.3/apache2/php.ini
```

Éditer la quantité de mémoire allouée à PHP ainsi que la taille des fichiers transférables via PHP dans la partie **File Uploads** de ce fichier.

Les variables à éditer sont les suivantes :

```
memory_limit = 128M         ->    memory_limit = 256M
post_max_size = 8M          ->    post_max_size = 130M
upload_max_filesize = 2M    ->    upload_max_filesize = 128M
```

Étape 5 : Configuration de MariaDB/MySQL

Se connecter à MySQL avec la commande `sudo mysql`, et utiliser les commandes suivantes pour créer votre utilisateur et votre mot de passe MySQL ainsi que pour vous donner les droits nécessaires.

```
CREATE USER 'nom_utilisateur'@'localhost' IDENTIFIED BY 'mot_de_passe';
GRANT ALL PRIVILEGES ON *.* TO 'nom_utilisateur'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
QUIT;
```

Les éléments que vous devez **impérativement** remplacer sont en gras dans les commandes ci-dessus.

Étape 6 : Configuration de Redis

Cette partie est un tout petit peu plus délicate.
Éditez le fichier `/etc/redis/redis.conf`.

```
$ sudo nano /etc/redis/redis.conf
```

A la fin du fichier, ajouter les lignes suivantes :

```
maxmemory 256mb
maxmemory-policy allkeys-lru
```

Une fois WordPress installé, il ne faudra pas oublier d'installer l'extension Redis, et de modifier votre fichier wp-config.php pour y ajouter les lignes suivantes après la section « Authentication Unique Keys and Salts » :

```
define('NONCE_SALT', 'phrase de passe unique à insérer ici');
define('WP_CACHE_KEY_SALT', 'exemple.com');
define('WP_CACHE', true);
```

Redémarrez Redis avec la commande `sudo service redis-server restart`.
Redémarrez Apache avec la commande `sudo service apache2 restart`.

Étape 7 : Création du dossier qui accueillera le site et attribution des permissions

Créez le dossier qui accueillera les fichiers du site sur votre serveur.
Le chemin et le nom du dossier doit être identique à ce qui a été renseigné dans le fichier de l'hôte virtuel Apache à l'étape 3 de ce guide.
Dans mon cas je vais donc créer le dossier `/var/www/site-de-toto` via la commande suivante :

```
$ sudo mkdir /var/www/site-de-toto
```

Je dois pour le moment utiliser sudo puisque pour mon utilisateur sur le serveur n'a pas encore les droits sur le dossier `/var/www`. Dans la commande précédente (étape 6), j'utilise donc sudo afin que ce soit l'utilisateur root qui créé le dossier.
Je vais donc maintenant me donner les droits sur l'ensemble du dossier `/var/www` afin de pouvoir ensuite envoyer les fichiers via SFTP dans mon dossier « site-de-toto » avec mon utilisateur personnel configuré sur le serveur lors de la configuration initiale d'Ubuntu Server.

La première commande ci-dessous ajoute mon utilisateur « paul » au groupe www-data, qui est le groupe de l'utilisateur du serveur web.
Ceci nous sera utile pour l'utilisation de WordPress.

```
$ sudo adduser paul www-data
```

```
$ sudo chown -R www-data:www-data /var/www
$ sudo chmod -R 775 /var/www
```

- La première commande ci-dessus donne à l'utilisateur **www-data** (utilisateur du serveur Apache) appartenant cette fois au groupe **www-data** la propriété sur le dossier `/var/www` de manière récursive. C'est à dire que les dossiers et fichiers contenus dans ce dossier « www » appartiendront eux aussi à l'utilisateur www-data. C'est l'option « -R » de la commande qui permet ce comportement.
- La seconde commande change les permissions du dossier `/var/www` de manière récursive (option « -R » encore une fois) en 775. C'est à dire lecture, écriture et exécution autorisées pour l'utilisateur propriétaire et le groupe auquel il appartient, mais lecture et exécution autorisées uniquement pour tout le monde.

Étape 8 : Création de la base de données via phpMyAdmin

Rendez-vous sur phpMyAdmin via l'adresse http://ip-du-serveur/phpmyadmin, connectez-vous à ce SGBD grâce aux identifiants créés à l'étape 5.
Créez une base de données pour votre site WordPress, par exemple, ici je vais créer une base de données que je vais nommer `toto_wordpress`.

Étape 9 : Envoi des fichiers de WordPress sur le serveur via SFTP

Connectez-vous en SFTP à votre serveur et envoyez les fichiers composants WordPress dans le dossier précédemment créé.
Pour reprendre mon exemple, c'est donc dans le dossier `/var/www/site-de-toto` que je vais envoyer les fichiers.

Voilà, il ne vous reste plus qu'à vous rendre à l'adresse de votre site et de configurer WordPress.