

Installer les serveurs web, php et mariadb sur un Linux

Testé avec :
RPI OS du 11janv.2021
Debian 10 et 11 (Bullseye)

Durée : 2 h

Pré-requis : Maîtriser les commandes de base sous Linux.

Installation d'Apache



Installer le paquet :

```
# apt update
# apt install apache2
```

S'assurer que le service est fonctionnel :

```
# service apache2 status    ou bien    # systemctl status apache2
```

S'assurer que le serveur écoute sur le port 80 :

```
$ ss -ltn
```

Tester l'accès au port 80 (en local ou depuis une machine distante) :

```
$ nc -z adresseIP 80; echo $? # doit renvoyer 0
```

Si problème, vérifiez que le pare-feu autorise bien le port 80 en TCP.

Si on dispose d'un navigateur web, tester l'accès local :

<http://127.0.0.1/>

Si on est en ligne de commande, utiliser la commande :

```
$ wget -O recu.html http://127.0.0.1
```

Puis, observer le contenu du fichier *recu.html*

Tests facultatifs

Donner les droits qui permettront de facilement modifier les fichiers hébergés :

```
# chown -R user:www-data /var/www/html/
# chmod -R 770 /var/www/html/
```

On remplace *user* par le nom d'utilisateur.

Vérifier que l'utilisateur concerné est bien membre du groupe *www-data* :

```
$ groups user
```

Si ce n'est pas le cas, ajouter l'utilisateur comme membre du groupe *www-data*, puis redémarrer :

```
# usermod -aG www-data user
```

```
# reboot
```

Pour que le service soit lancé à chaque démarrage de l'OS (c'est normalement déjà le cas par défaut) :

Si *InitV* utilisé :

```
# update-rc.d apache2 defaults
```

Si *SystemV* utilisé :

```
# systemctl enable apache2
```



Installation de php

Installer (il est en version 7.4 en 2022) :

```
# apt install php
```

C'est un méta paquet qui installe notamment :
libapache2-mod-php7.4 php7.4-fpm

Puis, relancer le service *apache2* :

```
# service apache2 restart    ou bien    # systemctl restart apache2
```

Remplacer le fichier *index.html* par un fichier nommé *index.php* (toujours dans */var/www/html/*).

Placer le contenu suivant dans le fichier *index.php* :

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

Dans le navigateur, recharger la page web. Les informations de php doivent apparaître.

Installation de MariaDB (le fork communautaire de MySQL)



Installer les paquets :

```
# apt install mariadb-server php-mysql
```

mariadb-server est le serveur du SGBD.
php-mysql permet de faire le lien entre php et mysql.

Relancer Apache et démarrer le serveur *mariadb* :

```
# service apache2 restart      ou bien      # systemctl restart apache2
# service mariadb start        ou bien      # systemctl start mariadb
```

Améliorer la sécurité en utilisant l'outil suivant :

```
# mariadb-secure-installation
```

```
Enter current password for root (enter for none):
Taper Entrée, car il n'y a pas encore de mot de passe root.
Switch to unix_socket authentication [Y/n] n
Non, car on veut un accès par mot de passe natif, pas par unix_socket.
Change the root password? [Y/n] Y
Oui, car ça permet de définir un mot de passe root.
$ mariadb -u root -p devient possible.
Remove anonymous users? [Y/n] Y
Oui, car on ne veut pas autoriser l'accès à n'importe qui.
Disallow root login remotely? [Y/n] Y
Oui, pour empêcher l'accès root distant.
Remove test database and access to it? [Y/n] n
Non, sauf si on veut supprimer la base test.
Reload privilege tables now? [Y/n] y
Oui, pour recharger la table des privilèges (prendre en compte les modifications).
```

Vérifier que *mariadb* fonctionne localement en se connectant au serveur :

```
# mariadb
```

On remarque que le prompt a changé : `MariaDB [(none)]>`

On peut afficher les comptes d'utilisateurs : `SELECT host,user,password FROM mysql.user ;`

Il doit y avoir un compte *root* qui permet un accès local seulement.

Faire `exit` pour sortir de *mariadb*.

Créer un nouveau compte utilisateur avec accès distant

Attention, ne pas faire d'erreur de saisie !

Dans la console MySQL (précédemment ouverte) :

On recrée un nouvel utilisateur 'bob' en personnalisant le mot de passe (ici 'pass') :

```
MariaDB [(none)]> CREATE USER 'bob'@'%' IDENTIFIED BY 'pass';
```

Remarque : On peut limiter à un accès local en remplaçant '%' par localhost

On donne à l'utilisateur tous les privilèges sur toutes les bases (*.*) :

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'bob'@'%' WITH GRANT OPTION;
```

On recharge la table des privilèges, avant de quitter :

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> quit;
```

Chaque commande
doit répondre par
Query OK

Tester l'accès local :

```
$ mariadb -u username -p
```

Plus besoin d'avoir les droits administrateur
pour lancer cette commande.

Autoriser l'accès distant au serveur *mariadb*

Modifier le fichier de configuration du serveur :

```
# nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Dans la section [mysqld], chercher la ligne *bind-address* et remplacer :

```
bind-address = 127.0.0.1 par bind-address = 0.0.0.0
```

Explications et autres méthodes ici :

<https://mariadb.com/kb/en/configuring-mariadb-for-remote-client-access/>

Redémarrer le service *mariadb* :

```
# service mariadb restart ou systemctl restart mariadb.service
```

Tester la connexion distante :

```
$ nc -z adresseIP 3306 ; echo $? # doit renvoyer 0
```

```
$ mariadb -h adresseIP -u username -p
```

Si problème, vérifiez que le pare-feu
autorise bien le port 3306 en TCP.

Choisir un outil graphique pour manipuler les bases de données :

- Phpmyadmin
- Adminer



Installer phpMyAdmin

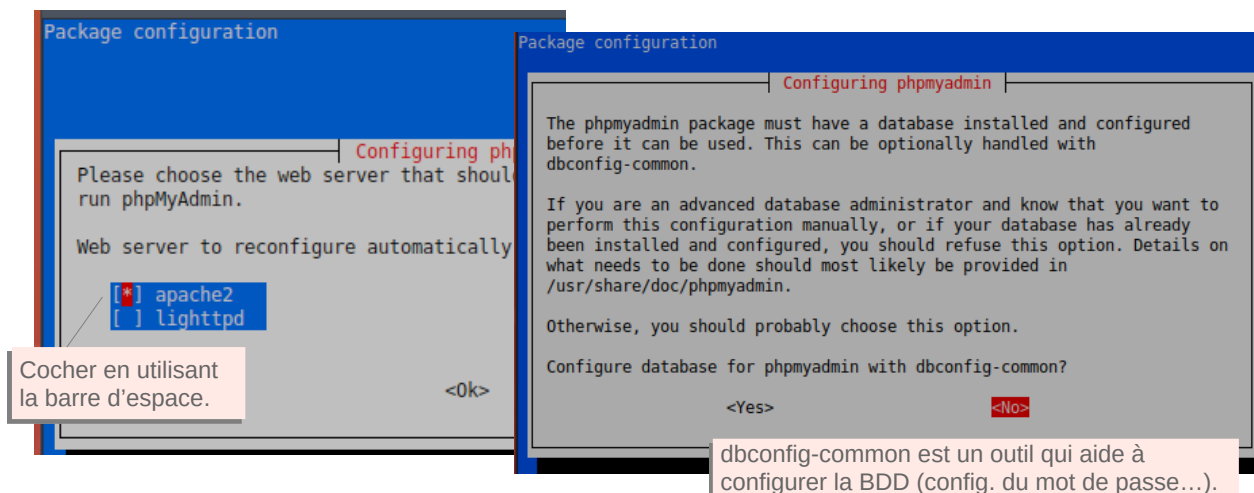
PHPMyAdmin est une application développée en php, et qui vise à fournir une interface simplifiée pour MySQL.

Installer : `# apt install phpmyadmin`

Dans certaines distri. Linux, le paquet ne sera disponible qu'en activant les backports dans `/etc/apt/sources.list.d/debian.list`

Attention, choisir le bon serveur http : sélectionner *Apache2* (mettre une * devant apache2).

À l'écran suivant, répondre « NON » à la question relative à la création d'une BDD phpmyadmin.



Puis, redémarrer le serveur web :

`# service apache2 restart`

ou bien

`# systemctl restart apache2`

Tester un accès local à l'adresse suivante : <http://127.0.0.1/phpmyadmin>

Tester un accès distant (depuis une autre machine).

Installer adminer



Installer le paquet :

`# apt install adminer`

Modifier la conf. d'apache :

`# a2enconf adminer.conf`

Redémarrer Apache :

`# service apache2 reload` ou bien `# systemctl restart apache2`

Tester un accès local à l'adresse suivante : <http://127.0.0.1/adminer>

Tester un accès distant (depuis une autre machine).