Mission 3

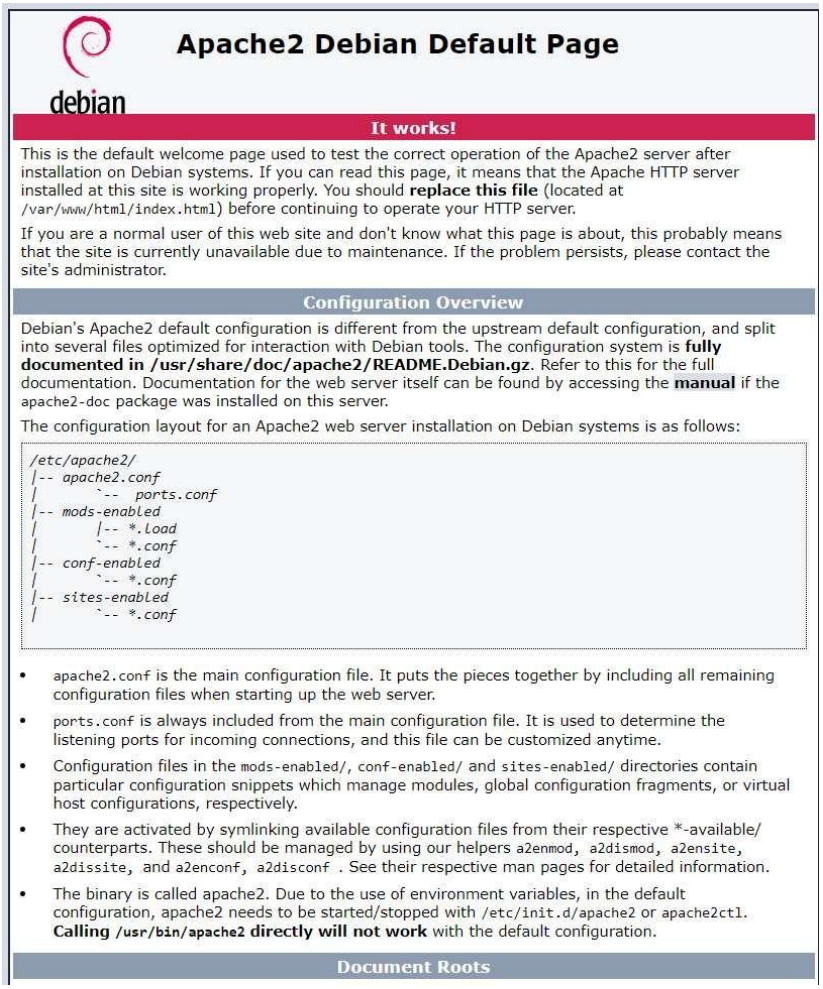
SOMMAIRE

Installation d'Apache2

Installation d'Apache2 :

```
# apt install apache2 -y
```

Test d'accès : ouvrir l'URL http://adresse_du_serveur pour vérifier que la page Apache s'affiche correctement.



Apache2 Debian Default Page

debian

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.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

Installation de PHP

Installation des modules nécessaires pour MariaDB :

```
# apt install libapache2-mod-php php-mysql -y
```

Installation des prérequis PHP :

```
# apt install apt-transport-https lsb-release ca-certificates curl -y
```

Importation de la clé de signature Sury :(mainteneur de la dernière version de PHP/ Debian)

```
# wget -O /etc/apt/trusted.gpg.d/php.gpg
https://packages.sury.org/php/apt.gpg
```

Installation de PHP :

```
# apt install php -y
```

Test d'accès : ouvrir http://adresse_du_serveur/test.php.

PHP Version 8.2.7

System	Linux debian12 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-12-30) x86_64
Build Date	Jun 9 2023 19:37:27
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/15-xml.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-curl.ini, /etc/php/8.2/apache2/conf.d/20-dom.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-mysqli.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-simplexml.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini, /etc/php/8.2/apache2/conf.d/20-xmlreader.ini, /etc/php/8.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.2/apache2/conf.d/20-xsl.ini, /etc/php/8.2/apache2/conf.d/20-zip.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829

Installation de MariaDB

Installation :

```
# apt install mariadb-server -y
```

Configuration de MariaDB :

Modification de la configuration réseau :

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

Changer bind-address de 127.0.0.1 à 0.0.0.0.

```
bind-address            = 0.0.0.0
```

Commandes de gestion du service MariaDB :

```
systemctl restart mariadb     # Redémarrer MariaDB

systemctl stop mariadb        # Arrêter MariaDB

systemctl start mariadb       # Démarrer MariaDB

ss -nlt                       # Vérifier l'écoute du port TCP
```

```
root@llteg:/etc/mysql/mariadb.conf.d# ss -nlt
State       Recv-Q      Send-Q           Local Address:Port           Peer Address:Port      Process
LISTEN      0           80                   0.0.0.0:3306                0.0.0.0:*
LISTEN      0           100                127.0.0.1:25                  0.0.0.0:*
LISTEN      0           4096                     *:22                        *:*
LISTEN      0           511                      *:80                        *:*
LISTEN      0           100                    [::1]:25                    [::]:*
```

Adresse en 0.0.0.0 car la capture d'écran a été effectué après le changement d'adresse 127.0.0.1 -> 0.0.0.0.

Sécurisation de MariaDB :

Définir un mot de passe pour l'utilisateur root :

```
SET PASSWORD FOR 'root'@'localhost' =
PASSWORD('P@$$word1');
```

Importation et gestion de la base de données GestStages

Importation du fichier SQL

```
scp geststages.sql admin@10.187.35.137:/home/admin
```

Création de la base de données :

```
mariadb -u admin -p < creationdbb.sql
```

Vérification de la présence des tables dans bdd_geststages.

```
Database changed
MariaDB [bdd_geststages]> show tables
    -> ;
+--------------------------+
| Tables_in_bdd_geststages |
+--------------------------+
| classe                   |
| entreprise               |
| etudiant                 |
| mission                  |
| prof_classe              |
| professeur               |
| spec_entreprise          |
| specialite               |
| stage                    |
+--------------------------+
9 rows in set (0.000 sec)

MariaDB [bdd geststages]>
```

Sauvegarde de la base de données

```
mysqldump -u root -p bdd_geststages > sauvegardebdd.sql
```

-u : identifiant utilisé

-p : demande de saisie du mot de passe

-bdd_geststages > sauvegardebdd.sql : export dans un fichier

Déploiement de l'application Web GestStages

On importe le fichier creationbdd depuis un pc tier au conteneur.

```
# scp geststage.zip admin10.187.35.140:/home/admin
```

Installer “unzip” pour pouvoir le dé-zipper.

```
# apt install unzip -y

# unzip geststages.zip
```

Ensuite on va le déplacer dans le dossier html.

```
# mv geststages /var/www/
```


```
html/
```

Gestion des utilisateurs Linux

Création de l'utilisateur chef :

```
# adduser chef
```

Ajout au groupe dev :

```
# usermod -aG dev chef
```

Définir les droits d'accès :

```
# chown -R :dev /var/www/html/

# chmod -R 775 /var/www/html/
```

Gestion des utilisateurs MariaDB

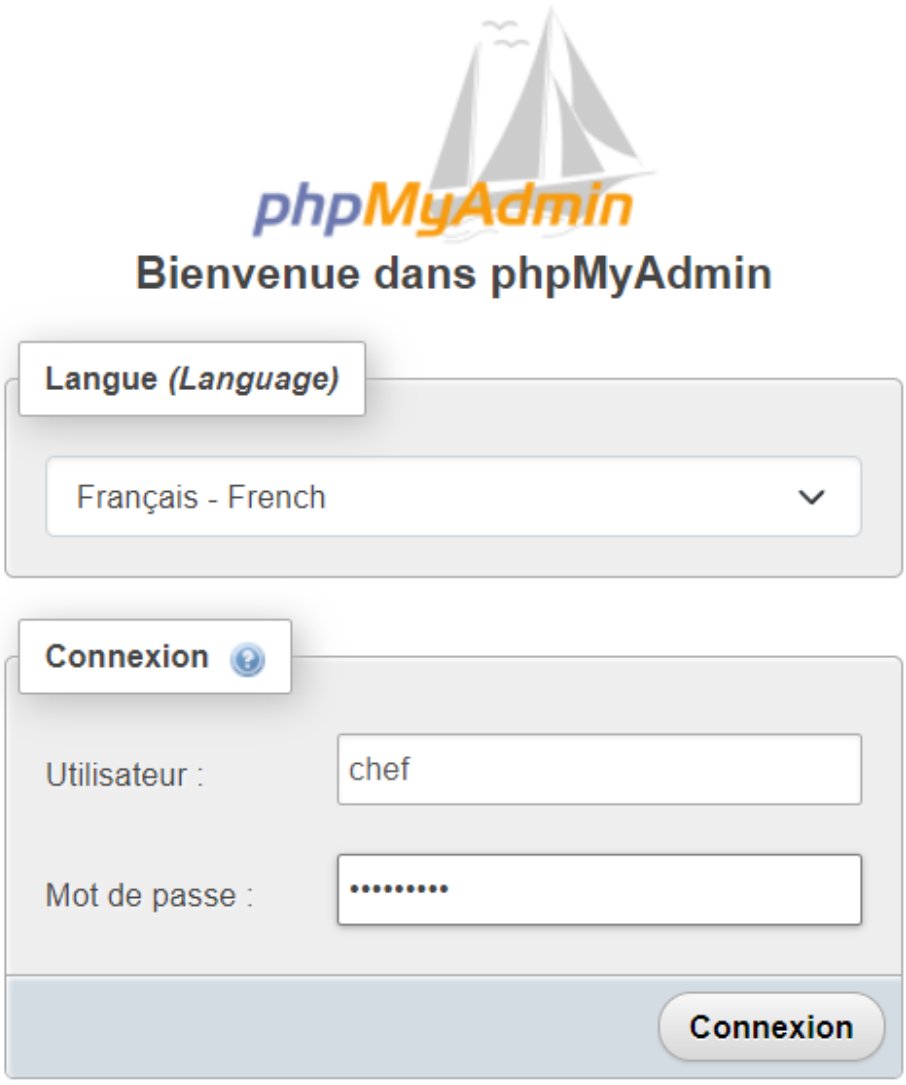
Création de l'utilisateur chef avec tous les privilèges sur la base :

```
CREATE USER 'chef'@'%' IDENTIFIED BY 'P@$$word1';

GRANT ALL PRIVILEGES ON bdd_geststages.* TO 'chef'@'%';

FLUSH PRIVILEGES;
```

On peut ensuite accéder à la base de données avec nos identifiants :



Installation et configuration de WordPress

Téléchargement:

```
# wget https://wordpress.org/latest.zip
```

Création de la base de données :

```
# CREATE DATABASE wp202110_itconnect;
```

Vérification de la présence des tables :

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| bdd_geststages     |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
| wp202110_itconnect |
+--------------------+
```

Attribution des privilèges à l'utilisateur 'chef' :

```
# GRANT ALL PRIVILEGES ON wp202110_itconnect.* TO
'chef'@'%';
```

On extrait le fichier latest.zip dans le répertoire /var/www/html :

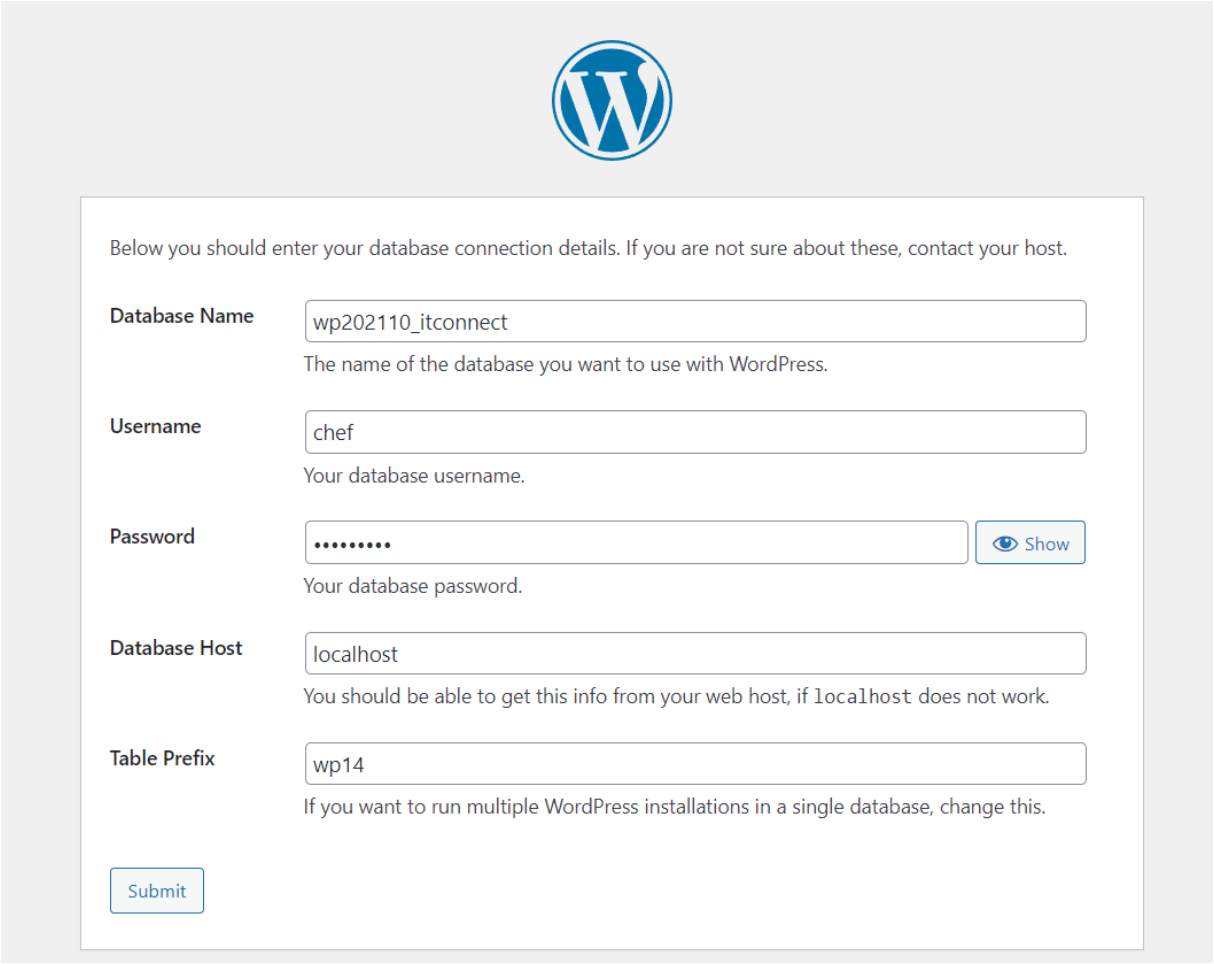
```
# unzip latest.zip -d /var/www/html
```

On donne les droits d'utilisateur avec les commandes :

```
# chown -R chef:chef /var/www/html/

# chmod -R 775 /var/www/html/
```

On accède à l'interface WordPress avec l'ip 10.187.35.140/wp-admin :



On a accès au site local :