

# Documentation sur la création d'un serveur web

---

[Retour à toute les documentations](#)

## Règles

- "Saisie utilisateur"
- 'Élément cliquable/sélectionnable'
- Nom de fichier, dossier ou autre
- <Élément à remplacer>

lien, raccourci clavier et phrase de demande de saisie

commande, extrait code et extrait de fichier

## Table des matières

---

- [Documentation sur la création d'un serveur web](#)
  - [Règles](#)
  - [Table des matières](#)
  - [Préambule](#)
  - [Récupération du matériel](#)
  - [Installation du système d'exploitation](#)
  - [Installation et configuration des outils](#)
    - [SSH](#)
    - [Apache](#)
      - [Installation de Apache](#)
      - [Création et déploiement d'un site web avec Apache](#)
    - [PHP](#)
    - [MySQL](#)
    - [Composer](#)
  - [Déploiement d'un projet Laravel](#)
  - [Licence](#)

## Préambule

- Source

[Playlist Youtube de Grafikart](#)

Certains passages de la playlist de Grafikart sont obsolètes, j'ai donc pris d'autre source pour les parties qui ne fonctionnaient pas. Les autres sources seront indiquées de la même façon que celle-ci dans les parties concernées.

Dans ce tutoriel, je vais installer un serveur web sur un ordinateur, de l'installation du système d'exploitation à la configuration des outils nécessaires pour un serveur web.

## Récupération du matériel

Vous pouvez créer un serveur avec n'importe quel ordinateur. Personnellement, j'utilise un ancien ordinateur de bureau qui à une dizaine d'années.

## Installation du système d'exploitation

Je vais utiliser Debian server pour ce tutoriel.

- Télécharger l'image ISO de Debian server sur le site officiel de Debian.

<https://www.debian.org/>

- Debian ne possède pas d'image particulières pour les serveurs, il faut donc télécharger l'image de l'installation de base de Debian. Pendant l'installation, nous ne sélectionnerons pas d'environnement graphique puisque nous n'en avons pas besoin pour un serveur.
- Démarrer l'ordinateur sur la clé USB.
- Sélectionner l'image ISO de Debian server.
- Sélectionner la langue.
- Entrer le nom de la machine
- Entrer le domaine (laisser vide si vous n'en avez pas). Si vous en avez un c'est ici qu'il entrer votre nom de domaine.
- Entrer le mot de passe de l'utilisateur **root**.
- Confirmer le mot de passe.
- Entrer le nom de l'utilisateur courant.
- Entrer le mot de passe de l'utilisateur courant.
- Confirmer le mot de passe.
- Sélectionner le partitionnement du disque dur. // TODO : Ajouter les détails
- Sélectionner le pays pour le miroir des paquets.
- Sélectionner le miroir des paquets **deb.debian.org**.
- Entrer le proxy (laisser vide si vous n'en avez pas).
- Sélectionner si vous voulez participer à l'envoi de données anonymes.
- Sélectionner les paquets à installer.
  - Serveur SSH
  - utilitaires usuels du système
- Installer le chargeur d'amorçage GRUB. (Si vous avez déjà GRUB, il ne vous sera pas demandé de l'installer)

- Cliquer sur **continuer** pour redémarrer l'ordinateur.
- Cliquer sur **Debian/GNU** pour démarrer le système.
- Entrer le nom d'utilisateur créé précédemment.
- Entrer le mot de passe.

Vous avez maintenant un serveur Debian fonctionnel sur lequel vous êtes connecté en tant qu'utilisateur courant.

- Passer en mode super utilisateur.

```
su root
```

- Mettre à jour les paquets.

```
apt update
```

- Installer sudo.

```
apt install sudo
```

- Ajouter l'utilisateur courant au groupe sudo.

```
usermod -aG sudo <nom_utilisateur>
```

- Se connecter en tant qu'utilisateur courant.

```
su <nom_utilisateur>
```

- Testez si sudo fonctionne.

```
sudo date
```

- Résultat attendu : **la date actuelle**

## Installation et configuration des outils

### SSH

- Changer la configuration de SSH

```
sudo nano /etc/ssh/sshd_config
```

- Modifier la ligne **Port** pour changer le numéro de port par défaut.
  - Vous pouvez choisir presque n'importe quel numéro de port entre 1024 et 65535.
- Modifier la ligne **PermitRootLogin** pour qu'elle soit égale à **no**.

```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

...
```

```
Port <port>

...

PermitRootLogin no

...
```

- Sauvegarder et quitter.
- Redémarrer le service SSH.

```
sudo service ssh restart
```

- Créer une clé SSH **sur votre ordinateur**.

```
ssh-keygen -t rsa -b 4096 -C "<email>"
```

- Entrer le nom du fichier de la clé (laisser vide pour utiliser le nom par défaut).
- Entrer une passphrase (optionnel mais plus sécurisé).
- Confirmer la passphrase.
- Connectez-vous au serveur en SSH.

```
ssh <nom_utilisateur>@<adresse_ip> -p <port>
```

- Entrer le mot de passe de l'utilisateur.
- Créer le dossier **.ssh** dans le dossier de l'utilisateur.

```
mkdir -m 700 ~/.ssh
```

- Créer le fichier **authorized\_keys** dans le dossier **.ssh**.

```
touch ~/.ssh/authorized_keys
```

- Copier la clé publique de votre ordinateur dans le fichier **authorized\_keys**.
- Modifier les permissions du fichier **authorized\_keys**.

```
chmod 600 ~/.ssh/authorized_keys
```

## Apache

- Source

<https://friendhosting.net/en/blog/install-apache-on-debian-11.php> <https://youtu.be/arVwa7jvp5M?si=MwPq8noTlilgu2iV>

Je choisis d'installer Apache plutôt que Nginx car c'est un peu plus simple à configurer et c'est plus modulaire. Nginx est plus performant mais mon serveur n'est pas destiné à être utilisé par des milliers de personnes.

## Installation de Apache

- Installer Apache

```
sudo apt install apache2
```

- Vérifier si Apache est bien installé.

```
sudo service apache2 status
```

- Ajouter votre utilisateur au groupe `www-data`.

```
sudo usermod -a -G www-data <nom_utilisateur>
```

- Changer les permissions du dossier `/var` si vous le souhaitez.

```
sudo chown -R www-data:www-data /var
```

## Création et déploiement d'un site web avec Apache

- Source

<https://friendhosting.net/en/blog/install-apache-on-debian-11.php>

- Créer un dossier pour le site web.

```
sudo mkdir /var/www/<nom_site>
```

- Changer les permissions et le groupe du dossier.

```
sudo chown -R www-data:www-data /var/www/<nom_site>
```

- Créer un fichier `index.php` dans le dossier du site web.

```
sudo nano /var/www/<nom_site>/index.php
```

- Ajouter le code suivant.

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

- Sauvegarder et quitter.

- Créer un fichier de configuration pour le site web.

```
sudo nano /etc/apache2/sites-available/001-<nom_site>.conf
```

- Ajouter le code suivant.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/<nom_site>
    ServerName <nom_site>
```

```

ServerAlias www.<nom_site>

<Directory /var/www/<nom_site>>
    Options +Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

- Remplacer **<nom\_site>** par le nom du site web.
  - **ServerAdmin** : Adresse email de l'administrateur du site.
  - **DocumentRoot** : Chemin du dossier du site web.
  - **ServerName** : Nom du site web.
  - **Attention** : Sur un site en production, il faudra changer **+Indexes** par **-Indexes** pour désactiver l'indexation des dossiers.
  - *Pour plus de détails sur les informations du fichier de configuration, voir la source et/ou la documentation officielle d'Apache.*
- Sauvegarder et quitter.

- Activer le site web.

```
sudo a2ensite 001-<nom_site>.conf
```

- Désactiver le site par défaut.

```
sudo a2dissite 000-default.conf
```

- Redémarrer Apache.

```
sudo service apache2 restart
```

## PHP

- Installer PHP ainsi que les extensions nécessaires.

```
sudo apt install php-common php-cli php-mysql php-curl php-xml php-json php-dev php-pear php-gd
```

- **php-common** : Fournit les fichiers communs pour la dernière version stable de PHP
  - **php-cli** : Fournit l'interface de ligne de commande pour PHP
  - **php-mysql** : Fournit un module de support pour l'extension MySQL de PHP
  - **php-curl** : Fournit un module de support pour l'extension CURL de PHP
  - **php-xml** : Fournit un module de support pour l'extension XML de PHP
  - **php-json** : Fournit un module de support pour l'extension JSON de PHP
  - **php-dev** : Fournit les fichiers de développement pour la dernière version stable de PHP
  - **php-pear** : Fournit un gestionnaire de paquets pour PHP
  - **php-gd** : Fournit un module de support pour l'extension GD de PHP qui permet de manipuler dynamiquement des images
- Installer le module PHP pour Apache.

```
sudo apt install libapache2-mod-php
```

- Activer des module PHP comme GD

```
sudo vim /etc/php/8.2/apache2/php.ini
sudo vim /etc/php/8.2/cli/php.ini
```

- Modifier la ligne `;extension=gd` pour qu'elle soit égale à `extension=gd`.

```
;extension=curl
;extension=ffi

...

extension=gd

...
```

## MySQL

- Source

<https://docs.vultr.com/how-to-install-mysql-on-debian-12>

- Télécharger le paquet d'installation de MySQL

```
wget https://dev.mysql.com/get/mysql-apt-config_0.8.33-1_all.deb
```

- Vérifier la version la plus récente sur le site de MySQL.

<https://dev.mysql.com/downloads/repo/apt/>

- Installer le paquet

```
sudo dpkg -i mysql-apt-config_0.8.33-1_all.deb
```

- Mettre à jour les paquets

```
sudo apt install mysql-server -y
```

- Lancer l'installation sécurisé de MySQL

```
sudo mysql_secure_installation
```

- Entrer le mot de passe de l'utilisateur `root` de MySQL.
- Activer `VALIDATE PASSWORD COMPONENT` et choisir le niveau de sécurité du mot de passe.
- Changer le mot de passe de l'utilisateur `root` de MySQL si vous le souhaitez.
- Supprimer les utilisateurs anonymes.
- Désactiver l'accès à distance à l'utilisateur `root` de la base de données.
- Supprimer la base de données de test.
- Recharger les privilèges pour que les modifications prennent effet.

- Vérifier si MySQL est bien installé et lancé (il doit être actif).

```
sudo systemctl status mysql
```



- Se connecter à MySQL en tant que root.

```
sudo mysql -u root -p
```

- Créer un nouvel utilisateur.

```
CREATE USER 'db_user'@'localhost' IDENTIFIED BY 'Strong@@password123';
```

- Donner les droits à l'utilisateur au base de données et tables que vous souhaitez.

```
GRANT ALL PRIVILEGES ON <database>.<table> TO 'db_user'@'localhost' WITH GRANT OPTION;
```

- Remplacer `<database>` par le nom de la base de données (ou `*` pour toutes les bases de données).
- Remplacer `<table>` par le nom de la table (ou `*` pour toutes les tables).

- Mettre à jour les privilèges.

```
FLUSH PRIVILEGES;
```

- Quitter MySQL.

```
exit
```

- Tester la connexion à la base de données.

```
mysql -u db_user -p
```

- Entrer le mot de passe de l'utilisateur.
- Afficher les bases de données.

```
SHOW DATABASES;
```

- Quitter MySQL.

```
exit
```

## Composer

- [Installation de la dernière version de Composer](#)

## Déploiement d'un projet Laravel

- Source

```
https://www.gekkode.com/developpement/installation-de-laravel-et-configuration-apache/  
https://codewithsusan.com/notes/multiple-laravel-apps-on-apache-server
```

- Créer un nouveau projet Laravel

```
composer create-project --prefer-dist laravel/laravel <nom_projet>
```

- En cas de problème de droits, vous pouvez soit ajouter **sudo** devant la commande ou lancez la commande dans votre répertoire personnel (`/home/${USER}`) puis déplacer le projet dans le dossier `/var/www/`
- Créer un fichier **.env** à la racine du projet Laravel

```
cp .env.example .env
```

- Modifier le fichier **.env** pour qu'il corresponde à votre base de données

```
...  
  
DB_CONNECTION=mysql  
DB_HOST=<adresse_ip>  
DB_PORT=<port>  
DB_DATABASE=<nom_base_de_données>  
DB_USERNAME=<nom_utilisateur>  
DB_PASSWORD=<mot_de_passe>  
  
...
```

- Modifier les permissions du dossier du projet Laravel

```
sudo chown -R www-data:www-data /var/www/<nom_projet>
```

- Modifier les permissions du dossier de **stockage**

```
sudo chmod -R 777 /var/www/<nom_projet>/storage
```

- Modifier les permissions du dossier de **bootstrap/cache**

```
sudo chmod -R 777 /var/www/<nom_projet>/bootstrap/cache
```

- Mettre à jour les dépendances de Composer

```
composer update  
npm update
```

- Installer les dépendances du projet Laravel

```
composer install  
npm install
```

- Générer une clé pour le projet Laravel

```
php artisan key:generate
```

- Migrer les tables de la base de données

```
php artisan migrate
```

- Activer le module Apache rewrite qui permet de gérer la réécriture d'URL

```
sudo a2enmod rewrite
```

- Créer un fichier de configuration pour le site web Laravel

```
sudo nano /etc/apache2/sites-available/002-<nom_projet>.conf
```

- Changer les permissions et le groupe du dossier

```
sudo chown -R www-data:www-data /var/www/<nom_projet>
```

- Créer un fichier de configuration pour le site web

```
sudo nano /etc/apache2/sites-available/001-<nom_projet>.conf
```

- Ajouter le code suivant

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/<nom_projet>/public
    ServerName <nom_projet>
    ServerAlias www.<nom_projet>

    <Directory /var/www/<nom_projet>/public>
        Options +Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Remplacer **<nom\_projet>** par le nom du site web
- **ServerAdmin** : Adresse email de l'administrateur du site
- **DocumentRoot** : Chemin du dossier du site web
- **ServerName** : Nom du site web
- **Attention** : Sur un site en production, il faudra changer **+Indexes** par **-Indexes** pour désactiver l'indexation des dossiers
- *Pour plus de détails sur les informations du fichier de configuration, voir la source et/ou la documentation officielle d'Apache*

- Sauvegarder et quitter

- Dans le cas où vous avez plusieurs sites web, vous pouvez créer un fichier de configuration pour chaque site web en incrémentant le numéro du fichier de configuration.

- Vous devrez également ajouter en première ligne du fichier de configuration apache **listen <port>** pour chaque site web ainsi que **<VirtualHost \*:<port>>** à la place de **<VirtualHost \*:80>**.

- Activer le site web

```
sudo a2ensite 002-<nom_projet>.conf
```

- Désactiver le site actuellement actif

```
sudo a2dissite 001-<nom_site>.conf
```

- Redémarrer Apache

```
sudo service apache2 restart
```

## Licence

doc\_creation\_serveur.md

Copyright (C) 2024 Floris Robart

Authors: Floris Robart

This program is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301, USA.

---

[Retour à toute les documentations](#)