

Installation d'un LAMP sur Debian

Installer Apache sous Debian 11

Tout d'abord faire la config IP

`nano /etc/network/interfaces`

Et configurer votre machine selon votre besoin :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens192
iface ens192 inet static
    address 192.168.100.14/24
    gateway 192.168.100.1
    dns-nameservers 192.168.100.11 1.1.1.1
    dns-domain it-connect.local
```

`apt-get update`

Ensuite, on installe le paquet "apache2" afin d'obtenir la dernière version d'Apache 2.4.

`apt-get install -y apache2`

Pour qu'Apache démarre automatiquement en même temps que Debian, saisissez la commande ci-dessous (même si normalement c'est déjà le cas) :

`systemctl enable apache2`

Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.

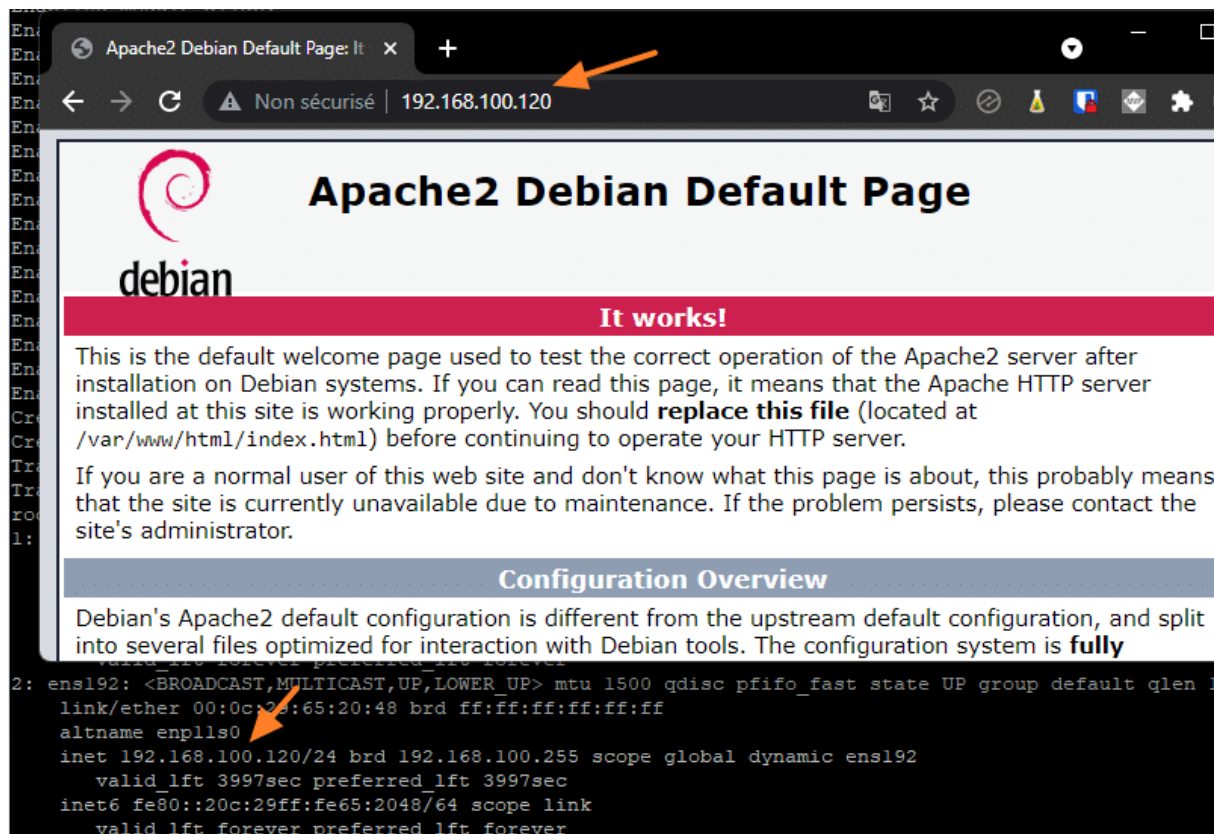
Executing: /lib/systemd/systemd-sysv-install enable apache2

À la suite de l'installation du paquet, le serveur Apache démarre directement. On devrait pouvoir accéder à sa page par défaut. Pour cela, il suffit de récupérer l'adresse IP du serveur :

`ip address`

Puis, à l'aide d'une machine équipée d'un navigateur, on peut accéder à notre serveur Apache :

`http://192.168.100.120`



a2enmod rewrite

L'occasion de découvrir **la commande "a2enmod" qui sert à activer un module**. A l'inverse, **la commande "a2dismod" sert à désactiver un module**.

Activons trois : autres modules :

- **"deflate"** pour la gestion de la compression, notamment en gzip, pour utiliser la mise en cache des pages sur votre site
- **"headers"** afin de pouvoir agir sur les en-têtes HTTP
- **"ssl"** pour gérer les certificats SSL et donc l'utilisation du protocole HTTPS

a2enmod deflate

a2enmod headers

a2enmod ssl

Après avoir activé ou désactivé un module, ou modifié la configuration d'Apache, il faut redémarrer le service apache2 :

systemctl restart apache2

Information : Le fichier de configuration d'Apache 2 est le suivant :

/etc/apache2/apache2.conf

Si vous souhaitez mettre en place l'authentification basique sur votre site, vous avez besoin de l'outil "htpasswd" inclus dans le paquet "apache2-utils" (comme d'autres outils). Vous pouvez l'installer à tout moment d'une simple commande :

sudo apt-get install -y apache2-utils

Installer PHP sous Debian 11

```
apt-get install -y php
```

Avant d'aller plus loin, nous allons installer quelques paquets supplémentaires pour compléter l'installation de PHP sur notre serveur. Par exemple, pour permettre les interactions entre PHP et notre instance MariaDB.

```
apt-get install -y php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath
```

Maintenant, pour nous assurer que notre moteur de script PHP est bien actif, nous allons créer un fichier "*phpinfo.php*" (ou un autre nom) à la racine de notre site Web :

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

Dans ce fichier, indiquez le code suivant :

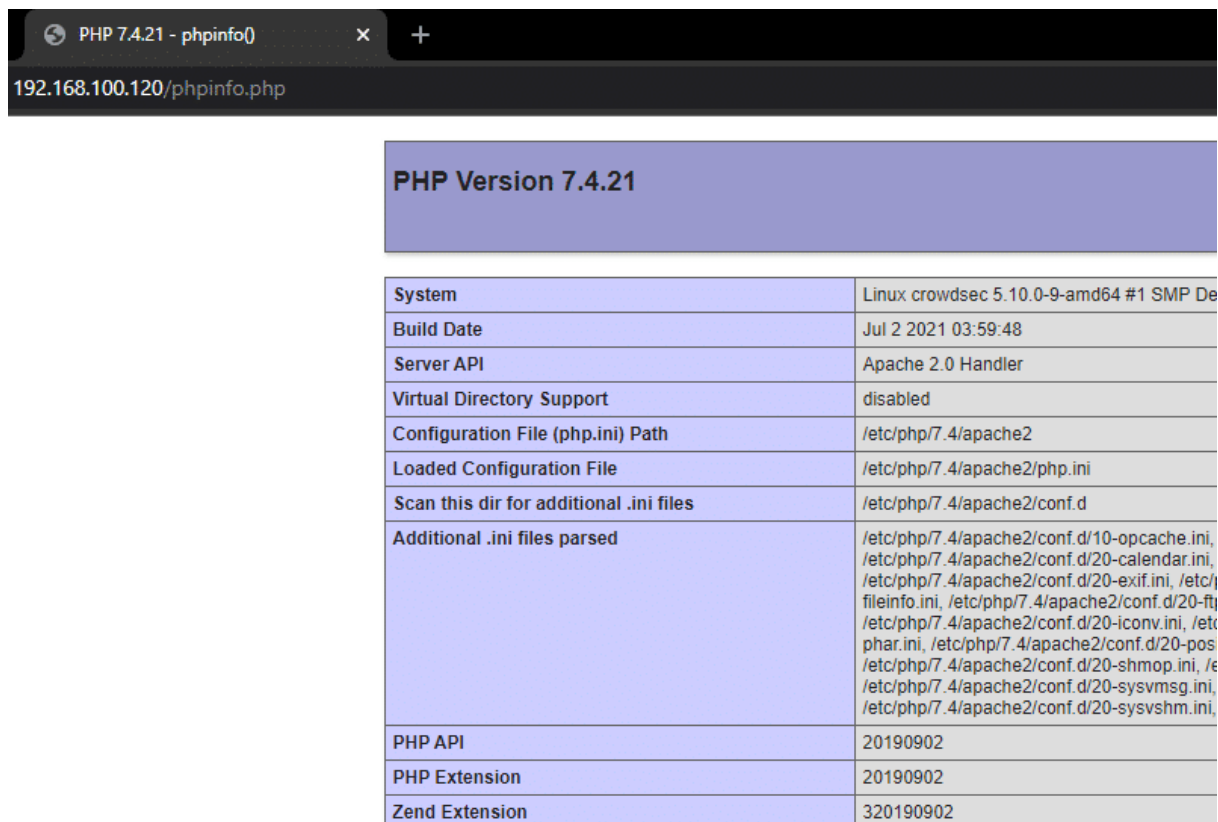
```
<?php
```

```
phpinfo();
```

```
?>
```

Elle sera accessible à partir de cette adresse :

<http://192.168.100.120/phpinfo.php>



PHP Version 7.4.21	
System	Linux crowdsec 5.10.0-9-amd64 #1 SMP De
Build Date	Jul 2 2021 03:59:48
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/ fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/ phar.ini, /etc/php/7.4/apache2/conf.d/20-pos /etc/php/7.4/apache2/conf.d/20-shmop.ini, /e /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini,
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902

Installer MySQL/MariaDB sous Debian 11

Pour installer MariaDB sous Debian 11, voici la commande à exécuter :

```
apt-get install -y mariadb-server
```

mariadb-secure-installation

En résumé, vous allez pouvoir définir un mot de passe pour le compte "root" de MariaDB, empêcher les connexions distantes sur votre instance à l'aide du compte "root", empêcher les connexions anonymes et supprimer la base de test.

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n

... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] Y

New password: *****

Re-enter new password: *****

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for

them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n] y

... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y

... Success!

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] y

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] y

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Avant de passer à la suite, vérifiez que vous parvenez à vous connecter à votre instance MariaDB :

```
mariadb -u root -p
```

Saisissez le mot de passe "root". Ensuite, vous avez accès à la console MariaDB / MySQL. Vous pouvez saisir vos requêtes SQL ici. Par exemple, pour lister les bases de données de votre instance :

```
show databases;
```

```
exit ;
```

Après un changement de configuration de MariaDB, vous devez redémarrer le service :

```
systemctl restart mariadb
```