**Atelier 2. Préparation de l’environnement de travail pour Symfony 5**

**Objectif :**

Préparer l’environnement de travail Linux (distribution Debian) pour développer des applications Web avec le framework Symfony. Pour cela, nous allons nous servir de deux machines virtuelles. Il faut installer au préalable le logiciel de virtualisation VirtualBox.

**Travail à faire :**

# Prérequis

Si le logiciel VirtualBox n’est pas installé sur votre machine physique, il faudra l’installer au préalable. Pour cela, télécharger ce logiciel sur le site : <https://www.virtualbox.org/>

Une fois le téléchargement terminé, lancer le fichier exécutable et suivre les étapes d’installation.

## Paquets à installer après démarrage de la machine :

Régulièrement, il faut penser à mettre à jour les paquets. Pour cela, vous allez ouvrir un terminal et lancer les commandes suivantes :

# apt update Mise à jour des paquets

**Remarque** : *Si après la mise à jour, le système propose de mettre à niveau le système d’exploitation, lancer la commande :* # apt upgrade

*Si des packages doivent être supprimés, lancer la commande :* # apt autoremove

On va installer quelques outils :

* vim (éditeur de texte) => la commande à exécuter est : # apt install vim
* net-tools => # apt install net-tools
* curl => # apt install curl

cURL est un utilitaire très important à connaître en tant que développeu.r.se. Il permet d’envoyer et de recevoir des données d’un site

<https://curl.se/docs/>

<https://geekflare.com/fr/curl-command-usage-with-example/>

## Installation de PHP

La version actuelle et stable de Symfony nécessite une version de PHP supérieure ou égale à PHP 8.1 (<https://www.php.net/supported-versions.php>).

En tant que root, lancer les commandes suivantes :

curl –sSL <https://packages.sury.org/php/README.txt> | bash -x

apt update

apt –y install php8.1

Installer les extensions suivantes de PHP 8.1

# apt install php8.1-mysql php8.1-gd php8.1-xml php8.1-mbstring php8.1-curl php8.1-bcmath

Installer le module Apache pour PHP :

# apt install libapache2-mod-php8.1

Installer PHP-FPM pour améliorer les performances :

# apt install php8.1-fpm

# a2enmod proxy\_fcgi setenvif

# a2enconf php8.1-fpm

# systemctl reload apache2

Un petit update suivi éventuellement d’un upgrade si cela est nécessaire :

# apt update

# apt upgrade

# php –v *pour vérifier l’installation de PHP8.1*

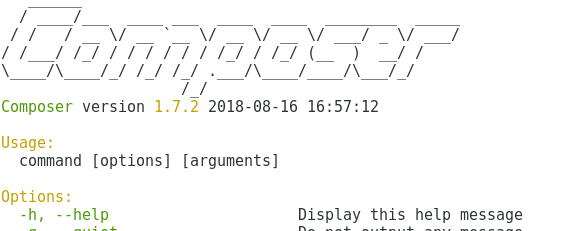
## Installation de composer

Composer est un gestionnaire de dépendances pour les projets PHP. Il permet aux développeurs de déclarer et d’installer les bibliothèques dont le projet a besoin et de les mettre à jour. Il permet aussi de garantir le même environnement en termes de dépendances et de versions quand on déploie le projet en production.

Aller sur le site de Composer (https://getcomposer.org) et dans l’onglet Download.

Exécuter les commandes en ligne sur votre terminal (Parapgraphe Command-line installation).

Pour vérifier l’installation de Composer, lancer la commande : $ composer

Si composer a été bien installé, vous aurez un message indiquant la version et l'aide sur Composer (voir extrait figure suivante).

# Installation du serveur Mariadb

Nous allons travailler avec le serveur de bases de données mariaDB. En effet, MySQL étant devenu la propriété d’Oracle, certains outils sont devenus payants et ne sont plus proposés dans la version gratuite (édition community). Consulter le site suivant qui présente un comparatif entre les deux moteurs de BDD : https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/mariadb-vs-mysql/

Pour répondre aux recommandations de la CNIL en termes de sécurisation des serveurs (voir annexe 1), nous allons installer le serveur mariaDB dans une autre machine Linux.

## Installer la machine Linux

Vous allez procéder de la même manière que l’étape 1.1 de l’étape précédente pour l’installation de la nouvelle machine. Elle doit avoir les caractéristiques suivantes :

* Nom de la VM : Serveur-Mariadb
* Capacité mémoire : 1Go
* Capacité du disque dur : 10 Go
* Nom de l’hôte : srvbdd
* Pas d’interface graphique
* Pas de serveur Web
* Pas de serveur ssh
* Installation des utilitaires usuels du système

## Installer mariaDB

Démarrer la machine après installation.

Se connecter en tant que root.

Mettre à jour le cache des paquets du système avec apt-update. Et si besoin, mettre à jour les paquets avec apt-upgrade.

Installer le paquet mariadb :

# apt install mariadb-server

Mettre à jour le cache des paquets :

#

Une fois l’installation terminée, vérifier à l’aide de la commande :

# systemctl status mariadb

## Sécuriser le serveur MySQL

Toujours dans l’optique de respecter les recommandations de la CNIL, nous allons sécuriser le serveur MariaDB en :

1. définissant le mot de passe du compte root du serveur de base de données,
2. supprimant le compte anonymous,
3. désactivant les accès distants pour root,
4. supprimant la base de données de test et les accès à cette base de données,
5. créant un nouvel administrateur de la base de données.

Pour cela, lancer le script :

root@srvbdd:~# mysql\_secure\_installation

Le script vous demande de définir le mot de passe du compte root. Saisir le mot de passe. Attention, le mot de passe ne s’affiche pas. Pour les questions suivantes :

Switch to unix\_socket authentication [Y/n] ? **n**

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

Continuer la configuration en respectant les points cités ci-dessus afin d’être conforme aux recommandations de la CNIL.

Remove anonymous users ? [Y/n] …..

Disallow root login remotely ? [Y/n] ….

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

Reload privilege tables now ? [Y/n] ….

Se connecter au serveur mariadb :

root@srvbdd:~# mysql

MariaDB [none]> use mysql => permet de sélectionner la BDD mysql

Lancer les requêtes SQL permettant de créer un utilisateur qui aura tous les privilèges sur toutes les bases de données, y compris le privilège de donner des permissions. Il ne peut se connecter qu’en local pour des raisons de sécurité :

MariaDB [mysql]> ……………………………………………………………………………………………………………………………..

MariaDB [mysql]> ……………………………………………………………………………………………………………………………..

Lancer la requête SQL permettant d’appliquer les nouvelles modifications :

MariaDB [mysql]> ……………………………………………………………………………………………………………………………..

MariaDB [mysql]> exit

Tester la création de votre nouveau compte :

Autoriser les connexions distantes au serveur de base de données. Pour cela, vous allez commenter la ligne bind-address du fichier : /etc/mysql/mariadb.conf.d/50-server.cnf.

## Installer le client mysql

Sur le serveur web (VM installée dans le TP de la séance 1), installer le client mysql avec la commande suivante :

root@srvbdd:~# apt install default-mysql-client

Tester les connexions distantes à votre serveur MariaDB avec les comptes root et le nouvel utilisateur précédemment créé :

root@srvweb:~$

root@srvweb:~$

**Remarque** : Avant de lancer mysql, vérifier que vos deux machines disposent d’adresses IP différentes et communiquent entre elles.

# Installation de l’IDE VS (Visual Studio Code)

Pour l’installation de Visual Studio Code, suivre le tutoriel à l’URL suivante : [How to Install Visual Studio Code on Debian 10 | Linuxize](https://linuxize.com/post/how-to-install-visual-studio-code-on-debian-10/#:~:text=The%20easiest%20and%20recommended%20way%20to%20install%20Visual,update%20sudo%20apt%20install%20gnupg2%20software-properties-common%20apt-transport-https%20curl)

### Annexe 1 – Extrait des recommandations de la CNIL pour sécuriser les serveurs

<https://www.cnil.fr/fr/securite-securiser-les-serveurs>

**Les précautions élémentaires**

1. Limiter l’accès aux outils et interfaces d’administration aux seules personnes habilitées.

* Utiliser des comptes de moindres privilèges pour les opérations courantes

1. Adopter une politique spécifique de mots de passe pour les administrateurs.
2. Installer les mises à jour critiques sans délai.
3. En matière d’administration de bases de données :

* Utiliser des comptes nominatifs pour l’accès aux bases de données et créer des comptes spécifiques à chaque application ;
* Mettre en œuvre des mesures contre les attaques par injection de code SQL, de scripts…

1. Effectuer des sauvegardes et les vérifier régulièrement.
2. Mettre en œuvre le protocole TLS (en remplacement de SSL) ou un protocole assurant le chiffrement et l’authentification, au minimum pour tout échange de données sur Internet.

**Ce qu’il ne faut pas faire**

1. Utiliser des services non sécurisés.
2. Utiliser pour d’autres fonctions les serveurs hébergeant les bases de données, notamment pour naviguer sur des sites web, accéder à la messagerie électronique, …
3. Placer les bases de données sur un serveur directement accessible depuis Internet.
4. Utiliser des comptes génériques.