Installation de LAMP

Projet m2l.org

DUMAS Lucie

Table des matières

LAMP 3
Qu'est-ce que LAMP ? 3
Le serveur web 4
SQLite3 6
MariaDB (MySQL) 9
PHPMyAdmin 12
Finalisation 15

LAMP

Qu'est-ce que LAMP ?

LAMP est un ensemble de logiciels libres permettant de construire des serveurs de sites web. L'acronyme LAMP signifie Linux Apache MySQL (ou MariaDB) PHP :

- « Linux » fait référence au système d'exploitation compatible avec cet ensemble de logiciels. Il en existe également pour Windows (WAMP) ou encore Mac (MAMP).
- Apache est un logiciel de serveurs gratuits et open-source permettant la création et la gestion de serveurs web.
- MySQL est un SGBDR, c'est-à-dire un Système de Gestion de Bases de Données Relationnelles open-source.
- PHP (ou PHP Hypertext Preprocessor) est un langage de scripts généraliste et open-source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML.

Le serveur web

Lors de la dernière mission, nous avions installé apache2 afin de créer notre serveur web. Pour plus de détails, nous pouvons nous référer à la mission « Installation de conteneurs LXC ».

Pour cette mission, nous effectuerons nos manipulations majoritairement dans le conteneur web. Nous pouvons vérifier qu'il soit bien installé et lancé grâce aux commandes netstat et systemctl :

```
systemctl status apache2
netstat -nat
```

La commande systemctl status apache 2 nous permet de voir si le service est en cours d'exécution, s'il est à l'arrêt ou s'il rencontre des difficultés à démarrer.

```
root@web:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-01-26 09:18:19 CET; 36min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 30082 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 30086 (apache2)
      Tasks: 11 (limit: 18970)
```

Nous pouvons également voir lorsque nous effectuons cette commande le PID du service. L'identifiant de processus, ou PID (Process Identifier) est un code unique attribué à tout processus lors de son démarrage (uniquement sur les systèmes Unix ou Windows). Le PID attribué à apache2 lors de ce démarrage est 30086. Le PID est aussi trouvable en effectuant la commande netstat -natp.

La commande netstat -nat nous indique tous les ports en mode « Listen » (ou écoute) de notre conteneur. Pour vérifier qu'apache2 soit fonctionnel, nous vérifions que le port 80 soit bien en écoute :

```
root@web:~# netstat -nat
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
```

Pour modifier le port d'écoute d'apache2, nous devons changer le fichier /etc/apache2/ports.conf comme ceci :

```
  GNU nano 5.4                          /etc/apache2/ports.conf *
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 1234

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Si nous redémarrons le service apache et que nous retapons la commande netstat -nat, nous voyons qu'apache n'écoute plus le port 80. Il écoute maintenant le port 1234.

```
root@web:~# systemctl restart apache2
root@web:~# netstat -nat
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale           Adresse distante         Etat
tcp        0      0 0.0.0.0:3306            0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN
tcp6       0      0 :::1234                 :::*                     LISTEN
tcp6       0      0 :::22                   :::*                     LISTEN
```

Pour le service apache2, les fichiers de log sont situés dans le dossier /var/log/apache2

SQLite3

Nous allons maintenant installer SQLite3. Pour ce faire, nous allons mettre à jour nos paquets et télécharger les nouveaux paquets nécessaires :

```
apt update && apt upgrade
apt install sqlite3 php-sqlite3 libapache2-mod-php
```

Voici la liste des modules complémentaires installés :

libapache2-mod-php
libapache2-mod-php7.4
libreadline8
libsodium23
php-common
php-sqlite3
php7.4-cli
php7.4-common
php7.4-json
php7.4-opcache
php7.4-readline
php7.4-sqlite3
psmisc
readline-common
sqlite3

Nous créons un script info.php dans notre dossier /var/www/html qui nous servira à vérifier la librairie PDO pour SQLite3 en accédant à notre site : http://10.31.96.50/info.php

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

pdo_sqlite

PDO Driver for SQLite 3.x	enabled
SQLite Library	3.34.1

Toujours dans le dossier /var/www/html, nous devons créer une base de données correspondant au cahier des charges à l'aide de la commande suivante et du SQL nécessaire :

```
sqlite3 myCDS.db
```

```
# Création des tables

create table artist (art_id INTEGER PRIMARY KEY, art_name TEXT);
create table cd (cd_id INTEGER PRIMARY KEY, art_id INTEGER NOT NULL, cd_title TEXT NOT NULL, cd_date TEXT);

# Insertion des données

insert into artist (art_id,art_name) values (NULL, 'Peter Gabriel');
insert into artist (art_id,art_name) values (NULL, 'Bruce Hornsby');
insert into artist (art_id,art_name) values (NULL, 'Lyle Lovett');
insert into artist (art_id,art_name) values (NULL, 'Beach Boys');
insert into cd (cd_id,art_id,cd_title,cd_date) values (NULL,1,'Us','1992');
insert into cd (cd_id,art_id,cd_title,cd_date) values (NULL,2,'The Way It Is', '1986');
insert into cd (cd_id,art_id,cd_title,cd_date) values (NULL,2,'Scenes from the Southside','1990');
insert into cd (cd_id,art_id,cd_title,cd_date) values (NULL,1,'Security','1990');
insert into cd (cd_id,art_id,cd_title,cd_date) values (NULL,3,'Joshua Judges Ruth','1992');
insert into cd (cd_id,art_id,cd_title,cd_date) values (NULL,4,'Pet Sounds', '1966');
```

Nous pouvons alors effectuer une requête SQL pour vérifier le contenu de notre base de données :

```
sqlite> select * from artist;
1|Peter Gabriel
2|Bruce Hornsby
3|Lyle Lovett
4|Beach Boys
```

```
sqlite> select * from cd;
1|1|Us|1992
2|2|The Way It Is|1986
3|2|Scenes from the Southside|1990
4|1|Security|1990
5|3|Joshua Judges Ruth|1992
6|4|Pet Sounds|1966
```

Nous créons le script PHP bdd.php qui nous permettra d'afficher le contenu de la base :

```
<?php
try {
        $db = new PDO('sqlite :myCDS.db');
        print ("<b>Connecté à la base</b><br><br> ") ;
        print (" <b>Les artistes</b><br>") ;
        $sql = 'SELECT * from artist' ;
        $result = $db->query($sql) ;

        foreach ($result as $row) {
                print ($row['art_id']. ".\t".$row['art_name'].'<br>');
        };

        print ("<br><b>Les albums</b><br>") ;
        $sql = 'SELECT * from cd' ;
        $result = $db->query($sql) ;

        foreach ($result as $row) {
                print ($row['art_id']."\t".$row['cd_title']."\t".$row['cd_date'].'<br>');
        };
        $db = NULL;
}
catch(PDOException $e) {
        print ('Exception : '. $e->getMessage());
]
?>
```

MariaDB (MySQL)

Nous allons maintenant installer MariaDB. Nous mettons à jour nos paquets :

```
apt update && apt upgrade
apt install mariadb-server php-mysql
```

Nous terminons l'installation de MariaDB grâce à l'installation sécurisée. Pour cela, nous entrons la commande :

```
mysql_secure_installation
```

Dans un premier temps, le script nous demande un mot de passe root (par défaut, il n'en mettra aucun) :

```
root@web:~# mysql_secure_installation

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):
```

Nous choisissons ensuite toutes les entrées par défaut pour tous les paramètres suivants :

```
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]
```

```
Enabled successfully!
Reloading privilege tables..
 ... Success!


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

Change the root password? [Y/n]
```

```
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]
```

```
 ... 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]
```

```
 ... 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]
```

Nous allons maintenant créer un nouvel utilisateur. Pour cela, nous commençons par nous connecter au client MariaDB :

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

Nous créons ensuite le compte dba qui aura tous les droits sur les bases de données :

```
# Création de l'utilisateur
create user 'dba'@'localhost' identified by 'drowssap';

# Attribution des droits sur toutes les bases de données
grant all privileges on *.* to 'dba'@'localhost' with grant option;

# Mise à jour des privilèges de la base de données
flush privileges ;
```

```
MariaDB [(none)]> create user 'dba'@'localhost' identified by 'drowssap';
Query OK, 0 rows affected (0,018 sec)

MariaDB [(none)]> grant all privileges on *.* to 'dba'@'localhost' with grant option;
Query OK, 0 rows affected (0,017 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0,001 sec)
```

Nous pouvons nous déconnecter de la base de données avec Ctrl+D et essayer une nouvelle connexion avec l'utilisateur dba :

```
mysql -u dba -p
```

Nous affichons ensuite la liste des bases disponibles :

```
show databases ;
```

```
sio@web:/var/www/html$ mysql -u dba -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 175
Server version: 10.5.18-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 |
| myCDS              |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sio                |
+--------------------+
6 rows in set (0,000 sec)
```

PHPMyAdmin

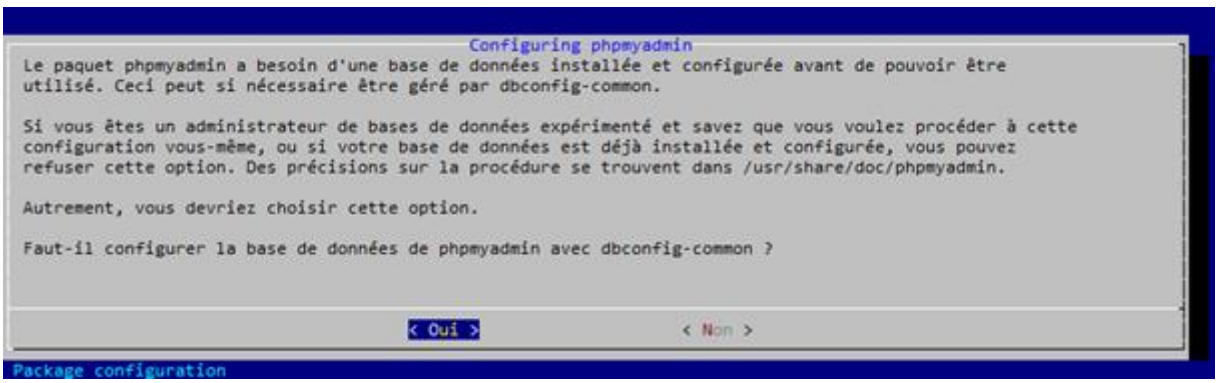
Nous allons maintenant installer PHPMyAdmin. Pour ce faire, nous allons d'abord télécharger les modules PHP nécessaires :

```
apt update && apt upgrade
apt install php-json php-mbstring php-zip php-gd php-xml php-curl
```

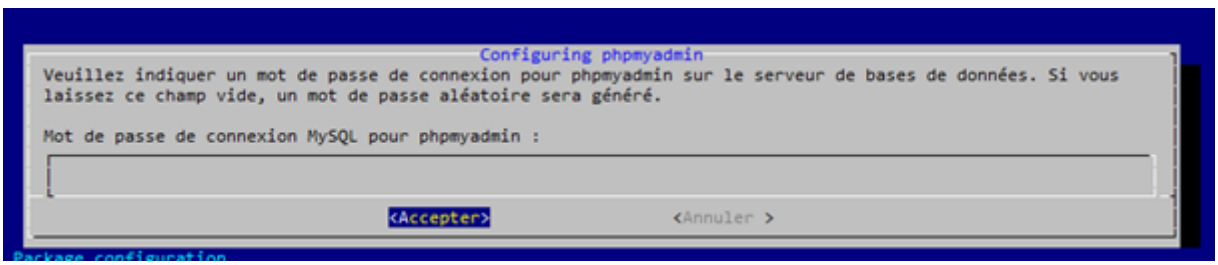
Nous installons ensuite PHPMyAdmin :

```
apt install phpmyadmin
```

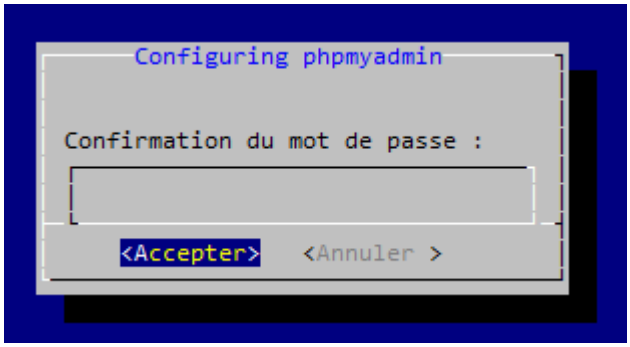
Nous suivons les indications en choisissant les mêmes options que sur les captures d'écrans suivantes. Nous commençons par choisir le mode de configuration de PHPMyAdmin par défaut.



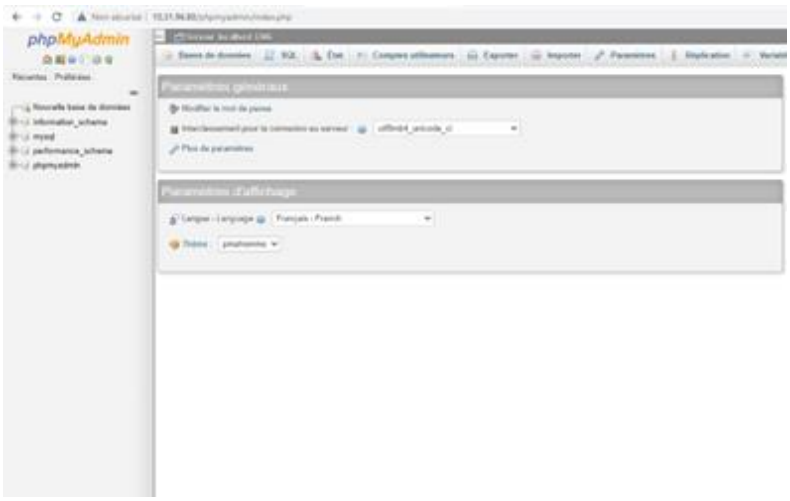
Nous laissons le mots de passe de connexion MySQL par défaut en laissant le champs vide.



Nous laissons également le champs de confirmation de mot de passe vide.



Nous nous connectons à PHPMyAdmin via l'interface web à l'adresse 10.31.96.80/phpmyadmin avec l'utilisateur dba créé, afin de tester les manipulations effectuées précédemment.



Enfin, nous entrons les informations de la base de données myCDS.db dans notre nouvelle base de données MariaDB et nous modifions le script bdd.php pour appeler cette nouvelle base de données :

```
< ?php
try {
        $servername = « mysql :host=localhost ;dbname=myCDS";
        $username = "dba";
        $password = "drowssap";
        $db = new PDO($servername, $username, $password);
        print ('<b>Connecté à la base</b><br><br>') ;
        print ('<b>Les artistes</b><br>') ;
        $sql = 'SELECT * from artist' ;
        $result = $db->query($sql) ;

        foreach ($result as $row) {
                print ($row['art_id'].'.\t'.$row['art_name'].'<br>')
        };

        print ('<br><b>Les albums</b><br>') ;
        $sql = 'SELECT * from cd' ;
        $result = $db->query($sql) ;

        foreach ($result as $row) {
                print ($row['art_id'].'\t'.$row['cd_title'].'.\t'.$row['cd_date'].'<br>');
        };

        $db = NULL;
}
catch(PDOException $e) {
        print ('Exception : '. $e->getMessage()):
}
?>
```

Finalisation

Par défaut, notre nouvelle base de données écoute sur localhost (l'adresse 127.0.0.1). Nous devons changer cette adresse par défaut pour pouvoir rendre notre base de données accessible à toutes les machines. Pour cela, nous devons modifier le fichier /etc/mysql/mariadb.conf.d/50-server.cnf, plus précisément la ligne bind = 127.0.0.1 que nous devons changer en bind = 0.0.0.0 :

```
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#

user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
basedir                 = /usr
datadir                 = /var/lib/mysql
tmpdir                  = /tmp
lc-messages-dir         = /usr/share/mysql
lc-messages             = en_US
skip-external-locking

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

Nous redémarrons le service de MariaDB pour que les changements soient effectifs :

```
systemctl restart mysql
```

Nous voyons maintenant à l'aide de la commande netstat -nat que mysql écoute l'adresse 0.0.0.0, c'est-à-dire toutes les machines.

```
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
```

Nous créons une nouvelle base de données sio pour répondre aux exigences du cahier des charges. Pour cela, nous nous connectons à mariadb à l'aide de la commande suivante et entrons ces commandes successivement :

```
mysql -u root -p

# Création de la base de données
create database sio ;

# Création de l'utilisateur sio
create user 'sio'@'localhost' identified by 'drowssap';

# Attribution des privileges (db.privileges to user@localhost)
grant all privileges on sio.* to 'sio'@'localhost' with grant option;

# Mise à jour des privilèges de la base de données
flush privileges ;
```