

Contexte GSB

Le laboratoire Galaxy Swiss Bourdin est issu de la fusion entre le géant américain Galaxy et le conglomerat européen Swiss Bourdin, lui-même déjà union de trois petits laboratoires. En 2009, les deux géants pharmaceutiques unissent leurs forces pour créer un leader de ce secteur industriel. L'entité GSB Europe a établi son siège administratif à Paris. Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis. La France a été choisie comme témoin pour l'amélioration du suivi de l'activité de visite.

Besoins :

Depuis la réorganisation de GSB l'entreprise fonctionne bien et a connu une hausse de son activité. Suite à cela Galaxy Swiss Bourdin a décidé de mettre en place

Solution

Installation, configuration et sécurisation d'un serveur LAMP

I. Présentation

C'est quoi un serveur LAMP ? Il s'agit d'un serveur qui s'appuie sur 4 composants : **L pour Linux**, c'est-à-dire le système d'exploitation (Debian, dans notre cas), **A pour Apache**, c'est-à-dire le serveur Web, **M pour MySQL/MariaDB**, c'est-à-dire le système de gestion de bases de données, et **P pour PHP**, c'est-à-dire le moteur de scripts.

Pour suivre ce tutoriel, vous avez besoin d'une machine sous Debian, ou une distribution basée sur Debian.

- Pour installer un serveur LAMP sur WSL (Linux sur Windows), suivez ce tutoriel :
Tutoriel - LAMP sur WSL

II. Serveur LAMP sous Debian 11

A. Installer Apache sous Debian 11

On commence par mettre à jour le cache des paquets :

```
sudo apt-get update
```

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

```
sudo 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) :

```
sudo systemctl enable apache2
```

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

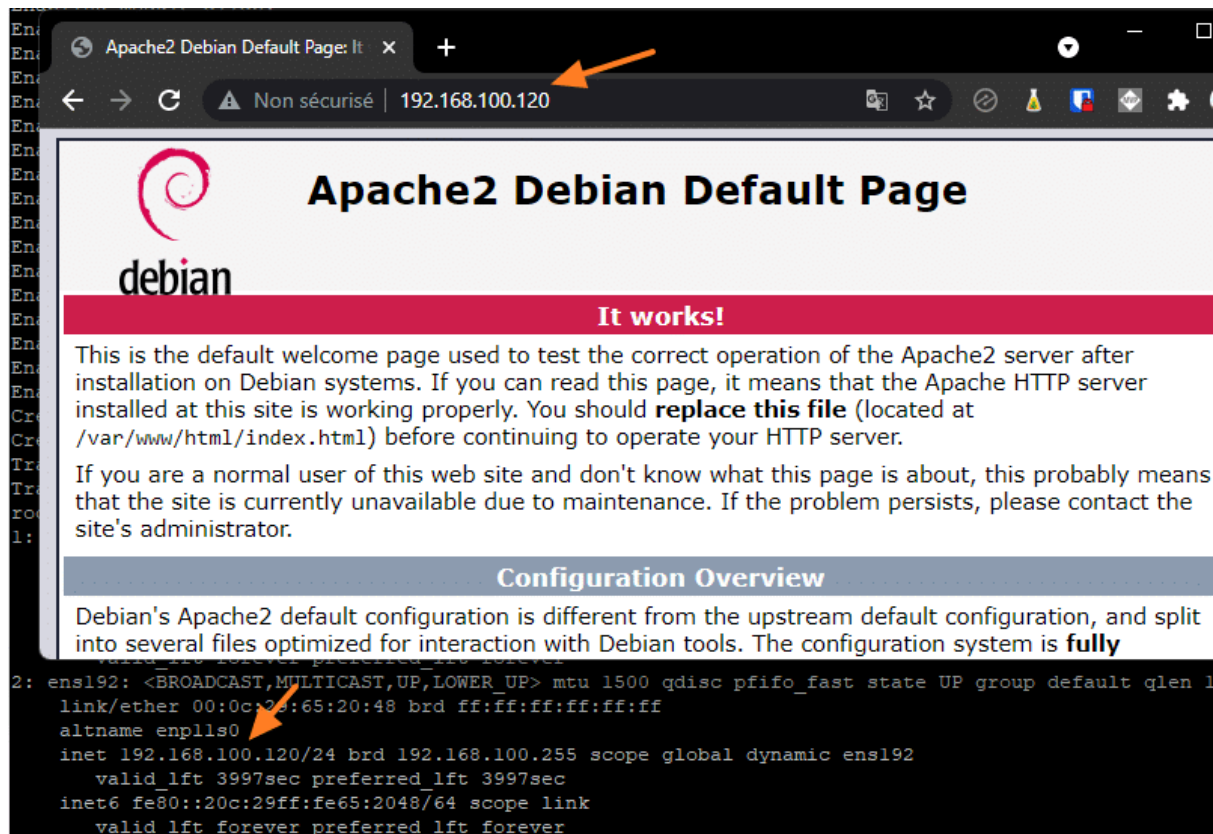
```
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

Suite à 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
```



Apache en ligne, sous Debian 11

Pour visualiser la version d'Apache que vous venez d'installer, c'est tout simple : exécutez la commande suivant :

```
sudo apache2ctl -v
Server version: Apache/2.4.51 (Debian)
Server built: 2021-10-07T17:49:44
```

Apache 2.4.51 est la dernière version d'Apache au moment où j'écris cet article.

Avant d'aller plus loin, je vous recommande d'**activer quelques modules d'Apache qui sont indispensables**, notamment pour faire tourner un site Internet. Commençons par **le module utilisé pour la réécriture d'URL** :

```
sudo 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

```
sudo a2enmod deflate
sudo a2enmod headers
sudo a2enmod ssl
```

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

```
sudo systemctl restart apache2
```

Où se situent la configuration d'Apache et des sites dans tout ça ?

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

```
/etc/apache2/apache2.conf
```

Dans un premier temps, il peut servir à configurer Apache pour ne pas afficher le numéro de version sur les pages d'erreurs. Même si cette option est gérable aussi dans le fichier *"/etc/apache2/conf-enabled/security.conf"*, c'est au choix.

Note : pour la configuration qui concerne PHP, le fichier de configuration est différent : *"/etc/php/7.4/apache2/php.ini"*

Tandis que pour **déclarer les hôtes virtuels, en anglais "Virtual hosts"**, ce qui correspond aux différents sites hébergés par Apache (oui, un serveur Apache peut gérer plusieurs sites indépendamment), il faudra s'intéresser à ces deux dossiers :

- Dossier qui contient les fichiers de configuration des sites disponibles :
/etc/apache2/sites-available/
- Dossier qui contient les fichiers de configuration (via un lien symbolique), des sites actifs : **/etc/apache2/sites-enabled**

Par défaut, nous accédons à la page d'accueil d'Apache grâce à l'hôte virtuel déclaré dans le fichier `"/etc/apache2/sites-enabled/000-default.conf"`, qui écoute sur le port 80 (HTTP) et dont la racine est le dossier `"/var/www/html"`.

Je vous invite à lire mon tutoriel dédié à la configuration d'un Virtual Host pour en savoir plus :

- Tutoriel - Apache - Déclarer un Virtual Host

Enfin, 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
```

B. Installer PHP sous Debian 11

PHP va venir se greffer sur notre serveur Apache, comme une extension, afin de pouvoir traiter les scripts intégrés aux pages `".php"`. Afin d'y aller progressivement, installons le paquet `"php"` en lui-même :

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

On peut voir que cette commande va installer une multitude de paquets :

```
libapache2-mod-php7.4  libsodium23  php-common  php7.4  php7.4-cli  
php7.4-common  php7.4-json  php7.4-opcache  php7.4-readline
```

C'est très bien, nous avons quelques modules de base indispensables et `"libapache2-mod-php7.4"` qui permet l'intégration avec Apache.

Actuellement, c'est PHP 7.4 qui est dans les dépôts de Debian, même si PHP 8 est déjà disponible, toutes les applications ne sont pas encore compatibles. **Il faut savoir que le support de PHP 7.4 assure les mises à jour de sécurité jusqu'au 28 novembre 2022.** Ce qui laisse un peu de temps, mais il faut garder en tête qu'il faudra envisager de passer sur PHP 8.

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.

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

Suite à cette installation, je vous invite à vérifier quelle version de PHP vous venez d'installer. Exécutez la commande suivante :

```
php -v  
PHP 7.4.21 (cli) (built: Jul 2 2021 03:59:48) ( NTS )
```

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 :

```
sudo 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

Cette page donne énormément d'informations sur toute la configuration de PHP et de notre serveur Apache. Il est fortement recommandé de la mettre en place seulement quand c'est nécessaire. **Autrement dit, vous ne devez pas laisser cette page accessible par n'importe qui.**

C. Installer MySQL/MariaDB sous Debian 11

MariaDB est un fork communautaire de MySQL et il présente l'avantage d'être open source et sous licence GPL, à la différence de MySQL qui est un logiciel propriétaire de chez Oracle, mais qui reste gratuit malgré tout. Il y a un excellent suivi pour MariaDB et c'est réellement un système très performant, vous pouvez miser sur ce composant sans aucun problème !

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

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

Suite à l'installation, je vous invite à exécuter le script "**mariadb-secure-installation**" afin de sécuriser un minimum votre installation de MariaDB.

```
sudo 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.

Voilà, l'interrogatoire est terminé.

Pour obtenir le numéro de version de MariaDB, on peut utiliser cette commande :

```
mariadb -V
mariadb Ver 15.1 Distrib 10.5.12-MariaDB, for debian-linux-gnu (x86_64)
using EditLine wrapper
```

Ou celle-ci en consultant le gestionnaire de paquets Aptitude (*apt*) :

```
apt policy mariadb-server
mariadb-server:
  Installé : 1:10.5.12-0+deb11u1
  Candidat : 1:10.5.12-0+deb11u1
  Table de version :
  *** 1:10.5.12-0+deb11u1 500
  500 http://ftp.fr.debian.org/debian bullseye/main amd64 Packages
  100 /var/lib/dpkg/status
```

Il est à noter que même si l'on a installé MariaDB, on peut utiliser la commande "mysql", notamment pour afficher le numéro de version avec "mysql -V" ou ouvrir une console MySQL/MariaDB.

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

```
sudo 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;
```

```
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 58
Server version: 10.5.12-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database           |
+-----+
| information_schema |
| mysql               |
| performance_schema |
+-----+
3 rows in set (0.001 sec)

MariaDB [(none)]> █
```

Première connexion à MariaDB en ligne de commande

Pour sortir de la console, saisissez la commande suivante :

```
exit
```

Il faudra revenir dans cette console lorsque vous allez déployer votre application sur votre serveur LAMP, par exemple WordPress, NextCloud, etc.... Afin de créer une base de données dédiée et un utilisateur dédié à cette application. Une alternative consiste à déployer PhpMyAdmin sur son serveur dans le but d'administrer MariaDB à partir d'une interface Web.

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

```
systemctl restart mariadb
```

III. Conclusion

Voilà, votre serveur LAMP est installé ! Pour la suite de la configuration, cela dépend de l'application que vous souhaitez déployer, ou peut-être même qu'il s'agit d'un projet que vous avez vous-même développé.

Généralement, on commence par créer un nouvel hôte virtuel sur Apache pour accueillir les sources d'installation de l'application. Ensuite, on crée une base de données dédiée à cette application, avec son propre utilisateur (qui aura les droits seulement sur cette base), et on lance l'installation.