

RAPPORT :
CONFIGURATION DU SERVEUR WEB ET
DE LA BASE DE DONNEES

PREREQUIS :	3
STACK TECHNIQUE :	3
SERVEUR WEB :	4
Installation d'apache	4
Installation de PHP	4
Configuration de la page web	5
BASE DE DONNEES :	6
Installation d'apache	6
Installation de PHP	6
Installation de MariaDB	7
Installation de phpMyAdmin	8
Configuration du firewall :	8
FICHIERS DE CONFIGURATIONS :	9
SOURCES :	11

PREREQUIS :

- 2 machines virtuelles, voici les spécifications utilisées :
 - RAM : 2 GB
 - Hard Disk : 20 GB
 - OS : Debian 12
 - IP fixe paramétrés au préalable (tuto [ici](#)) :
 - 192.168.45.10 pour la machine host serveur WEB.
 - 192.168.45.20 pour la machine host base de données.
- Postman
- Connaissances basiques en infrastructure informatique

STACK TECHNIQUE :

Cette installation est fondée sur la LAMP Stack :

- Linux (Debian étant une distribution Linux)
- Apache : Serveur HTTP recevant les requêtes
- MariaDB : Serveur de la base de données
- PHP : Langage de script, utilisé pour renvoyer les données souhaitées dans notre cas

SERVEUR WEB :

VM 192.168.45.10

⚠ Par soucis de clarté, nous allons considérer que toutes les commandes sont saisies par root. Si ce n'est pas votre cas, pensez à inclure le mot-clé "sudo" si nécessaire

Installation d'apache

Avant toutes chose, assurez-vous que le cache des paquets soit bien à jour :

```
sudo apt update
```

Ensuite, installez le paquet "apache2" :

```
Apt install apache2
```

Le serveur Apache est maintenant disponible sur l'adresse de la machine virtuelle <http://192.168.45.10> .

Installation de PHP

Installation du paquet php ainsi que du module php-mysql dont on aura besoin pour les requêtes vers la BDD :

```
apt install php php-mysql
```

Vous pouvez vérifier la version de php avec la commande suivante :

```
php -v
```

Configuration de la page web

Création d'un dossier personnalisé où nous mettrons les scripts php faisant fonctionner l'API (scripts disponibles dans le dossier "endpoint" sur le repertoire github) :

```
mkdir /var/www/e-commerce
```

Copiez ensuite les scripts dans le dossier puis pensez à changer le propriétaire du dossier à www-data. Cela permettra à Apache de le parcourir et de l'afficher.

```
chown -R www-data: /var/www/e-commerce
```

Créez le fichier /etc/apache2/sites-available/e-commerce.conf puis copiez-y la configuration suivante (également disponible sur le github) :

```
1 <VirtualHost *:80>
2     ServerName ecommerce.com
3     DocumentRoot /var/www/e-commerce
4
5     <Directory /var/www/e-commerce>
6         AllowOverride All
7         Require all granted
8     options indexes followsymlinks
9     </Directory>
10
11     ErrorLog ${APACHE_LOG_DIR}/e-commerce_error.log
12     CustomLog ${APACHE_LOG_DIR}/e-commerce_access.log combined
13 </VirtualHost>
14
```

Remplacement de la configuration à prendre en compte :

```
a2ensite e-commerce.conf
```

```
A2dissite 000-default.conf
```

```
Systemctl restart apache2
```

Les endpoints sont à présent paramétré.

BASE DE DONNEES :

VM 192.168.45.10

Bien que l'installation de MariaDB en standalone pourrait en théorie suffire, Nous avons décidé d'héberger phpmyadmin afin de faciliter l'administration de la base.

Il faudra donc reproduire les étapes d'installation d'Apache et de php.

Installation d'apache

Avant toutes chose, assurez-vous que le cache des paquets soit bien à jour :

```
sudo apt update
```

Ensuite, installez le paquet "apache2" et le module PHP d'apache :

```
apt install apache2
```

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

Le serveur Apache est maintenant disponible sur l'adresse de la machine virtuelle <http://192.168.45.20> .

Installation de PHP

Installation du paquet php ainsi que des module requis pour le bon fonctionnement de phpmyadmin (liste juste [ici](#)) :

```
apt install php php-mysql
```

```
apt install php8.2-
```

```
{bz2,mbstring,zip,gd,curl,xml,common,opcache,imagick}
```

Vous pouvez vérifier la version de php avec la commande suivante :

```
php -v
```

Une fois les étapes reproduites, il nous faudra installer MariaDB.

Installation de MariaDB

Installation du paquet mariaDB :

```
apt install mariadb-server
```

Vous pouvez vérifier la version de mariadb avec la commande suivante :

```
mariadb -V
```

Installation de phpMyAdmin

Installation comme pour chaque paquet :

```
apt install phpmyadmin
```

Une fois la commande saisie, il vous sera demandé de sélectionner un serveur web, sélectionnez apache2.

Il vous sera également proposé de configurer une base de données “test”, acceptez pour plus de facilité. Cette installation créera automatiquement le fichier de configuration “etc/apache2/conf-available/phpmyadmin.conf”.

Configuration du firewall :

Pour gérer le firewall, nous utilisons le paquet ufw :

```
apt install ufw
```

Authorization pour Apache :

```
ufw allow "WWW Full"
```

Authorisation pour le serveur web de communiquer avec la BDD :

```
ufw allow from 192.168.45.10 to any port 3306
```


FICHIERS DE CONFIGURATIONS :

Configurations des endpoints :

/etc/apache2/sites-enabled/e-commerce.conf

```
1 <VirtualHost *:80>|
2     ServerName ecommerce.com
3     DocumentRoot /var/www/e-commerce
4
5     <Directory /var/www/e-commerce>
6         AllowOverride All
7         Require all granted
8     options indexes followsymlinks
9     </Directory>
10
11     ErrorLog ${APACHE_LOG_DIR}/e-commerce_error.log
12     CustomLog ${APACHE_LOG_DIR}/e-commerce_access.log combined
13 </VirtualHost>|
14
```

Configuration phpMyAdmin (généré automatiquement) :

/etc/apache2/conf-enabled/phpmyadmin.conf

```
1 # phpMyAdmin default Apache configuration
2
3 Alias /phpmyadmin /usr/share/phpmyadmin
4
5 <Directory /usr/share/phpmyadmin>
6     Options SymLinksIfOwnerMatch
7     Options Indexes FollowSymLinks MultiViews
8     AllowOverride all
9     Require all granted
10    DirectoryIndex index.php
11
12    # Limit libapache2-mod-php to files and directories necessary by pma
13    <IfModule mod_php7.c>
14        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
15        php_admin_value open_basedir /usr/share/phpmyadmin:/usr/share/doc/phpmyadmin:/etc
16        /phpmyadmin:/var/lib/phpmyadmin:/usr/share/php:/usr/share/javascript/
17    </IfModule>
18
19    # PHP 8+
20    <IfModule mod_php.c>
21        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
22        php_admin_value open_basedir /usr/share/phpmyadmin:/usr/share/doc/phpmyadmin:/etc
23        /phpmyadmin:/var/lib/phpmyadmin:/usr/share/php:/usr/share/javascript/
24    </IfModule>
25 </Directory>
26
27 # Disallow web access to directories that don't need it
28 <Directory /usr/share/phpmyadmin/templates>
29     Require all denied
30 </Directory>
31 <Directory /usr/share/phpmyadmin/libraries>
32     Require all denied
33 </Directory>
34
```

Configuration mariaDB :

/etc/mysql/mariadb.cnf

```
1 # The MariaDB configuration file
2 #
3 # The MariaDB/MySQL tools read configuration files in the following order:
4 # 0. "/etc/mysql/my.cnf" symlinks to this file, reason why all the rest is read.
5 # 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,
6 # 2. "/etc/mysql/conf.d/*.cnf" to set global options.
7 # 3. "/etc/mysql/mariadb.conf.d/*.cnf" to set MariaDB-only options.
8 # 4. "~/.my.cnf" to set user-specific options.
9 #
10 # If the same option is defined multiple times, the last one will apply.
11 #
12 # One can use all long options that the program supports.
13 # Run program with --help to get a list of available options and with
14 # --print-defaults to see which it would actually understand and use.
15 #
16 # If you are new to MariaDB, check out https://mariadb.com/kb/en/basic-mariadb-articles/
17
18 #
19 # This group is read both by the client and the server
20 # use it for options that affect everything
21 #
22 [client-server]
23 # Port or socket location where to connect
24 port = 3306
25 socket = /run/mysqld/mysqld.sock
26
27 # Import all .cnf files from configuration directory
28 !includedir /etc/mysql/conf.d/
29 !includedir /etc/mysql/mariadb.conf.d/
30
31 [mysqld]
32 #log_error = /var/log/mysql/error.log
33 bind-address = 0.0.0.0
34
```

ACLs de la base de données :

```
root@stock-database:~# ufw status
Status: active

To Action From
--
www Full Anywhere
3306 ALLOW 192.168.45.10
1166 ALLOW 192.168.45.10
```

SOURCES :

Documentations officielles :

- <https://docs.phpmyadmin.net/en/latest/setup.html>
- <https://httpd.apache.org/docs/2.4/>
- <https://www.php.net/manual/fr/index.php>
- <https://mariadb.com/kb/en/documentation/>

Installation simplifiée : <https://kifarunix.com/>

Résolution de problème : <https://stackoverflow.com/>