

Présentation du compte rendu :

Ce compte rendu va porter sur le travail pratique que j'ai eu à faire en utilisant un logiciel de virtualisation intitulé VMWARE WORKSTATION PRO dans lequel j'ai installé une machine dotée d'une distribution Linux que l'on appelle Debian.

Tout ce qui va être expliqué seront les commandes que j'ai eu à utiliser dans le terminal de la machine virtuelle.

Pour résumer, toutes les manœuvres consistaient à mettre en place des installations de logiciel via le terminal notamment Apache2, MariaDB et également le langage PHP et ce sera expliqué en 3 étapes dans l'ordre.

1) APACHE2 :

Installation de Apache

Dans un premier temps avant de commencer le TP il fallait agir en tant que super utilisateur ou plutôt en tant que "root" en tapant la commande " su " qui nous demandait le mot de passe pour nous laisser entrer en mode super utilisateur.

Une fois fait, il est possible de commencer à installer Apache2, comme on peut le voir sur le screen juste en dessous c'est l'utilisateur "root" qui agit en insérant la commande : sudo apt-get install -y apache2 .

```
root@debian:/home/sofiane# sudo apt-get install -y apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.56-1~deb11u2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 5 non mis à jour.
```

Accès au serveur Apache

Après l'avoir installé nous devons vérifier s'il est possible d'aller dans le serveur Apache et pour ce faire voici le screen ci-dessous qui nous montre comment faire.

Entrer la commande : `ip address` . Elle permet rejoindre le site du serveur apache lorsque l'on entre l'adresse Ip : 192.168.50.133 dans la barre de recherche d'Internet. (Voir résultat juste en dessous)

```
sofiane@debian: ~  
sofiane@debian:~$ ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 00:0c:29:b8:99:45 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.50.133/24 brd 192.168.50.255 scope global dynamic noprefixroute ens33  
        valid_lft 1767sec preferred_lft 1767sec  
    inet6 fe80::20c:29ff:feb8:9945/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
sofiane@debian:~$ sudo apache2ctl -v  
  
Nous espérons que vous avez reçu de votre administrateur système local les consignes traditionnelles. Généralement, elles se concentrent sur ces trois éléments :  
  
#1) Respectez la vie privée des autres.
```

Voici le résultat :

Page serveur Apache2 :

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Activer le démarrage d'Apache2 en meme temps que Debian :

Cette commande que l'on peut voir " `sudo systemctl enable apache2` " rend possible le démarrage simultané de Debian et Apache2 ce qui permet de gagner du temps si l'on revient sur la machine Apache2 sera prêt à l'usage.

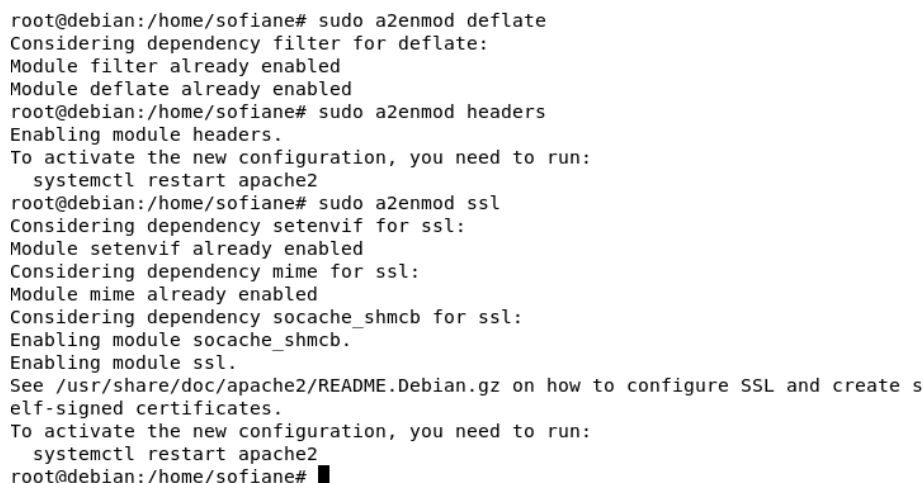
A terminal window titled 'sofiane@debian: ~' with search, menu, and close buttons. The user 'sofiane@debian' runs 'su' to become root. The root prompt shows the user is at '/home/sofiane'. The command 'sudo systemctl enable apache2' is entered. The system output shows it synchronizes the state of 'apache2.service' with the SysV script and executes '/lib/systemd/systemd-sysv-install enable apache2'. The prompt returns to root at '/home/sofiane#'.

```
sofiane@debian:~$ su
Mot de passe :
root@debian:/home/sofiane# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@debian:/home/sofiane#
```

Activation des modules, deflate,headers,ssl :

Il était nécessaire de disposer des modules deflate,headers,ssl alors j'ai tapé la commande suivante :

`sudo a2enmod [nom du module]`. En voici le résultat.

A terminal window showing the execution of 'a2enmod' commands. The user is root at '/home/sofiane'. The first command 'sudo a2enmod deflate' shows 'deflate' is already enabled. The second command 'sudo a2enmod headers' shows 'headers' is being enabled and provides instructions to restart 'apache2'. The third command 'sudo a2enmod ssl' shows 'ssl' is being enabled along with its dependencies 'setenvif', 'mime', and 'socache_shmcb'. It also provides instructions to restart 'apache2' and mentions a README file for SSL configuration.

```
root@debian:/home/sofiane# sudo a2enmod deflate
Considering dependency filter for deflate:
Module filter already enabled
Module deflate already enabled
root@debian:/home/sofiane# sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:/home/sofiane# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:/home/sofiane#
```

A la fin de la mise en place des modules il a été obligatoire de redémarrer apache2 voilà donc le screen ci-dessous:

Rédemarrage de Apache2 :

```
root@debian:/home/sofiane# systemctl restart apache2
root@debian:/home/sofiane# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@debian:/home/sofiane#
```

2) PHP:

Installation de PHP

Il est possible de voir la commande “ sudo apt-get install -y php “ pour avoir php sur la machine virtuelle

on y voit ci dessous le résultat de cette commande.

```
root@debian:/home/sofiane# sudo apt-get install -y php
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common php7.4-json
  php7.4-opcache php7.4-readline
Paquets suggérés :
  php-pear
Les NOUVEAUX paquets suivants seront installés :
  libapache2-mod-php7.4 php php-common php7.4 php7.4-cli php7.4-common
  php7.4-json php7.4-opcache php7.4-readline
0 mis à jour, 9 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 4 126 ko dans les archives.
Après cette opération, 18,0 Mo d'espace disque supplémentaires seront utilisés.
```

Installation des paquets pour les interactions entre PHP et MariaDB :

Successivement à l'installation de php comme MariaDB allait nous être utile par la suite il a fallu intégrer en l'occurrence les paquets destinés à faire communiquer php à MariaDB voici la commande de manière non exhaustive ainsi que le résultat obtenu sur le screen :

“ sudo apt-get install -y [liste de tout les paquets] “.

```

root@debian:/home/sofiane# sudo apt-get install -y php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Note : sélection de « php7.4-common » au lieu de « php-pdo »
php7.4-common est déjà la version la plus récente (7.4.33-1+deb11u4).
php7.4-common passé en « installé manuellement ».
Les paquets supplémentaires suivants seront installés :
  libonig5 libzip4 php7.4-bcmath php7.4-curl php7.4-gd php7.4-mbstring
  php7.4-mysql php7.4-xml php7.4-zip
Les NOUVEAUX paquets suivants seront installés :
  libonig5 libzip4 php-bcmath php-curl php-gd php-mbstring php-mysql php-pear
  php-xml php-zip php7.4-bcmath php7.4-curl php7.4-gd php7.4-mbstring
  php7.4-mysql php7.4-xml php7.4-zip
0 mis à jour, 17 nouvellement installés, 0 à enlever et 0 non mis à jour.

```

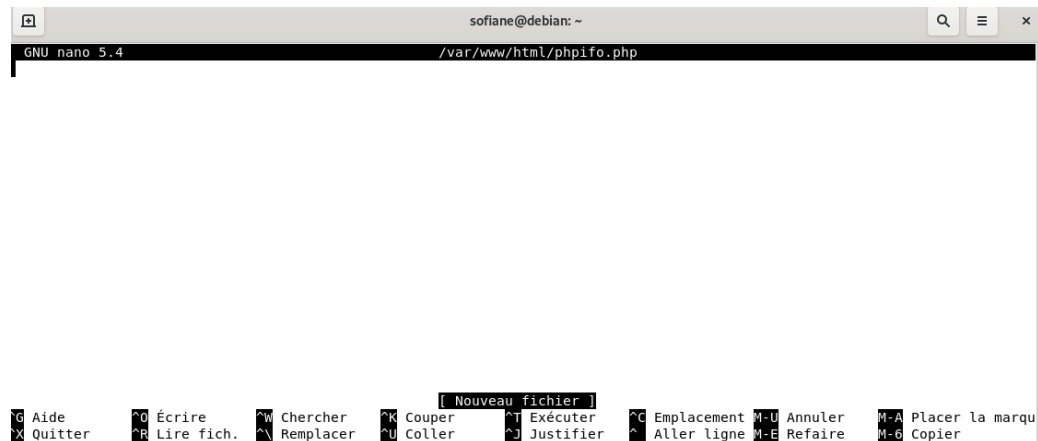
Création fichier phpinfo :

Par la suite la création du fichier phpinfo était également l'une des consigne du TP voici le tableau de commande lors de la création de celui-ci on ne le voit pas sur le screen mais il fallait y inscrire ceci :

```
<?
```

```
php.info();
```

```
?>
```




Fichier phpinfo créé :

```
|root@debian:/home/sofiane# sudo nano /var/www/html/phpinfo.php
```

Tout cela nous a permis d'accéder à la page serveur php via l'adresse php qui nous était fourni ainsi que l'adresse ip obtenue au début du TP (192.168.50.133).
<http://192.168.50.133/phpinfo.php>



PHP Version 7.4.33	
	
System	Linux debian 5.10.0-27-amd64 #1 SMP Debian 5.10.205-2 (2023-12-31) x86_64
Build Date	Jun 9 2023 16:51:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

3) MARIA DB :

Installation MariaDB :

Nous voilà à la dernière étape du TP, MariaDB.

Pour ce faire il faut d'abord taper : `sudo apt-get install -y mariadb-server` .

Cela mettra MariaDB dans la machine virtuelle.

```
root@debian:/home/sofiane# sudo apt-get install -y mariadb-server
_ecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
_ecture des informations d'état... Fait
_es paquets supplémentaires suivants seront installés :
 galera-4 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdbd-mariadb-perl libdbi-perl
 libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-template-perl libmariadb3 libsigsegv2 libterm-readkey-perl
 mariadb-client-10.5 mariadb-client-core-10.5 mariadb-common mariadb-server-10.5 mariadb-server-core-10.5 mysql-common
 rsync socat
```

Affichage de la version de MariaDB et Affichage de la connexion à la Base de Donnée:

Une fois l'étape précédent faite, on vérifie si MariaDB est bel et bien installé avec la bonne version avec la commande : `mariadb -V` .

Après la vérification, on se connecte à la base de donnée mis à notre disposition via MariaDB grâce à la commande : `sudo mariadb -u root -p`.

Et enfin la démonstration de la commande servant à quitter l'espace de la base de donnée avec : `exit`

```

root@debian:/home/sofiane# mariadb -V
mariadb Ver 15.1 Distrib 10.5.21-MariaDB, for debian-linux-gnu (x86_64) using EditLine wrapper
root@debian:/home/sofiane# sudo mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.21-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 |
| mysql |
| performance_schema |
+-----+
3 rows in set (0,001 sec)

MariaDB [(none)]> exit
Bve

```

Redémarrage de MariaDB :

Voici la dernière étape, tout simplement redémarrer mariadb.
 Commande : `systemctl restart mariadb` .

```

root@debian:/home/sofiane# systemctl restart mariadb
root@debian:/home/sofiane# █

```