**TUTO installation de LAMP**

**CENTOS 7**

**Introduction**

LAMP est un acronyme :

* **L**inux : Le système d'exploitation
* **A**pache : Le serveur Web
* **M**ariaDB : Le serveur de Base de données (Fork de MySQL)
* **P**HP : Le langage de script

Avec LAMP, on peut donc mettre en place un serveur Web, hébergeant un site web dynamique écrit en PHP, tout en allant chercher des données dans une base MariaDB.  
  
Ce tutoriel est identique pour CentOS 7 et CentOS 8.  
Pour CentOS 8, vous pouvez remplacer les commandes **yum install** par **dnf install**.

**Installation des prérequis**

On va donc installer Apache (le paquet est **httpd**), MariaDB, PHP, et les modules PHP permettant la connexion à la base de données :

**Code BASH :**

**yum install** httpd php mariadb-server mariadb php-**{**gd,pdo,xml,mbstring,**zip**,mysqlnd,opcache,json**}**

L'extension php-mysqlnd installée grâce à la précédente commande permet à PHP de pouvoir communiquer avec les bases MySQL / MariaDB.  
Si vous ne voulez pas utiliser MySQL / MariaDB, il n'est pas nécessaire de l'installer. (Et vous pouvez ignorer la section MariaDB)

## Configuration du système

Si SELinux est actif, on active ce booléen permettant à **httpd**d'écrire dans le répertoires d'apache

setsebool -P httpd\_unified on

Et on autorise dans le pare-feu le protocole http (ici si la zone de l'interface est public) :

firewall-cmd --permanent --zone=public --add-service=http

firewall-cmd --reload

## Configuration des services

### Activation et démarrage des services

On active les services concernés au démarrage :  
  
Pour Apache :

systemctl **enable** --now httpd.service

Pour MariaDB :

systemctl **enable** --now mariadb.service

### Initialisation du moteur de base de données

On initialise le moteur de base de données :

mysql\_secure\_installation

On va pouvoir ainsi définir le mot de passe root, supprimer les comptes anonymes ...

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Set root password? [Y/n] Y

New password:

Re-enter new password:

Password updated successfully!

Reloading privilege tables..

... Success!

Remove anonymous users? [Y/n]

... Success!

Disallow root login remotely? [Y/n]

... Success!

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

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reload privilege tables now? [Y/n]

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB

installation should now be secure.

Thanks for using MariaDB!

**Si et seulement si**vous devez accéder à MariaDB depuis une autre machine; on autorise dans le pare-feu le protocole mysql (port 3306) (ici si la zone de l'interface est public) :

**Copier vers le presse-papierCode BASH :**

firewall-cmd --permanent --zone=public --add-service=mysql

firewall-cmd --reload

## Fichiers de configuration

### Apache2

#### Fichier de configuration httpd.conf

Le fichier de configuration principal d'Apache est **/etc/httpd/conf/httpd.conf**  
  
On peut sauvegarder le fichier original de configuration d'Apache avant toute modification :

**Copier vers le presse-papierCode BASH :**

**cp** **/**etc**/**httpd**/**conf**/**httpd.conf ~**/**httpd.conf.backup

Vous pouvez aussi inclure vos propres fichiers de configuration dans **/etc/httpd/conf.d/\*.conf**

### PHP

#### Fichier de configuration php.ini

Le fichier de configuration de PHP est **/etc/php.ini**  
  
Après chaque modification, il est nécessaire de recharger httpd :

**Copier vers le presse-papierCode BASH :**

systemctl restart httpd

Les lignes commentées sont précédées d'un point-virgule.

#### Quelques astuces php.ini

Si on développe, on peut améliorer la gestion des erreurs et des logs :

error\_reporting = E\_COMPILE\_ERROR**|**E\_RECOVERABLE\_ERROR**|**E\_ERROR**|**E\_CORE\_ERROR

error\_log = **/**var**/**log**/**php**/**error.log

max\_input\_time = 30

Et on créé le répertoire associé :

**mkdir** **/**var**/**log**/**php

**chown** apache **/**var**/**log**/**php

## Configuration pour un serveur web Sécurisé (SSL)

### Installation des prérequis pour SSL

Dans le cas d'un serveur web sécurisé, rajouter les paquets correspondant au module SSL :

**yum install** mod\_ssl openssl

On relance httpd :

systemctl restart httpd

### Configuration du pare-feu

On autorise dans le pare-feu le protocole https (ici si la zone de l'interface est public) :

firewall-cmd --permanent --zone=public --add-service=https

firewall-cmd --reload