

# TRAVAUX PRATIQUES

## SOUTIEN LINUX

Configuration d'un serveur



### OBJECTIFS

- Utiliser un gestionnaire de paquets
- Installer et configurer une pile Linux - Apache - MariaDB - PHP (LAMP Stack)
- Installer un pare-feu
- Configurer un nom de domaine
- Installer Wordpress

### CONSIGNES

Vous rédigerez un compte-rendu personnel reprenant chacune des commandes nécessaires à la réalisation de ce sujet.

Un compte-rendu numérique est recommandé afin de pouvoir facilement y copier vos commandes.

### CONTEXTE

De retour à ESICORP, vous trouvez une note de Max vous indiquant qu'il a dû remplacer votre N+1 à un comité de direction et qu'il faudra vous occuper de préparer un serveur Web à sa place.

L'instance du serveur est déjà prête avec son adresse IP, il ne vous reste plus qu'à tout configurer.

### CONNEXION A DISTANCE

- Connectez-vous en SSH à l'adresse IP qui vous a été communiquée.

Vos identifiants sont :

- Login : esicorp

- Mot de passe : esirem

Si vous vous rappelez le début du TP 1, vous devriez être tenté de faire quelque chose.

- Faites-le !

## PREPARATION

### Vérifications

- Exécutez la commande **cat /etc/os-release**
- Quelle distribution de Linux est installée sur l'ordinateur distant ?
- Quelle est la version de la distribution ?
- D'après le cours, quel est le gestionnaire de paquets par défaut pour cette distribution ?

### Mise à jour

Avant de procéder à l'installation, il est nécessaire de vérifier que le système est à jour.

- Tout d'abord, mettez à jour les liens du gestionnaire de paquet avec la commande **apt update**

Un problème ? Ah oui, **apt update** est une commande d'administration que seul le super utilisateur (« root ») peut utiliser. L'utilisateur « esicorp » fait partie des « sudoers », les utilisateurs pouvant utiliser la commande **sudo**. Lorsque **sudo** précède une commande, cette dernière est exécutée par l'utilisateur « root » avec tous les privilèges qui lui sont associés.

- Réessayez de mettre à jour les liens du gestionnaire de paquets avec la commande **sudo apt update**

#### Note

L'utilisation de la commande **sudo** ne sera pas indiquée dans la suite du sujet. A vous de juger quand elle est nécessaire et quand vous pouvez vous en passer, mais **ne l'utilisez pas systématiquement**

- Consultez la page **man** de la commande **apt**
- Quelle différence y a-t-il entre **apt upgrade** et **apt full-upgrade** ?
- Dans le cas présent, quel est le plus approprié ?
- Mettez à jour le système avec la commande de votre choix. Mais, choisissez judicieusement...

#### Note

L'option **-y** permet de valider automatiquement les demandes de confirmation au cours de la mise à jour. Vous lisez ces lignes trop tard ? Dommage...

## APACHE

Apache est un serveur Web, c'est-à-dire, de manière très simpliste, un logiciel qui attend qu'un navigateur lui demande un page Web et si cette dernière existe, la lui transmet.

### Installation

Pour installer un paquet, vous allez utiliser la commande **apt install nom\_du\_paquet**

- Quel est le paquet à installer pour le serveur Apache ?
- Installez le serveur Apache

### Vérification

Vous allez commencer par regarder ce que vous dit le système par rapport au fonctionnement d'Apache

- Exécutez la commande **systemctl status apache2**

Si tout est bon vous devriez voir « active (running) » en vert.

A présent, faisons un autre test.

- Ouvrez un navigateur web sur votre ordinateur physique
- Dans la barre d'adresse du navigateur, saisissez « http://adresse\_ip » (bien sûr, vous remplacerez « adresse\_ip » par l'adresse IP de votre ordinateur distant)

Alors, bien ou bien ? Par contre, elle vient d'où cette page web ?

- Allez dans le dossier « /var/www/html »
- Listez le contenu du dossier.

Vous devriez trouver un fichier « index.html ». Il s'agit de la page Web par défaut chargée lorsqu'un navigateur se connecte à votre serveur Web. Apache la génère automatiquement lors de son installation.

- Supprimez le fichier index.html (commande **rm**)
- Créez un nouveau fichier avec la commande **nano index.html**

Nano est un éditeur de texte rudimentaire vous permettant d'éditer le contenu d'un fichier texte. N'essayez pas de vous déplacer à la souris, cela ne marchera pas. Seules les flèches directionnelles vous permettront de vous déplacer.

- Saisissez « Ma page à moi ! »
- Enregistrez et quittez l'éditeur Nano en appuyant sur les touches « Ctrl+x »
- Actualisez la page de votre navigateur. Wow !

## PHP

PHP est un langage de programmation initialement conçu pour réaliser des modèles de pages web dynamiques. Nous verrons un peu plus loin ce que cela signifie. Aujourd'hui PHP est au cœur de Framework tels que Symfony ou Laravel.

### INSTALLATION

Vous allez installer un ensemble de paquets contenant PHP ainsi que des modules apportant des fonctionnalités supplémentaires.

- Installez les paquets suivants :
  - php (interpréteur PHP)
  - php-pdo (interface de connexion aux bases de données)
  - php-mysql (driver de connexion aux base de données MySQL)
  - php-zip (gestion des archives ZIP)
  - php-gd (traitement d'images)
  - php-mbstring (gestion de l'UTF-8 et autres chaines de caractères multi-octets)

#### Note

Vous pouvez installer plusieurs paquets en une seule ligne de commande, en séparant le nom des paquets à installer par un espace : **sudo apt install paquet1 paquet2 paquet3 ...**

### Vérification

La commande **php -v** affiche la version de PHP installée si celui est bien présent sur l'ordinateur.

- Exécutez la commande si dessus.

C'est tout bon ? Pour vous en assurer pleinement, vous allez créer un fichier PHP dans le dossier du site par défaut d'Apache.

- Allez dans le dossier « /var/www/html »
- Créez un fichier « test.php » à l'aide de la commande **nano test.php**
- Copiez/collez le code suivant dans l'éditeur :

```
<?php
echo "Sur le serveur, il est ".date("H:i:s");
```

- Dans la barre d'adresse de votre navigateur, saisissez [http://adresse\\_ip/test.php](http://adresse_ip/test.php)
- Vous devriez voir l'heure du serveur apparaître. Actualisez la page, l'heure change. C'est ça, une page web dynamique. Une page dont le contenu change sans que l'on ait à éditer le code HTML de cette dernière.

## MARIADB

MariaDB est un système de gestion de base de données (SGBD). Si vous ne savez pas encore ce qu'est une base de données, dites-vous que c'est un programme chargé de stocker et d'organiser de l'information dans des fichiers. On verra cela plus en détail au second semestre. MariaDB est un fork de MySQL.

### Installation

- Installez le paquet « mariadb-server »

### Configuration

MariaDB et MySQL fournissent un outil de configuration permettant d'effectuer le paramétrage de base du SGBD.

- Exécutez la commande **mysql\_secure\_installation**
- Saisissez les informations suivantes
  - « Enter current password for root » : saisissez le mot de passe de votre choix pour l'administrateur de vos bases de données.

#### Important

Conservez précieusement ce mot de passe. Sans quoi, vous ne pourrez pas terminer ce TP.

#### Bonne pratique

Afin d'augmenter la sécurité de votre serveur, il est impératif de choisir des mots de passe forts ET différents pour chacun des services que vous configurez (base de données, ftp, ssh, ...). De cette façon, si un compte est compromis, les autres services restent protégés.

#### Astuce

Pour gérer facilement une multitude de mots de passe complexe, utilisez un coffre-fort à mots de passe comme KeePass. Choisissez des coffres-forts qui s'installe localement et évitez les coffres-forts en ligne comme NordPass, ils finissent tous par se faire hacker. Faites régulièrement une sauvegarde de votre coffre-fort car, si vous perdez les mots de passe de vos serveurs, vous perdez aussi toutes les données qui s'y trouvent. Vous et vos clients n'allez pas apprécier ce moment 😊

- « Switch to unix\_socket authentication », appuyez sur « y »

- « Change the root password », appuyez sur « n » (vous l'avez déjà défini à la première étape)
- « Remove anonymous users », appuyez sur « y »
- « Disallow root login remotely », appuyez sur « y » (à votre avis, pourquoi ?)
- « Remove test database and access ot it », appuyez sur « y »
- « Reload privilege table now », appuyez sur « y »

Voilà qui est fait !

### Vérification

- Vérifiez que le service « mariadb » fonctionne correctement avec la commande `systemctl` vue précédemment.

## PARE-FEU

L'installation d'un pare-feu permet de limiter l'accès aux ports de l'ordinateur tant en entrée qu'en sortie.

- Installez le paquet « ufw » (pour Uncomplicated FireWall)
- Exécutez la commande `sudo ufw allow "WWW Full"` pour ouvrir les ports liés aux protocoles Web
- Exécutez la commande `sudo ufw allow "SSH"` pour ouvrir les ports nécessaires à la connexion à distance avec votre ordinateur.

### **DANGER**

Si vous oubliez d'indiquer au Pare-feu que les ports SSH doivent être ouverts, vous perdrez définitivement l'accès à votre ordinateur distant.

- Exécutez la commande `sudo ufw enable` pour activer le pare-feu
- Exécutez la commande `sudo ufw status` pour vérifier l'état du pare-feu et les ports ouverts.

## CONFIGURATION

### Heure du serveur

Vous n'y avez peut-être pas prêté attention tout à l'heure, mais lorsque vous avez affiché la page « test.php » dans votre navigateur, l'heure indiquée par le serveur avait deux heures de retard.

- Retournez sur votre navigateur et affichez de nouveau la page test.php pour confirmer le problème
- Puis exécutez la commande `date` pour afficher l'heure du serveur

Par défaut, les serveurs sont configurés sur l'heure UTC. Or le fuseau horaire de la France est UTC+2.

- Exécutez la commande **timedatectl**

Vous devriez voir que la « Time zone » est configurée sur « Etc/UTC (UTC, +0000) ». Vous allez modifier cette configuration mais pour cela vous allez avoir besoin de connaître le nom de la time zone à utiliser.

- Exécutez la commande **timedatectl list-timezones** et trouvez le nom de la time zone de Paris
- Exécutez la commande **timedatectl set-timezone nom\_de\_la\_timezone**
- Exécutez de nouveau la commande **date**
- Actualisez la page de votre navigateur. Pardon ?!

PHP possède sa propre gestion des timezones.

- Allez dans le dossier « /etc/php/8.2/apache2 »
- Editez le fichier « php.ini » avec Nano
- Recherchez le mot clé « timezone » en appuyant sur « Ctrl+w »
- Modifiez la ligne suivante (attention au point-virgule) :

```
;date.timezone =
```

devient

```
date.timezone = Europe/Paris
```

Il faut à présent redémarrer le serveur Apache pour que les nouveaux paramètres soient pris en compte

- Exécutez la commande **sudo service apache2 restart**
- Actualisez la page de votre navigateur. Yes !

## Nom de domaine

Il est temps d'associer votre site Internet à un nom de domaine.

- Commencez par aller voir votre N+1 (M. Meunier) qui s'occupe de la réservation des noms de domaines. N'oubliez pas de prendre l'adresse IP de votre serveur avec vous.

A présent, il va falloir configurer Apache pour que ce dernier sache quoi faire lorsqu'il reçoit une requête adressée à votre nom de domaine.

- Allez dans le dossier « /etc/apache2/sites-available »

- Avec Nano, créez un fichier « nom\_de\_domaine.conf » où vous remplacerez « nom\_de\_domaine » par... votre nom de domaine.
- Copiez/collez le texte suivant dans l'éditeur :

```
<VirtualHost *:80>
    ServerName nom_de_domaine
    DocumentRoot /var/www/html/
    <Directory /var/www/html/ >
        Options -Indexes +FollowSymLinks +MultiViews
        AllowOverride All
    </Directory>
    ErrorLog /var/log/apache2/error.log
    LogLevel warn
    CustomLog /var/log/access.log combined
    ServerSignature Off
</VirtualHost>
```

Cela indiquera à Apache2 d'aller chercher les éléments relatifs à votre nom de domaine dans le dossier « /var/www/html ».

Le dossier « sites-available » contient l'ensemble des sites web hébergés par le serveur. Un serveur peut en effet héberger plusieurs sites. Il faut à présent copier le fichier de configuration précédent, dans le dossier « sites-enabled » pour indiquer que le site est actif.

Le problème d'une copie, c'est que si on modifie la configuration dans le dossier « sites-available » il faudra penser à copier de nouveau le fichier dans le dossier « sites-enabled ». Ce qui est pénible et sour d'erreur si on oublie.

Nous allons donc créer un lien symbolique, c'est à dire une sorte de raccourci, vers le fichier de configuration se trouvant dans le dossier sites-available. Le lien symbolique sera vu comme un fichier à part, mais en réalité, il fera référence au fichier de configuration initial. Ainsi, toutes modifications apportées au fichier dans sites-available est répercutée dans sites-enabled.

- Exécutez la commande suivante en prenant soin de remplacer votre nom de domaine :

```
ln -s /etc/apache2/sites-available/nom_de_domaine.conf /etc/apache2/sites-enabled/nom_de_domaine.conf
```

- Listez le contenu (`ls -al`) du dossier « /etc/apache2/sites-enabled » et observez comment apparait le lien symbolique
- Redémarrez le serveur Apache comme vu précédemment.
- Ouvrez un navigateur et saisissez l'adresse [http://nom\\_de\\_domaine](http://nom_de_domaine)

## Certificat SSL

Les certificats de sécurité permettent de chiffrer la communication entre le navigateur et le serveur. Ils permettent également de vérifier que le serveur est bien celui que l'on s'attend à trouver derrière un nom de domaine et qu'il n'y a pas eu de détournement de l'information.



Let's Encrypt fournit gratuitement des certificats de sécurité pour une durée renouvelable de 3 mois.

Suivez les instructions de <https://certbot.eff.org/> pour mettre en place un certificat SSL et pouvoir utiliser le protocole HTTPS.

## BONUS : WORDPRESS

Wordpress est un CMS (Content Management System). Il permet de créer facilement un site Internet et propose une interface pour la publication de contenus.

### Base de données

Wordpress a besoin d'une base de données pour fonctionner.

- Exécutez la commande **sudo mariadb** pour entrer dans l'invite de commande de MariaDB
- Exécutez la requête SQL **CREATE DATABASE wordpress ;**

Nous allons également avoir besoin d'un utilisateur MariaDB dédié à Wordpress

- Exécutez la requête SQL **CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'mot\_de\_passe' ;** (à vous de définir le mot de passe et ne l'oubliez pas !)

Il faut à présent donner le droit d'utiliser la base de données « wordpress » à l'utilisateur « wordpress ».

- Exécutez la requête SQL **GRANT ALL PRIVILEGES ON wordpress.\* TO 'wordpress'@'localhost' ;**
- Saisissez ensuite **exit** pour quitter l'invite de commandes de MariaDB.

### Téléchargement de Wordpress

- Placez vous dans le dossier **/var/www/html**
- Supprimez tous les fichiers présents dans ce dossier

La commande **wget** permet de télécharger un contenu depuis Internet.

- Exécutez la commande **wget https://fr.wordpress.org/latest-fr\_FR.zip**

Vous venez de télécharger une archive ZIP. Pour la décompresser vous allez avoir besoin d'installer la commande **unzip**

- Installez la commande **unzip**
- Décompressez l'archive
- Listez le contenu du dossier « **/var/www/html** » avec l'option **-al**
- Qui est le propriétaire du dossier « **wordpress** » ?

Pour que Wordpress puisse s'installer, il faut qu'il puisse écrire dans le dossier où il se trouve. Pour cela, nous allons changer le propriétaire du dossier par « www-data ». C'est un utilisateur caché créé et utilisé par Apache pour accéder au système de fichiers.

- Exécutez la commande **sudo chown -R www-data wordpress**
- Listez de nouveau le contenu du dossier pour vérifier que le propriétaire a bien changé.

Vous devriez être prêt pour la dernière étape.

## Installation

- Dans votre navigateur, saisissez l'adresse [https://nom\\_de\\_domaine/wordpress](https://nom_de_domaine/wordpress)
- Suivez les instructions.