CESI
ÉCOLE D'INGÉNIEURS

INSTALLATION DE LAMP SOUS WSL

Développement Web – CPI A2 Sciences du Numérique 25-26

V1.0 – 06/10/2025 – J. GALLET

Préambule

Lors du projet *Conception et Programmation Objet* de ce début d'année, vous avez mis en place un environnement Ubuntu sur votre machine en utilisant WSL.

Nous allons dans cette séquence repartir de cet environnement et y ajouter l'installation d'un environnement Web.

Avant de procéder à l'installation de l'environnement Web, vous devez vous assurer que :

1. Votre machine physique contient assez d'espace disque et de mémoire
2. Votre environnement WSL avec Ubuntu fonctionne correctement
3. Votre environnement est connecté à Internet (faire un ping vers les serveurs de Google par exemple)

Si votre environnement n'est pas fonctionnel, veuillez reprendre la séquence 0 du bloc POO avant de poursuivre.

SOMMAIRE

1. MISE EN PLACE DE L'ENVIRONNEMENT

La mise en place d'un serveur Web est caractérisée par l'installation de différents modules ayant chacun un rôle précis. Nous allons voir comment installer chaque élément dans un environnement Linux.

La pile logicielle LAMP, correspondant aux acronyme Linux Apache MySQL PHP, désigne un environnement complet regroupant un système d'exploitation, un serveur HTTP, un système de gestion de base de données et un langage de programmation interprété permettant de mettre en place un serveur Web.

Cet environnement peut être déployé sur n'importe quelle version Linux. Il existe également des variantes pour d'autres systèmes d'exploitation :

- WAMP – Windows, Apache, MySQL, PHP
- MAMP – Mac, Apache, MySQL, PHP
- XAMPP – Multiplateforme (X), Apache, MariaDB, PHP, Perl

Il y a plusieurs façons de déployer la pile LAMP, nous allons utiliser ici un terminal et procéder à toute l'installation et la configuration en lignes de commandes.

En effet, beaucoup de serveurs Web Linux sont dépourvus d'interface graphique (serveur headless) et sont administrés à distance en ligne de commande via un protocole sécurisé (SSH).

LANCEMENT

A partir du terminal Windows, lancez votre sous-système Linux Ubuntu :

```
wsl -d Ubuntu
```

MISE A JOUR DES PAQUETS

Afin de bénéficier des dernières versions des paquets identifiés sur les dépôts configurés, mettez à jour le cache du gestionnaire de paquets :

```
$ sudo apt update
```

Cette commande n'est qu'une mise à jour de la liste, elle n'installe ni ne met à jour aucun paquet. Pour cela il faut utiliser par la suite la commande suivante, qui met à jour tous les paquets installés vers leurs versions les plus récentes disponibles dans les dépôts et qui installe les nouvelles versions des paquets tout en conservant les configurations existantes :

```
$ sudo apt upgrade
```

INSTALLATION ET CONFIGURATION DES MODULES

Notre environnement Web sera constitué de 3 modules :

- **Apache** pour le serveur Web
- **PHP** pour le langage de programmation
- **MySQL** pour la base de données

Nous allons dans ce workshop nous focaliser sur le serveur Web Apache. Bien que le langage de programmation PHP et la base de données MySQL soient installés pendant cette séquence, vous verrez leur utilisation plus en détails dans les prochaines séquences.

Les modules peuvent être installés séparément. Dans notre cas, pour faciliter l'installation des paquets Apache2, MySQL et PHP, ainsi que certains modules, nous allons directement utiliser le meta package "lamp-server".

Saisissiez la ligne de commande suivante dans votre terminal (le caractère "^" à la fin est important) :

```
$ sudo apt upda sudo apt-get install lamp-server^te
```

Serveur Web (Apache)

Démarrage du serveur Web

Une fois que l'installation des paquets est terminée, vous pouvez démarrer le serveur Web Apache2 :

```
$ sudo service apache2 start
```

Page par défaut

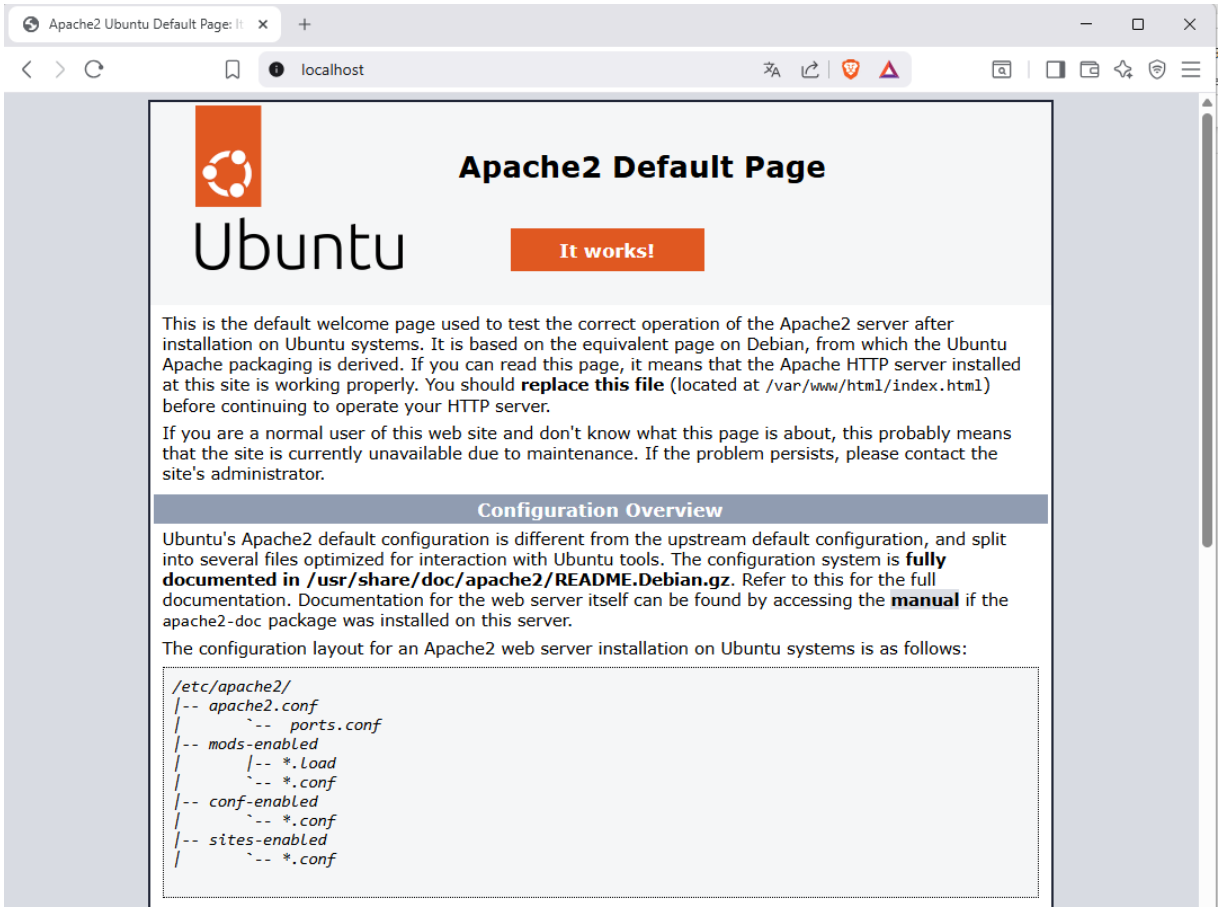
Le package est livré avec une page `index.html` très basique, mais qui va permettre de s'assurer que le serveur Web est joignable. Ce fichier est disponible dans le répertoire "`/var/www/html`", qui correspond au répertoire du site par défaut d'Apache2.

Vous pouvez vérifier le contenu du répertoire avec la commande ci-dessous :

```
$ ls /var/www/html
```

Visualisation de la page

Depuis votre machine Windows, vous devriez pouvoir accéder à la page d'accueil de votre site en ouvrant un navigateur web et en appelant la page http://localhost :



Langage interprété (PHP)

Contrôle de la version PHP

Dans le terminal, tapez la commande suivante pour valider l'installation de PHP et contrôler la version installée :

```
$ php -v


PHP 8.3.6 (cli) (built: Jul 14 2025 18:30:55) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
```

Vérification du fonctionnement de l'environnement Apache/PHP

Nous allons à présent vérifier que PHP fonctionne bien avec le serveur Apache.

Le dossier public par défaut d'Apache où sont situées les pages Web se trouvent dans le répertoire `/var/www/html` (cette information est également visible dans la page par défaut d'Apache que vous avez visualisé tout à l'heure).

Nous allons créer un fichier `info.php` qui va appeler une fonction PHP permettant d'afficher des informations détaillées sur la configuration PHP. Pour cela nous pouvons utiliser l'éditeur de texte en ligne de commande `nano` :

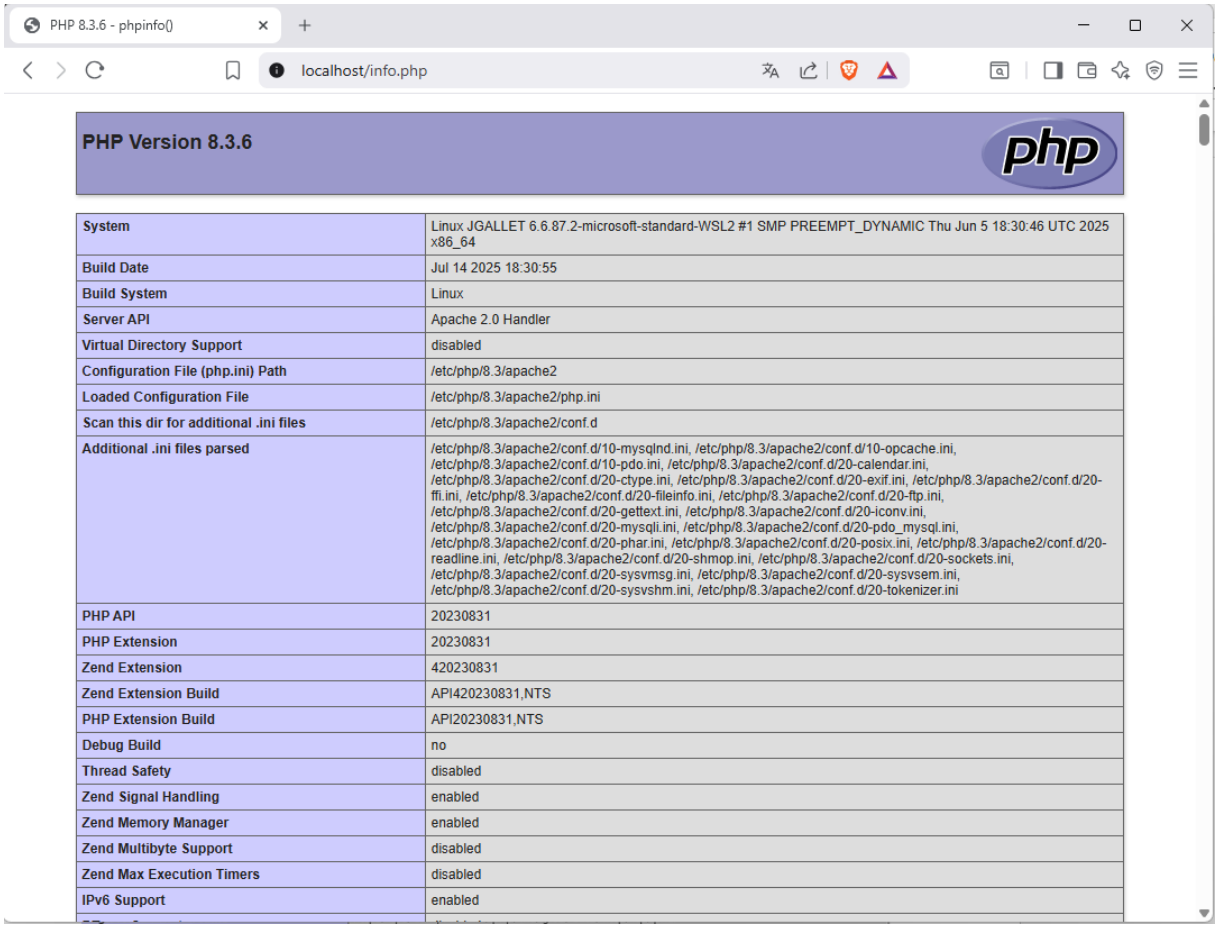
```
$ sudo nano /var/www/html/info.php
```

Dans l'éditeur qui s'ouvre, ajoutez la ligne suivante : `<?php phpinfo(); ?>`

Enregistrez et fermez l'éditeur en utilisant la combinaison de touches « `CTRL+X` » puis « `Y+entrée` » pour valider les changements.

Ouvrez le navigateur Web et tapez dans l'url http://localhost/info.php

Vous devriez avoir les informations de PHP qui s'affichent :



System	Linux JGALLET 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun 5 18:30:46 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-mysqli.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled

Cette page est destinée à tester le bon fonctionnement de PHP et de fournir des informations sur son environnement, il ne faudra pas la mettre à disposition du public dans un environnement de production.

Base de données (MySQL)

Démarrage du service MySQL

```
$ sudo service mysql start
```

Si vous souhaitez que MySQL se lance automatiquement au démarrage, tapez la commande suivante :

```
$ sudo systemctl enable mysql
```

Contrôle de la version

La commande ci-dessous va permettre d'obtenir la version actuelle de MySQL que l'on a déployé sur notre distribution Ubuntu :

```
$ sudo mysql -V
```

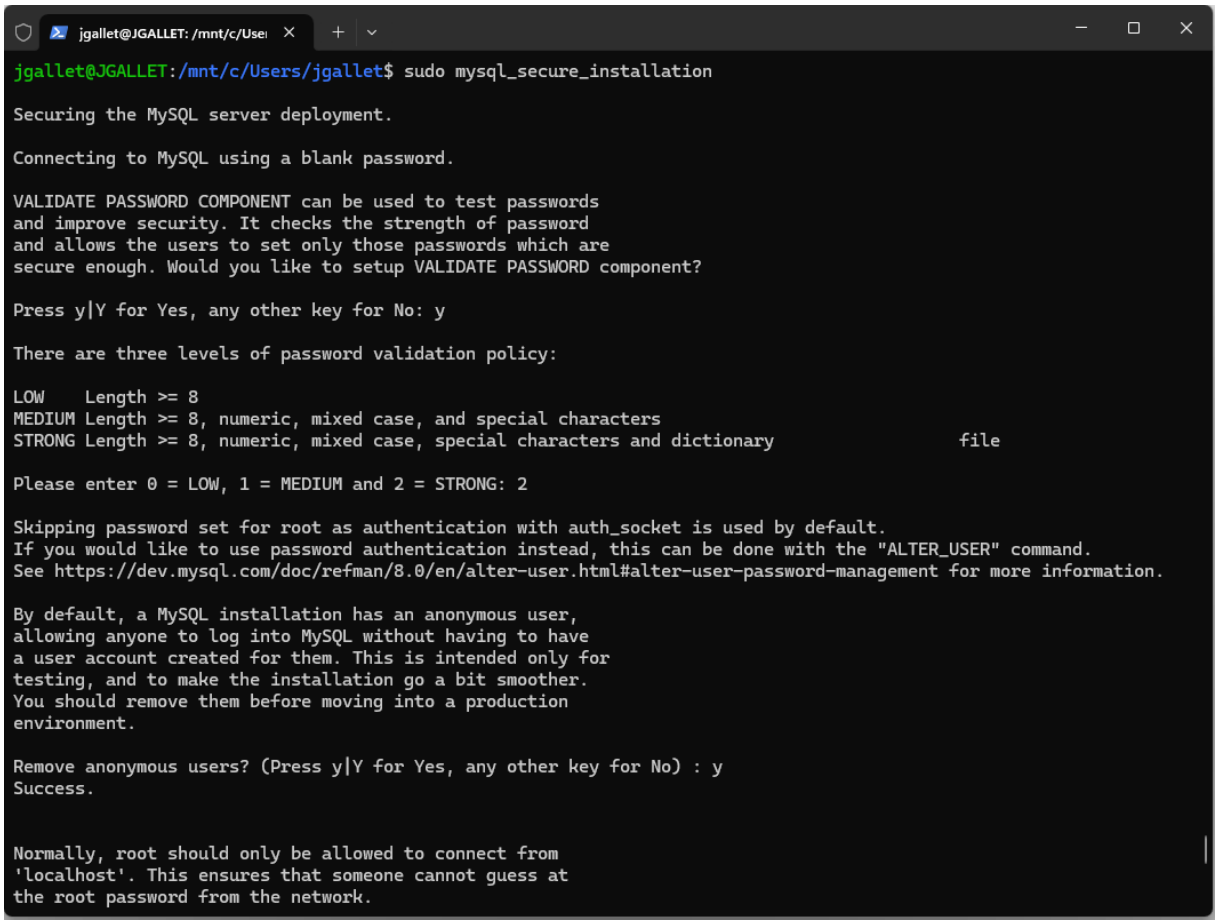
Sécurisation des accès

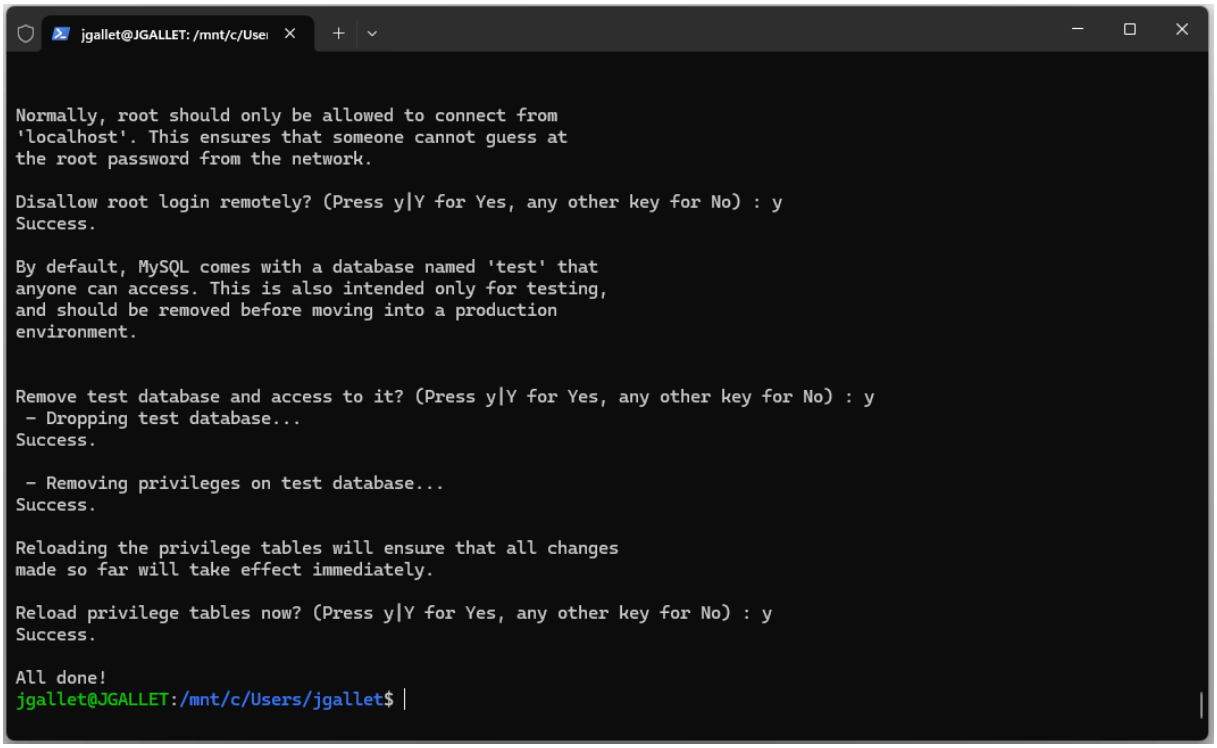
Les étapes suivantes vont permettre de configurer et de sécuriser l'accès à MySQL. Pour cela un script tout fait existe, qu'il suffit de lancer via la commande suivante :

```
$ sudo mysql_secure_installation
```

Suivez les instructions pour définir la robustesse du mot de passe, supprimer les utilisateurs anonymes, désactiver la connexion root à distance, supprimer la base de données de test et recharger les tables de privilèges :

- `VALIDATE PASSWORD PLUGIN : yes`
- `Validation policy : 2 (strong)` → Cette configuration signifie que le mot de l'utilisateur devra contenir au moins 8 caractères avec des majuscules, minuscules, chiffres et caractères spéciaux et ne pas être présent dans le dictionnaire des mots de passe connus. Dans le cas d'un serveur de test en développement vous pouvez toutefois choisir une authentification plus faible.
- `Remove anonymous users : yes` Désactivation des utilisateurs anonymes
- `Disable root login remotely : yes` Désactivation du login root à distance (seulement en local)
- `Remove test database : yes`
- `Remove privilege tables now? : yes` Rechargement des paramètres configurés





Création d'un utilisateur

Par défaut, l'utilisateur `root` est configuré pour s'authentifier via le plugin `auth_socket` de MySQL. Ainsi, MySQL ne demande pas de mot de passe mais vérifie que l'utilisateur système qui lance le service est bien `root`. Nous devons modifier ce comportement car nous souhaitons pouvoir accéder à la base de données via un client MySQL.

Le principe consiste à mettre à jour la méthode d'authentification et de définir un mot de passe pour l'utilisateur root (veuillez à définir un mot de passe fort, dans cet exemple « `A2#DevWeb!` ») :

1. Lancez MySQL en tant qu'utilisateur root :

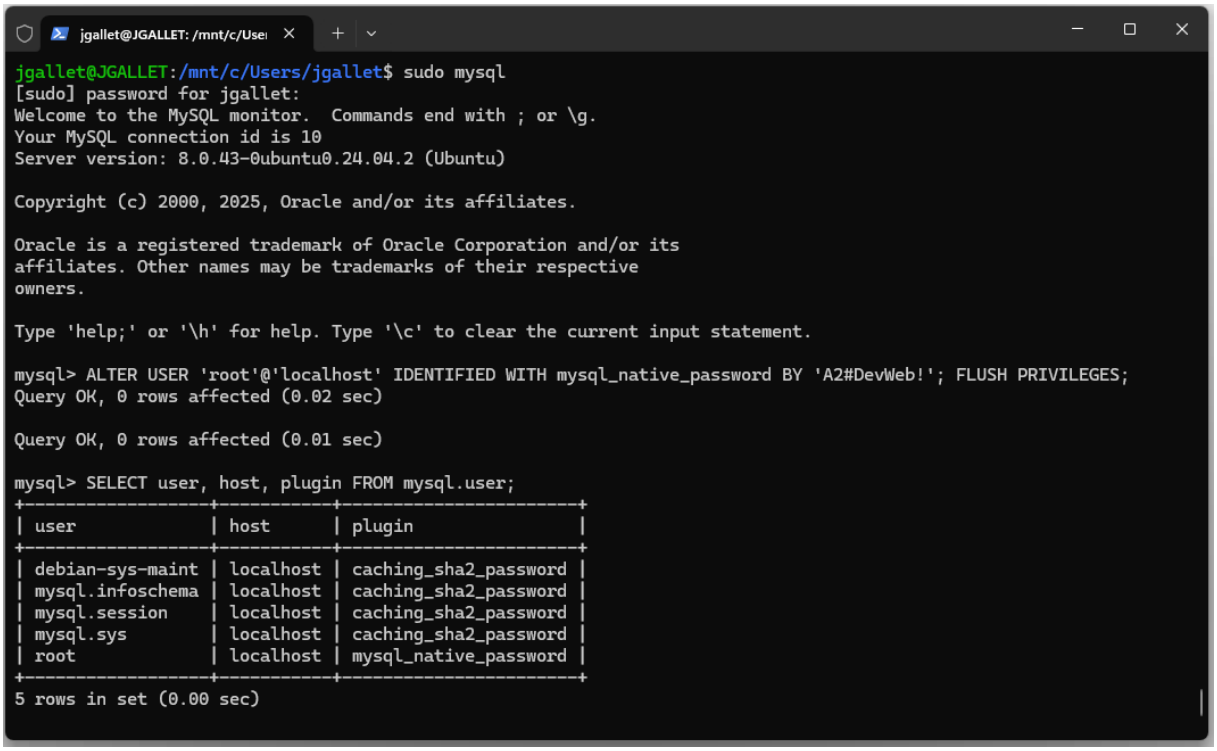
```
$ sudo mysql
```

2. Changez le mode d'authentification de root :

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
'A2#DevWeb!'; FLUSH PRIVILEGES;
```

3. Vérifiez que les changements ont bien été pris en compte (vous devriez avoir `mysql_native_password` pour l'utilisateur `root`) :

```
SELECT user, host, plugin FROM mysql.user;
```



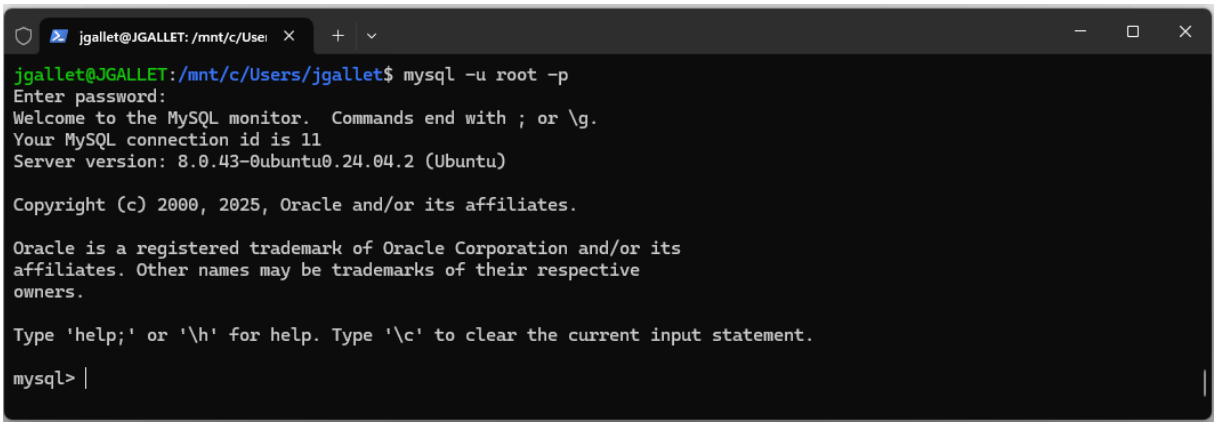
Quittez l'interface de MySQL en tapant '`exit`'. Vous reviendrez au prompt du terminal.

Test de connexion

Pour vérifier que le couple utilisateur / mot de passe est correct, lancez la commande suivante après avoir quitté l'environnement MySQL (commande « exit ») :

```
$ mysql -u root -p
```

En saisissant le mot de passe définit dans la requête SQL, vous devriez pouvoir accéder à l'interface :



Accès à la base de données via PHPMyAdmin

Afin d'accéder à une interface graphique permettant de gérer la base de données MySQL, nous allons utiliser un outil nommé PHPMyAdmin.

PHPMyAdmin nécessitant une base de données, nous allons nous appuyer sur MySQL que nous avons précédemment installé.

Création de la base de données et de l'utilisateur

Avant de procéder à l'installation de PHPMyAdmin, nous allons créer la base de données *phpmyadmin* et un utilisateur dédié à celle-ci. Pour l'utilisateur `phpmyadmin` vous devez définir un mot de passe qui respecte les critères de sécurité définis dans l'étape précédente (dans cet exemple il s'agit de « `PhpMy@dm1n` ».

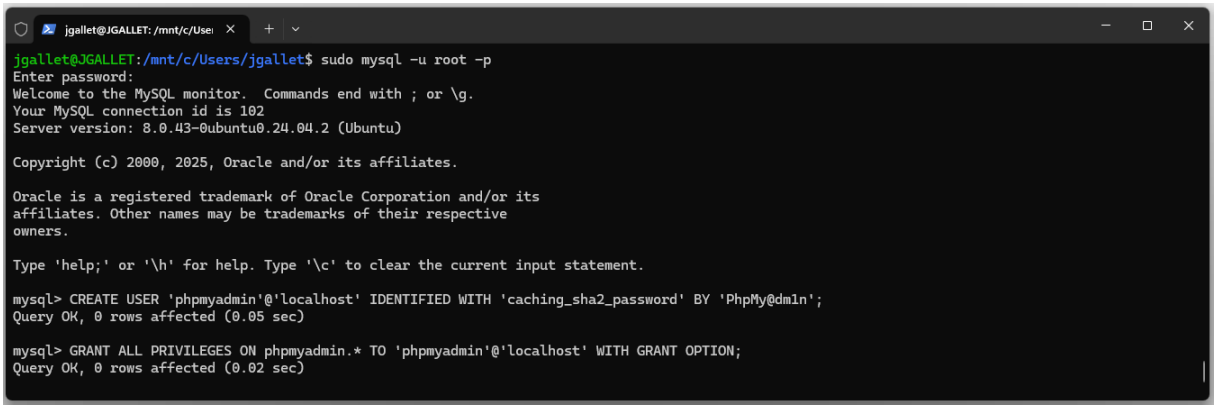
Dans l'interprétateur de commande MySQL, saisissez les requêtes suivantes :

```
CREATE DATABASE phpmyadmin;

CREATE USER 'phpmyadmin'@'localhost' IDENTIFIED WITH 'caching_sha2_password' BY 'PhpMy@dm1n';

GRANT ALL PRIVILEGES ON phpmyadmin.* TO 'phpmyadmin'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;
```



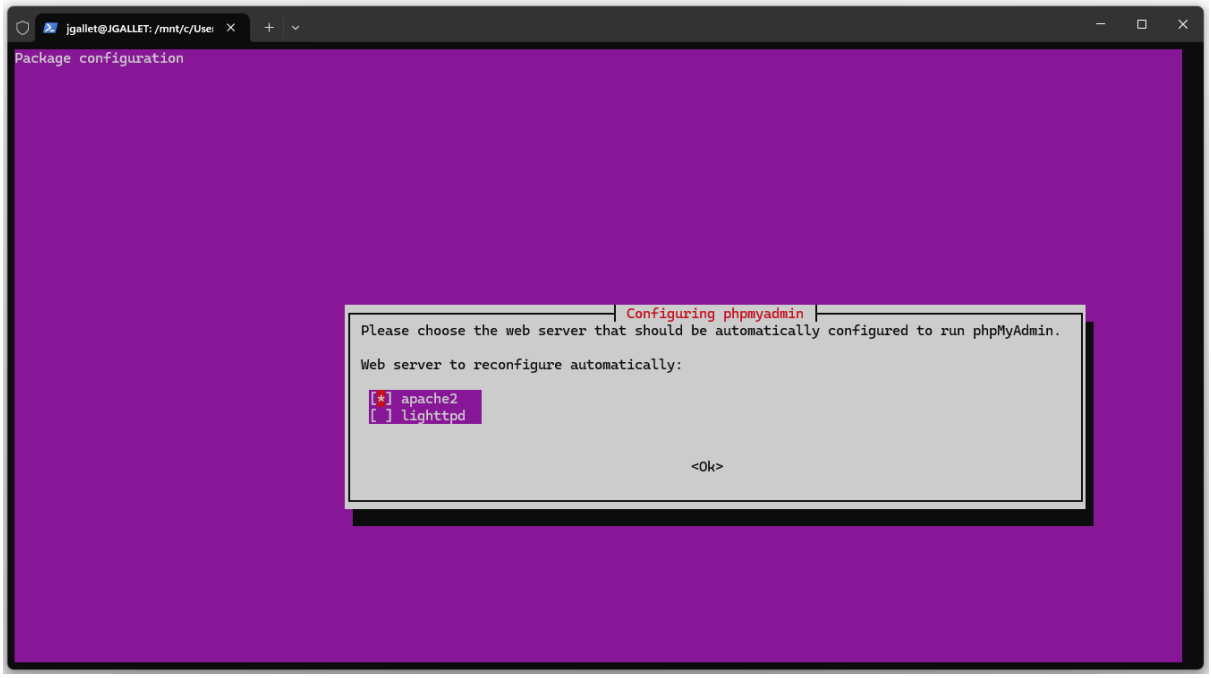
Installation

Quittez l'interprétateur de commandes MySQl et lancez l'installation de PHPMyAdmin :

```
$ sudo apt install phpmyadmin
```

Sélection du serveur Web

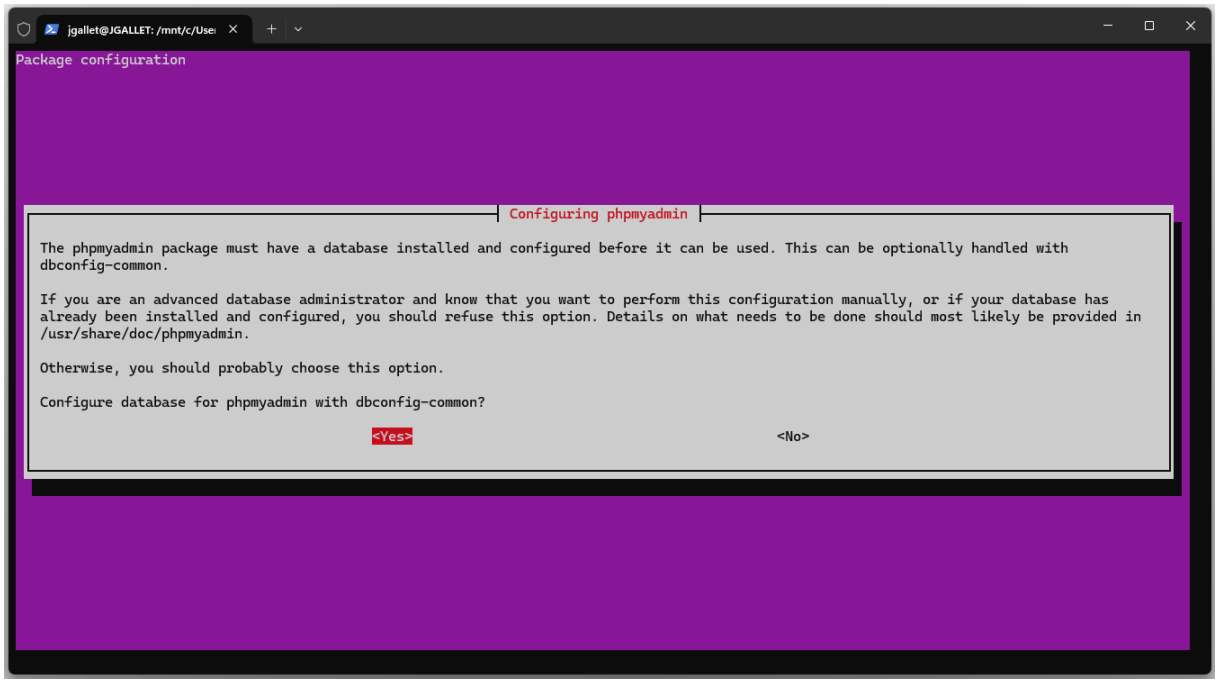
La première étape consiste à sélectionner le serveur Web apache2 en appuyant sur la barre d'espace puis en validant par OK (une étoile devrait apparaître sur la sélection) :



L'installation débute.

Configuration de la base de données

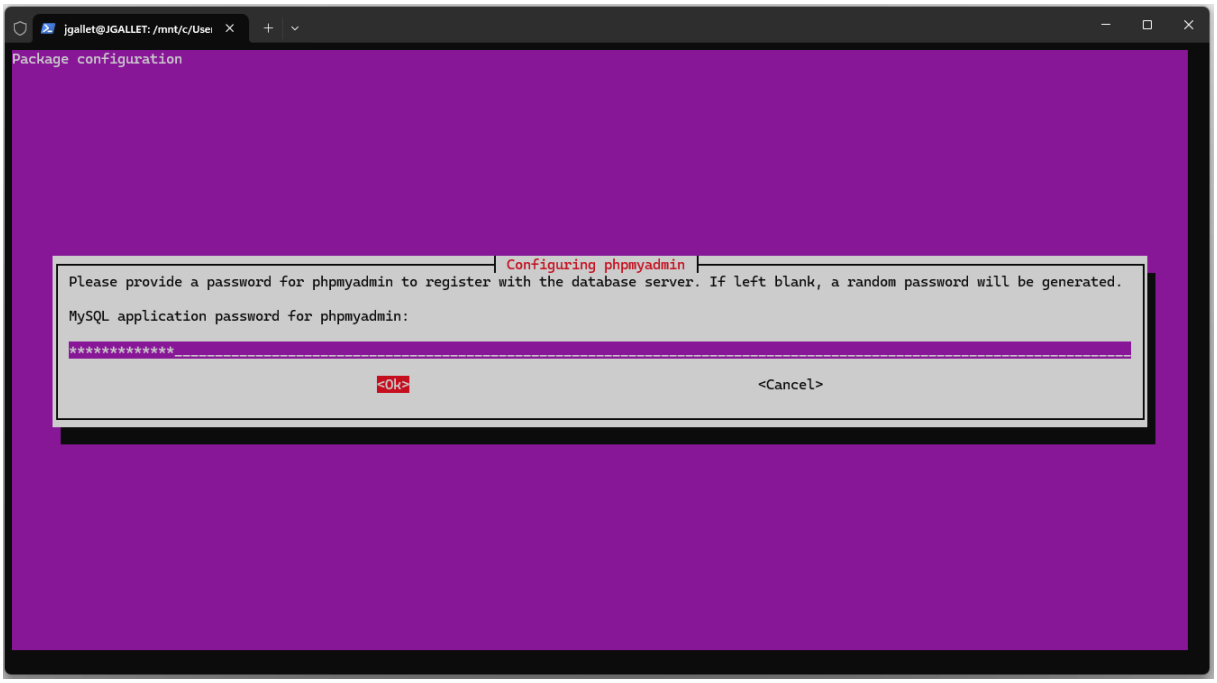
PHPMyAdmin requiert une base de données, l'installeur propose de la configurer automatiquement via `dbconfig-common`, il faut sélectionner « Yes » :



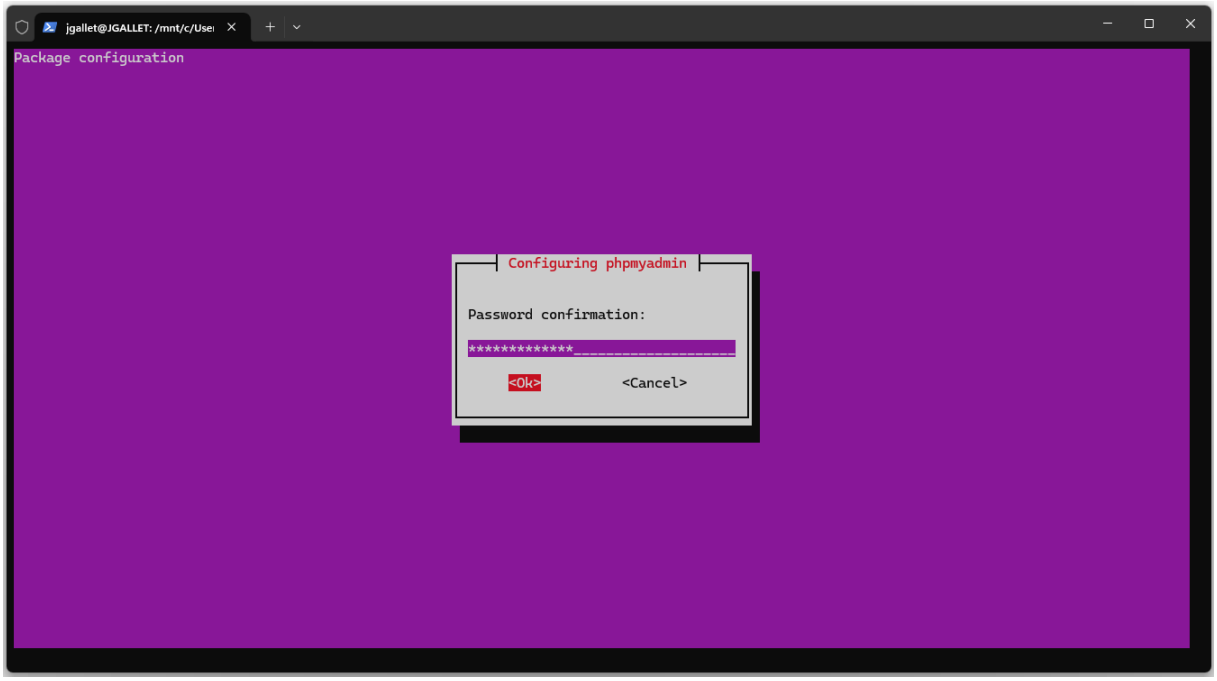
Saisie du mot de passe

Il faut indiquer dans cette étape le mot de passe qui a été définie lors de la création de l'utilisateur `phpmyadmin`. Pour rappel dans notre exemple, il s'agit de « `PhpMy@dm1n` ».

Saisissez ce mot de passe et utilisez la touche tabulation pour sélectionner « OK » puis validez :



Confirmez le mot de passe (idem utilisez la touche tabulation pour sélectionner OK et valider) :



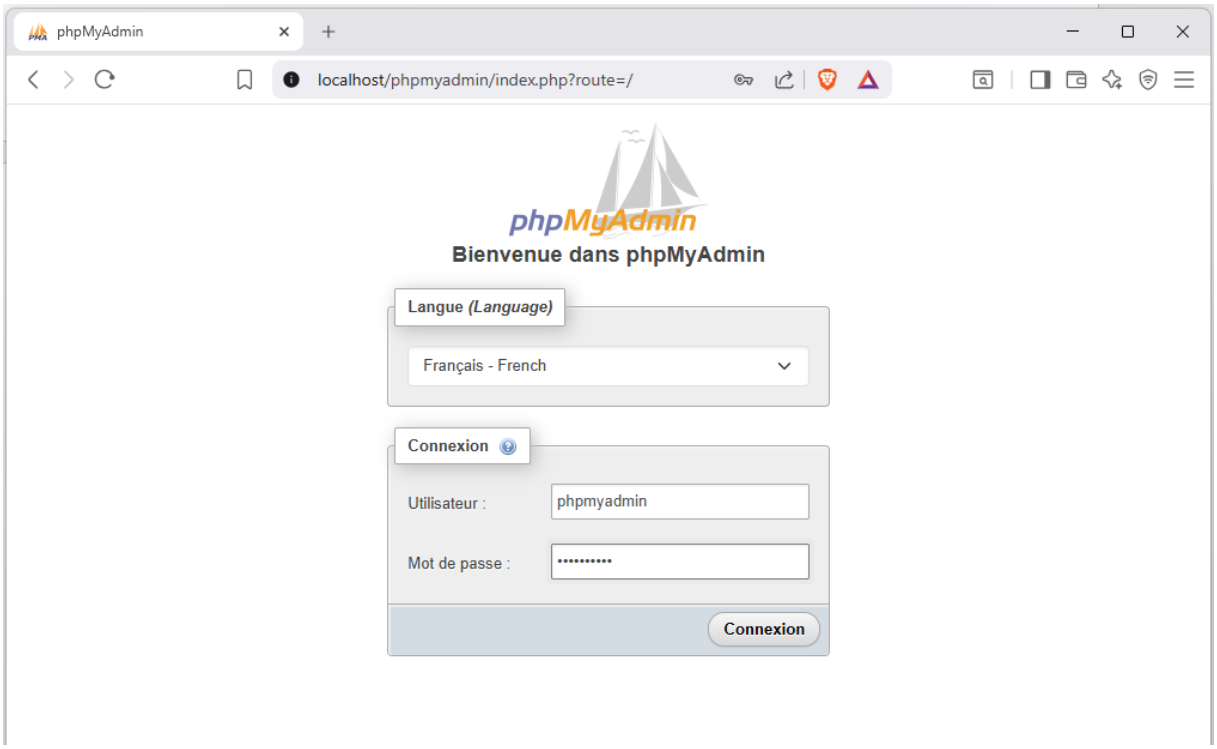
L'installation de PhpMyAdmin se termine.

Vérification

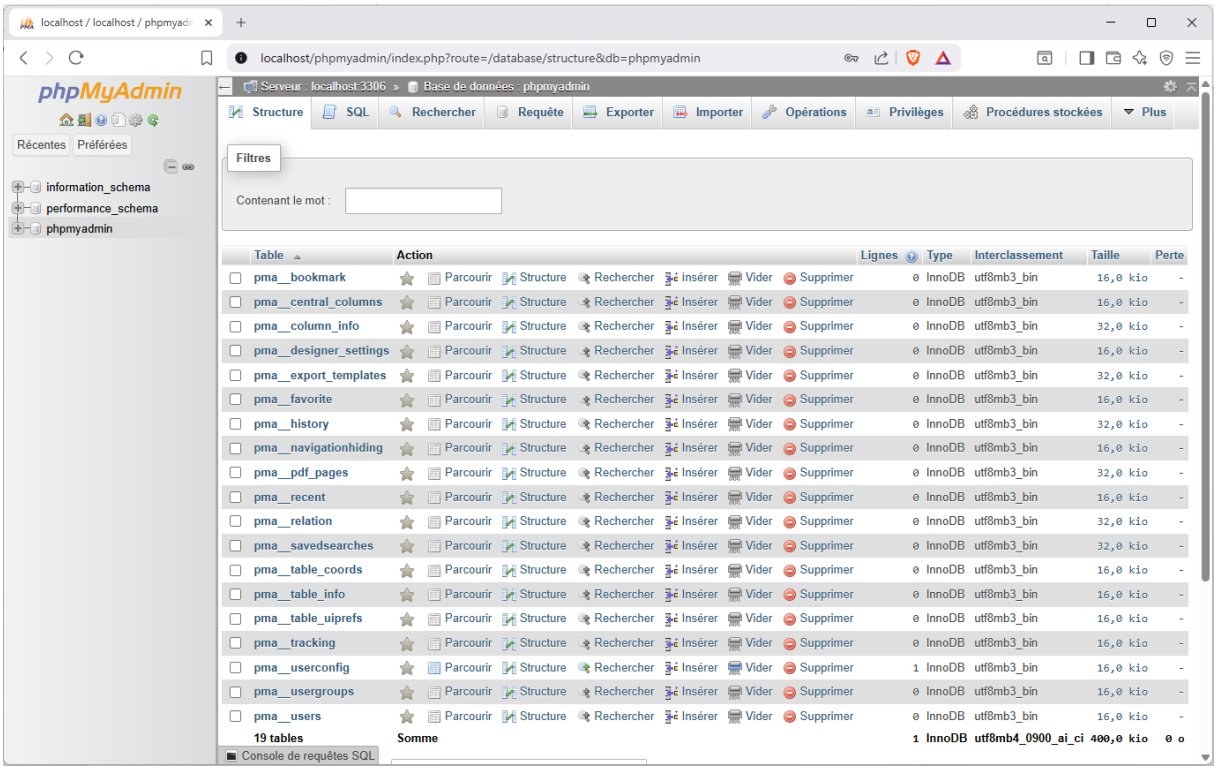
Ouvrez votre navigateur Web et tapez l'URL suivante : http://localhost/phpmyadmin

Utilisateur : `phpmyadmin`

Mot de passe : `PhpMy@dm1n` (ou celui que vous avez défini)



Vous devriez avoir accès aux bases de données déjà installées :



2. CREATION D'UN HOTE VIRTUEL (VHOST)

Lorsque vous mettez en place un serveur web Apache, vous avez la possibilité de créer des hôtes virtuels (virtual hosts). Cela vous permet d'héberger plusieurs sites web sur un même serveur, tout en utilisant des noms de domaine différents.

Par défaut, Apache ne contient qu'un seul hôte virtuel qui est configuré pour desservir les pages contenues dans le répertoire `/var/www/html`. Cela fonctionne pour l'hébergement d'un seul site, comme nous avons pu le tester dans la partie précédente avec le fichier `info.php`, mais cela complexifie la mise en place, la maintenance et la sécurité lorsque vous souhaitez ajouter d'autres sites.

Dans cette partie nous allons créer une structure de répertoires différente au sein de `/var/www` qui sera dédiée à votre site web.

Vous pouvez nommer votre site avec le nom de domaine que vous souhaitez. Pour la suite de ces explications nous prendrons comme exemple « ajob4u.fr ». Si vous optez pour un autre nom, ne prenez pas un nom qui existe déjà sur Internet sinon lorsque vous saisirez l'URL dans le navigateur il pointera vers ce site et non celui hébergé en local (à moins de changer dans votre fichier host local, ce que nous verrons par la suite).

MISE EN PLACE

Pour cet exemple nous allons partir du principe que le site sera localisé dans le répertoire `/var/www/ajob4u.fr`. Libre à vous de choisir un autre emplacement si vous le souhaitez.

Création du répertoire

Dans le terminal, tapez la commande suivante pour créer le répertoire racine de votre site :

```
$ sudo mkdir /var/www/ajob4u.fr
```

Vous pouvez vérifier la création et les droits du répertoire via la commande :

```
$ ls -la /var/www
```

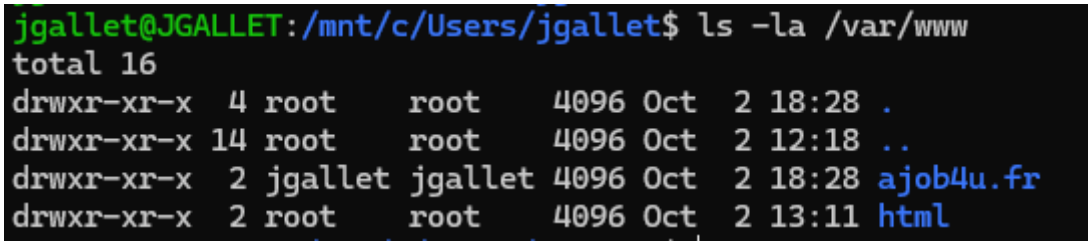
```
jgallet@JGALLET:/mnt/c/Users/jgallet$ ls -la /var/www
total 16
drwxr-xr-x  4 root root 4096 Oct  2 18:28 .
drwxr-xr-x 14 root root 4096 Oct  2 12:18 ..
drwxr-xr-x  2 root root 4096 Oct  2 18:28 ajob4u.fr
drwxr-xr-x  2 root root 4096 Oct  2 13:11 html
```

Vous pouvez voir ici que le répertoire est assigné à l'utilisateur root. Si vous souhaitez que l'utilisateur courant soit propriétaire, vous pouvez changer les droits d'accès :

```
$ sudo chown -R $USER:$USER /var/www/ajob4u.fr
```

Puis vérifiez que le changement a été pris en compte :

```
$ ls -la /var/www
```



Création du fichier de configuration du VirtualHost

Les fichiers de configuration d'Apache sont localisés dans le répertoire `/etc/apache2.`

Le sous-répertoire `sites-enabled` contient les fichiers de configuration de chaque hôte virtuel.

Dans le terminal, tapez la commande suivante pour éditer un nouveau fichier de configuration Apache relatif à votre domaine :

```
$ sudo nano /etc/apache2/sites-available/ajob4u.fr.conf
```

Cela aura pour effet d'ouvrir l'éditeur `nano` dans lequel vous devrez ajouter les lignes suivantes :

```
<VirtualHost *:80>
  ServerName ajob4u.fr
  ServerAlias www.ajob4u.fr
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/ajob4u.fr

  <Directory /var/www/ajob4u.fr>
    Options -Indexes +FollowSymLinks
    AllowOverride None
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/ajob4u_error.log
  CustomLog ${APACHE_LOG_DIR}/ajob4u_access.log combined
</VirtualHost>
```

Quitter et enregistrer votre nouveau fichier (`CTRL+X` suivi de `y+entrée`).

Cette configuration permet d'utiliser les ressources localisées dans le répertoire `/var/www/ajob4u.fr` lorsque l'utilisateur saisira http://ajob4u.fr ou http://www.ajob4u.fr dans l'URL de son navigateur.

ⓘ Veuillez bien prendre connaissance de toutes les directives et savoir à quoi elles correspondent.

A chaque modification du fichier VirtualHost, il faut qu'Apache soit redémarré pour prendre en compte les modifications :

```
$ sudo systemctl reload apache2
```

Vérification du fichier de configuration

Dans le terminal, tapez la commande suivante pour vérifier que la syntaxe du fichier de configuration soit correcte :

```
$ sudo apachectl configtest
```

Il peut y avoir un message d'avertissement relatif au ServerName mais cela n'est pas gênant à partir du moment où il est mentionné « `Syntax OK` ».

Activation du Virtual Host

Dans le terminal, tapez la commande suivante pour activer l'hôte virtuel via `a2ensite` :

```
$ sudo a2ensite ajob4u.fr
```

Vous pouvez en profiter pour désactiver le site par défaut (celui contenu dans le répertoire `/var/www/html`) via la commande `a2dissite` :

```
$ sudo a2dissite 000-default
```

Redémarrez ensuite le serveur Apache pour prendre en compte la configuration :

```
$ sudo systemctl reload apache2
```

Déplacement du fichier php.info

Dans le terminal, tapez la commande suivante pour déplacer le fichier `info.php` créé précédemment dans le répertoire correspondant à votre site :

```
$ sudo mv /var/www/html/info.php /var/www/ajob4u.fr
```

Edition du fichier host

Votre site est activé mais si vous tapez dans l'URL du navigateur l'adresse configurée, rien ne s'affichera. Cela est normal car le domaine ne devrait pas exister (à moins que vous ayez choisi un nom de domaine existant, le cas échéant ce sera le site hébergé sur Internet qui sera affiché).

Pour que vous puissiez utiliser ce nom de domaine en local, il va falloir ajouter une entrée dans le fichier de configuration des hôtes locaux. Ce fichier est appelé avant la requête émise vers les serveurs DNS

qui, pour rappel, permettent de résoudre les noms de domaines en adresse IP. Ainsi vous pouvez soit ajouter une entrée si elle n'existe pas dans les serveurs DNS configurés, soit écraser une valeur existante.

Votre serveur Web sous WSL étant accessible uniquement en lignes de commandes (aucune couche graphique n'a été installée), vous vous appuyez sur le navigateur de votre machine hôte pour atteindre votre site Internet. Le principe consiste donc à modifier le fichier host de votre machine Windows et non celui de l'environnement Linux sur lequel est installé votre serveur Web.

Le fichier hosts de Windows se trouve à cet emplacement :

`C:\Windows\System32\drivers\etc\hosts`

Ouvrez ce fichier avec les droits administrateurs de votre éditeur de texte sous Windows et ajouter les lignes suivantes :

```
#IPV4
127.0.0.1    ajob4u.fr
127.0.0.1    www.ajob4u.fr

#IPV6
::1    ajob4u.fr
```

Enregistrez le fichier une fois les modifications apportées.

ⓘ Si vous avez une couche graphique dans votre environnement Linux et que vous souhaitez utiliser le navigateur du sous-système pour accéder à votre site Internet vous devriez modifier le fichier hosts localisé dans `/etc/hosts`.

Vérification des accès

Ouvrez un navigateur et saisissez l'URL du site que vous avez configuré.

Vous devriez arriver sur une page de ce type :

La configuration de l'hôte virtuel fonctionne, mais vous voyez qu'Apache affiche une liste de fichiers et non un site Web. Cela est normal, car par défaut le serveur Web cherche un fichier nommé `index.html` ou `index.php` à la racine du site.

Vous pouvez :

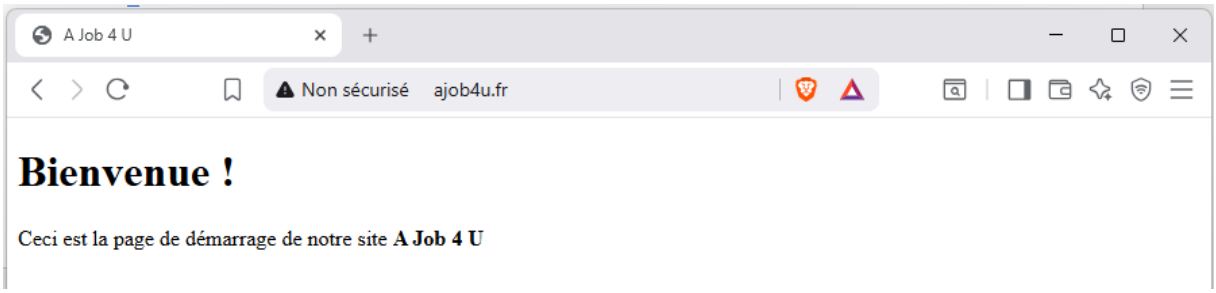
- soit renommer le fichier `info.php` en `index.php` pour vérifier temporairement le fonctionnement :

```
$ sudo mv /var/www/ajob4u.fr/info.php /var/www/ajob4u.fr/index.php
```

- soit créer un fichier `index.html` (ou `index.php`) que vous agrémenterez avec le contenu de votre choix :

```
$ sudo nano /var/www/ajob4u.fr/index.php
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>A Job 4 U</title>
  </head>
  <body>
    <h1>Bienvenue !</h1>
    <p>Ceci est la page de démarrage de notre site <strong>A Job 4 U</strong></p>
  </body>
</html>
```



3. MISE EN PLACE DU HTTPS

Le serveur Web est fonctionnel. Vous allez à présent ajouter une couche de sécurité en implémentant le protocole HTTPS qui chiffre les communications grâce à SSL/TLS.

Tous les serveurs Web doivent aujourd'hui être accessible via https pour assurer la sécurité des données qui transitent, donner confiance aux utilisateurs et optimiser le référencement naturel.

CREATION DU CERTIFICAT

Pour la génération du certificat nous allons nous appuyer sur l'outil en ligne de commandes OpenSSL qui permet de créer des certificats auto-signés.

Génération du fichier SAN

Par défaut, OpenSSL ne prend pas en charge les champs SAN (Subject Alternative Name) qui sont indispensables pour être compatible avec les navigateurs Web modernes. Vous allez donc créer en premier lieu un fichier de configuration qui sera ensuite utilisé par OpenSSL :

```
$ sudo nano openssl-san.cnf
```

Insérer le contenu suivant dans ce fichier, vous pouvez modifier les champs de la section `[dn]` comme vous le souhaitez, sauf pour CN que vous devez conserver ainsi :

```
[req]
default_bits       = 2048
prompt             = no
default_md         = sha256
req_extensions     = req_ext
distinguished_name = dn

[dn]
C  = FR
ST = Loire-Atlantique
L  = Saint-Nazaire
O  = A Job 4 U
CN = ajob4u.fr

[req_ext]
subjectAltName = @alt_names

[alt_names]
DNS.1 = ajob4u.fr
DNS.2 = www.ajob4u.fr
```

Quitter et enregistrer votre nouveau fichier (`CTRL+X` suivi de `y+entrée`).

Génération du certificat SSL

Il faut à présent générer le certificat auto-signé.

Dans le terminal, tapez les commandes suivantes :

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ajob4u.fr.key
-out ajob4u.fr.crt
```

Déplacement des fichiers

Les fichiers `ajob4u.fr.crt` et `ajob4u.fr.key` ont été générés à l'endroit où vous avez exécuté la commande.

Ils ne doivent surtout pas être positionnés dans le même répertoire que votre site Web qui est accessible au public. Déplacez les fichiers `ajob4u.fr.crt` et `ajob4u.fr.key` respectivement dans les répertoires `/etc/ssl/certs/` et `/etc/ssl/private` :

```
$ sudo mv ajob4u.fr.crt /etc/ssl/certs/
$ sudo mv ajob4u.fr.key /etc/ssl/private/
```

CREATION DU VHOST HTTPS

Il faut maintenant créer un hôte virtuel qui va écouter sur le port 443 du serveur Web (port HTTPS par défaut) et pointer vers les fichiers ssl :

```
$ sudo nano /etc/apache2/sites-available/ajob4u.fr-ssl.conf
```

Ajoutez les définitions suivantes dans le fichier :

```
<VirtualHost *:443>
  ServerName ajob4u.fr
  DocumentRoot /var/www/ajob4u.fr/
  SSLEngine on
  SSLCertificateFile "/etc/ssl/certs/ajob4u.fr.crt"
  SSLCertificateKeyFile "/etc/ssl/private/ajob4u.fr.key"
  <Directory /var/www/ajob4u.fr>
     Options Indexes FollowSymLinks
     AllowOverride All
     Require all granted
  </Directory>
  ErrorLog ${APACHE_LOG_DIR}/ajob4u_error.log
  CustomLog ${APACHE_LOG_DIR}/ajob4u_access.log combined
</VirtualHost>
```

Quittez et enregistrez votre nouveau fichier (`CTRL+X` suivi de `y+entrée`).

Activation du Virtual Host

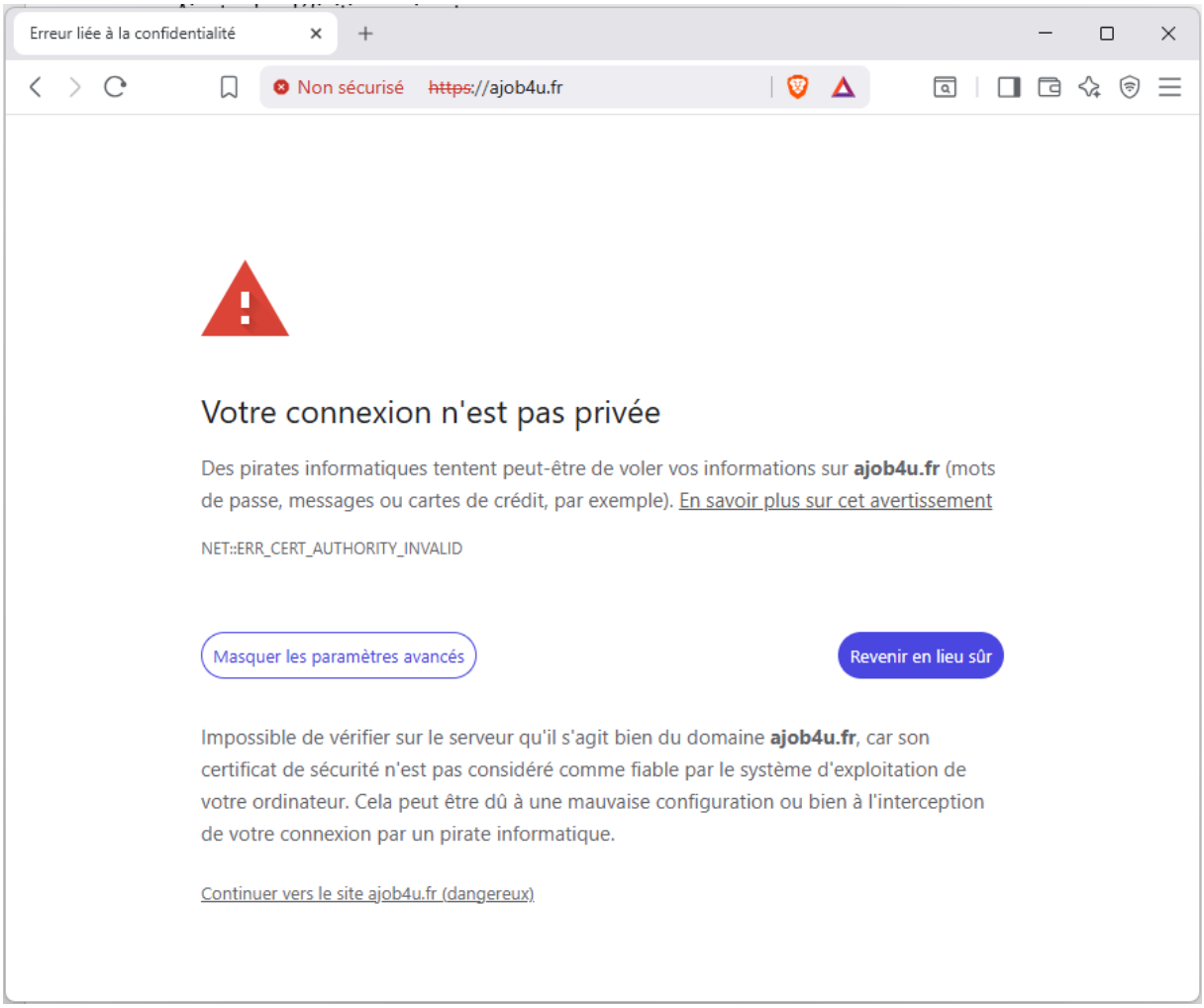
```
$ sudo a2ensite ajob4u.fr-ssl.conf
```

Redémarrage d'Apache

```
$ sudo systemctl reload apache2
```

VERIFICATION

Le certificat est à présent en place, vous pouvez vérifier que le HTTPS a bien été implémenté sur votre site en saisissant https://ajob4u.fr dans votre navigateur :



Vous remarquerez que même si le certificat a été mis en place, le navigateur indique que la connexion n'est pas privée car il considère que le certificat est invalide.

Ceci est un comportement normal car il s'agit d'un certificat auto-signé et non délivré par une autorité de confiance.

Dans un environnement de production, des certificats du type *Let's Encrypt* peuvent être utilisés. Pour notre environnement local, nous allons installer ce certificat sous Windows pour qu'il le considère valide.

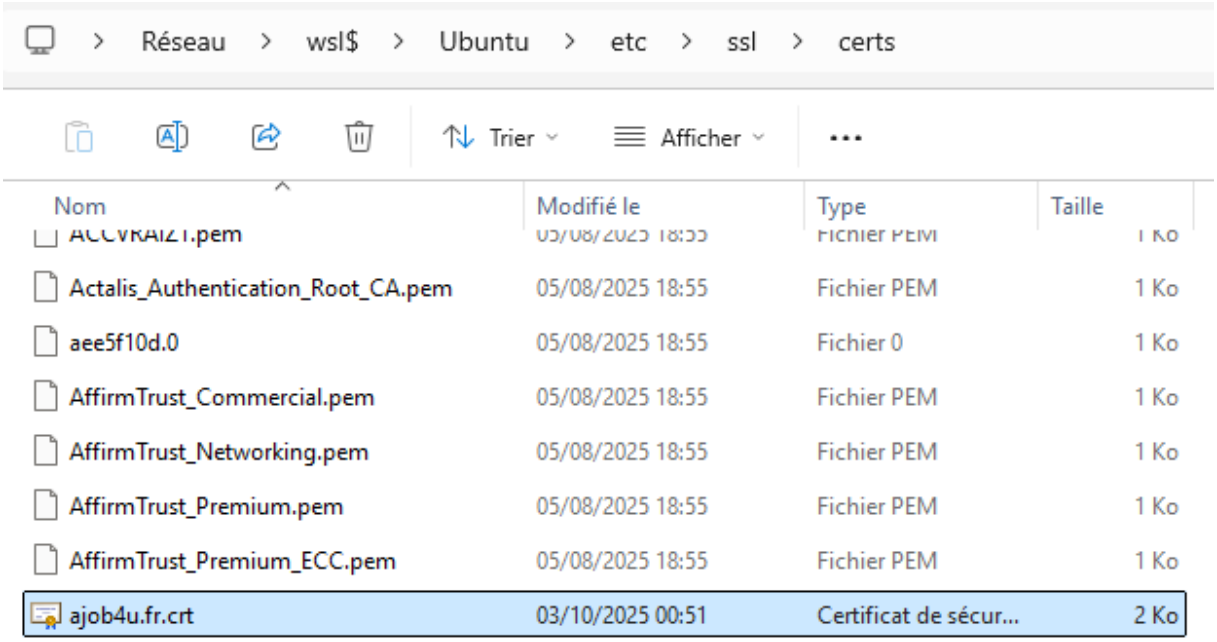
INSTALLATION DU CERTIFICAT SOUS WINDOWS

Le principe consiste à ajouter le certificat généré par OpenSSL au magasin de confiance de Windows.

Accès au certificat

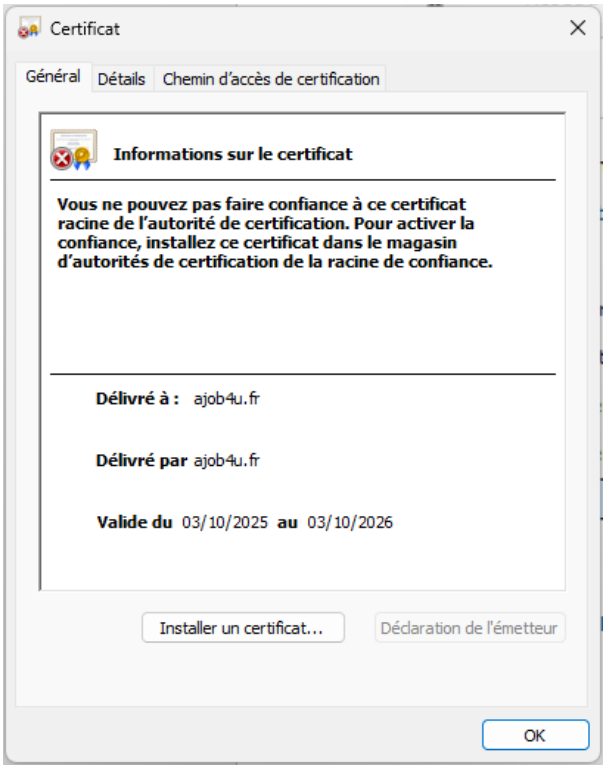
Dans l'explorateur Windows, ouvrez l'emplacement où vous avez stocké le certificat SSL et localisez votre fichier. Pour accéder à l'arborescence de votre sous-système Linux, utilisez le chemin suivant :

`\\wsl$\Ubuntu\etc\ssl\certs`

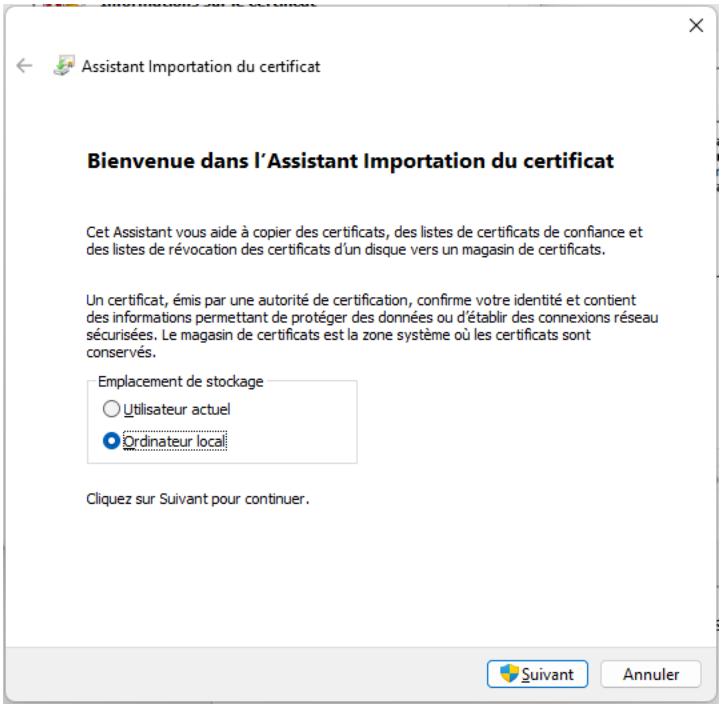


Ajout au magasin de confiance

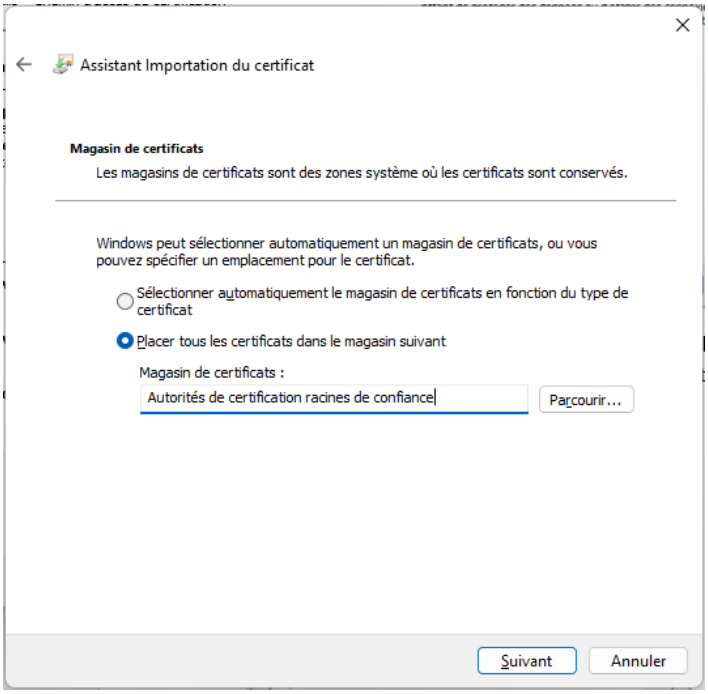
Double-cliquez sur le fichier `ajob4u.fr.crt`, la fenêtre suivante apparaît :



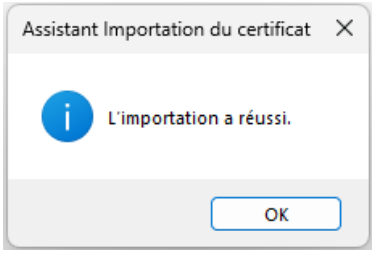
Cliquez sur "Installer un certificat" et sélectionnez "Ordinateur Local" dans l'emplacement de stockage (deuxième option) puis cliquez sur Suivant :



Dans le magasin de certificats sélectionnez "Placer tous les certificats dans le magasin suivant" (deuxième option) puis cliquez sur "Parcourir" et sélectionnez « Autorités de certification racines de confiance » :

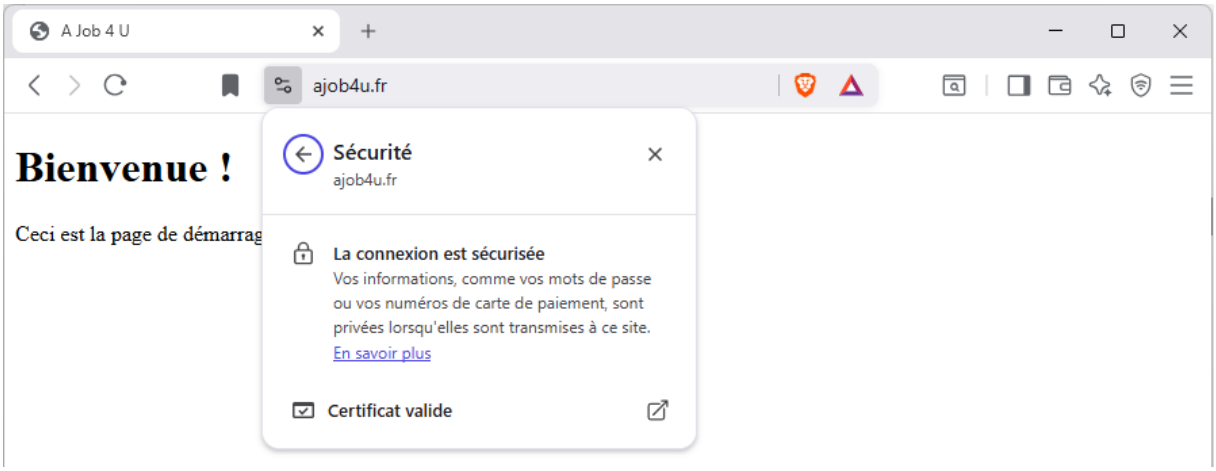


Cliquez sur « Suivant » puis « Terminer ».



Vérification

Vous pouvez maintenant rafraîchir la page ou ouvrir de nouveau votre site, le message d'avertissement a disparu et vous avez un site fonctionnel et sécurisé :



4. UTILISATION DU FICHIER .HTACCESS

Le fichier .htaccess est un fichier de configuration utilisé par Apache pour appliquer des règles locales à un répertoire spécifique.

Il peut contenir de nombreuses directives pour ajouter ou remplacer la configuration par défaut du serveur Apache, notamment :

- Contrôler les accès (authentification, interdiction d'accès)
- Rediriger des URL (ex. redirection vers un contenu sécurisé ou une nouvelle page)
- Réécrire des URL (ex. transformer article.php?id=3 en /article/3)
- Définir des pages d'erreur personnalisées (404, 403, etc.)
- Activer ou désactiver des fonctionnalités Apache (comme mod_rewrite)
- Protéger des dossiers (avec mot de passe via .htpasswd)

REDIRECTION D'URL

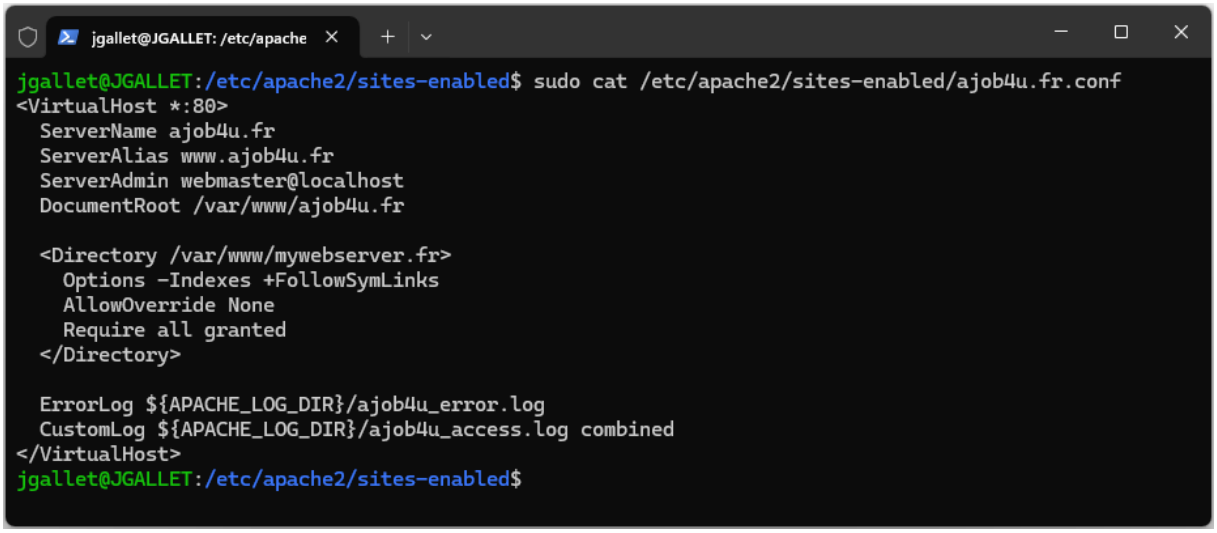
Nous allons utiliser ce fichier pour créer une redirection d'une page vers une autre.

Vérification de la configuration

Il faut avant tout s'assurer que le paramètre `AllowOverride All` est bien présent dans le fichier de configuration du site.

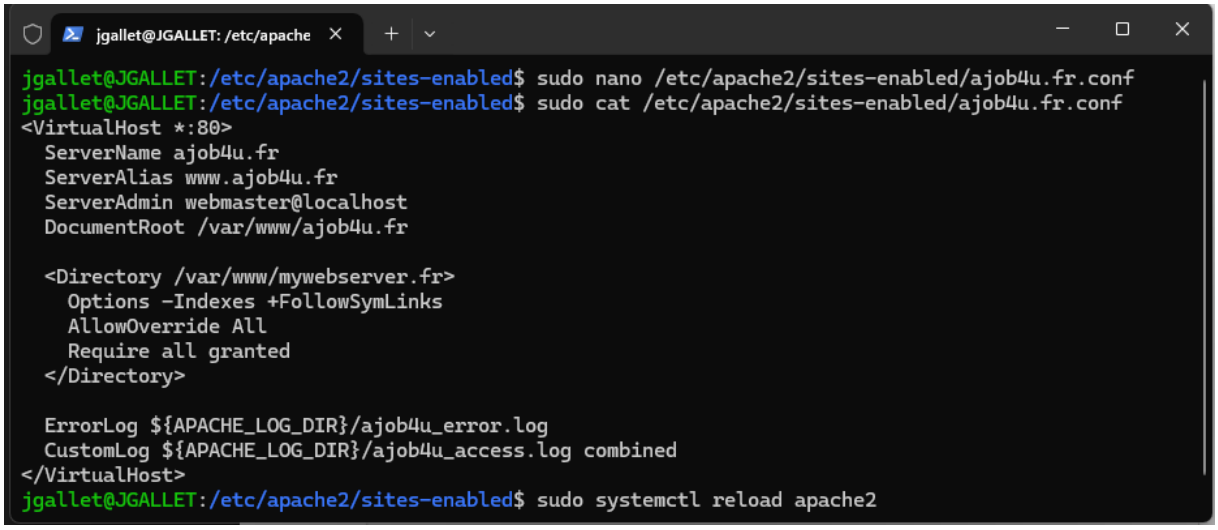
Affichez le contenu du site non sécurisé via la commande suivante :

```
$ sudo cat /etc/apache2/sites-enabled/ajob4u.fr.conf
```



Dans la configuration que nous avons effectuée précédemment, le paramètre `AllowOverride` est à None. Cela signifie qu'Apache n'autorise aucune surcharge de paramètre (les directives effectuées dans le fichier .htaccess seront ignorées).

Editez le fichier de configuration, changez le paramètre `AllowOverride à All` puis redémarrez le serveur Apache pour prendre en compte les changements :



Création d'une page HTML

Pour tester la redirection de page, vous allez créer une nouvelle page.

A la racine de votre site, créez puis éditez un fichier que vous nommerez « `home.html` » :

```
$ sudo nano home.html
```

Vous pouvez y intégrer un bout de code HTML pour test :

```
<!DOCTYPE html>
<html>
  <meta charset="UTF-8">
  <head>
    <title>Accueil – A Job 4 U</title>
  </head>
  <body>
    <h1>Accueil</h1>
    <p>Ceci est la page d'accueil de notre site <strong>A Job 4 U</strong></p>
  </body>
</html>
```

Quittez et enregistrez votre nouveau fichier (`CTRL+X` suivi de `y+entrée`).

Création du fichier .htaccess

Toujours à la racine de votre site, créez un fichier nommé simplement « `.htaccess` » :

```
$ sudo touch .htaccess
```

Ajout d'une redirection simple

Editez ce fichier puis ajoutez-y la commande suivante qui va permettre de rédiger vers la page d'accueil créée précédemment en utilisant un code de statut de réponse 301 (Moved Permanently) :

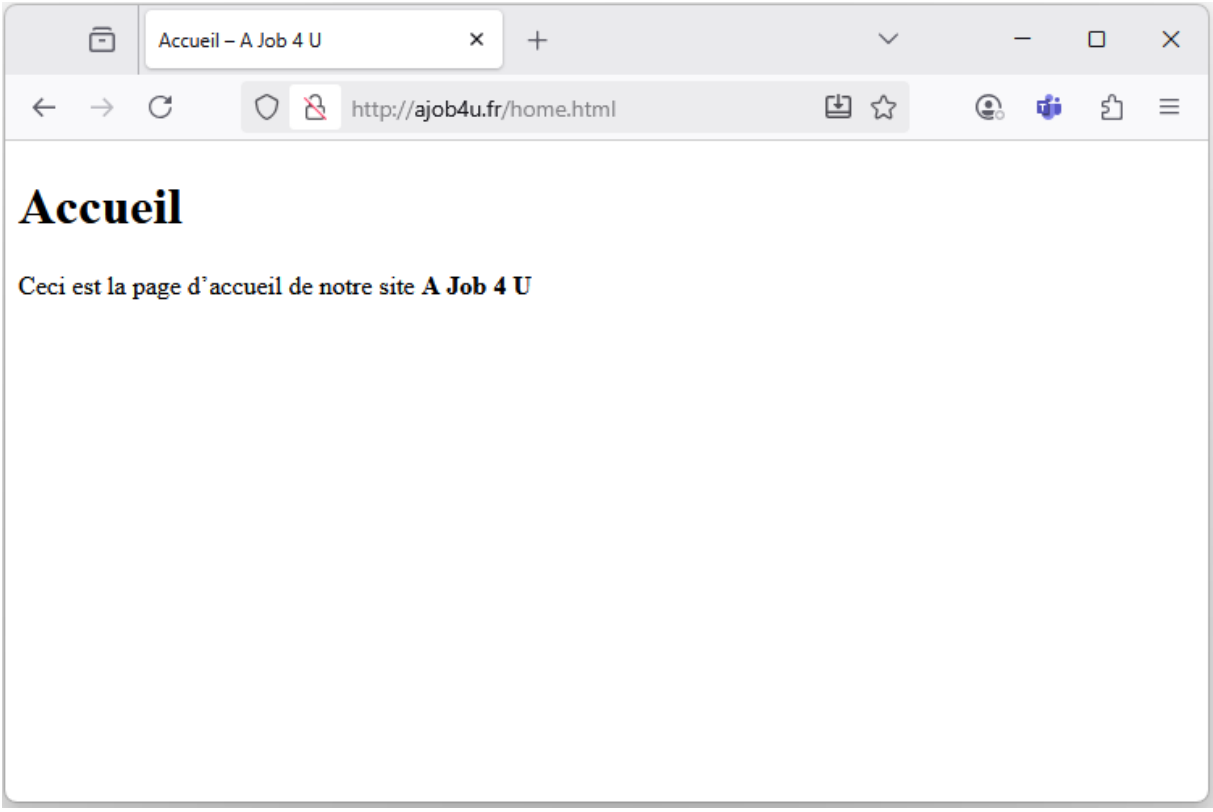
```
$ sudo nano .htaccess
```

```
Redirect 301 /index.php /home.html
```

Quittez et enregistrez (`CTRL+X` suivi de `y+entrée`).

Test de la redirection

Dans votre navigateur ouvrir l'URL http://ajob4u.fr ou rafraichissez la page. Vous devriez être redirigé automatiquement vers la page `home.html` :



Réécritures d'URL avancées

La méthode précédente permet de spécifier la redirection d'une page vers une autre. Cette méthode peut vite être limitée, notamment s'il y a de nombreuses pages à rediriger ou que le nom de la page n'est pas connu.

Modification du fichier .htaccess

Ouvrez et modifier le fichier .htaccess afin d'utiliser le mode de réécriture avancé et de rediriger la page :

```
RewriteEngine On
RewriteRule ^index\.php$ /home.html [R=301,L]
```

Quittez et enregistrez (`CTRL+X` suivi de `y+entrée`).

ⓘ Veuillez à vous familiariser avec la syntaxe du mode d'écriture au format regex ainsi que de connaître les codes de redirections (entre crochets).

Activation du mod_rewrite

Si ce n'est déjà fait, il faut demander à Apache d'activer le mode de réécriture avancé (`mod_rewrite`). Un redémarrage du serveur Web sera nécessaire :

```
$ sudo a2enmod rewrite
$ sudo systemctl restart apache2
```

Test

Dans votre navigateur, lancer l'URL http://ajob4u.fr/index.php

Vous devriez être redirigé vers la page http://ajob4u.fr/home.html

Redirection du trafic

Le mode de réécriture avancé peut également être utilisé pour rediriger tout le trafic HTTP vers HTTPS.

Cela assure d'utiliser toujours la version sécurisée du site, quelle que soit l'URL appelée.

Modification du fichier .htaccess

Modifier votre fichier `.htaccess` afin d'y intégrer les directives suivantes :

```
# Activation du moteur de réécriture
RewriteEngine On

# Vérification si la requête est HTTP
RewriteCond %{HTTPS} off

# Redirection vers la même URL, en HTTPS
RewriteRule ^(.*)$ https://%{HTTP_HOST}/$1 [R=301,L]
```

Quittez et enregistrez (`CTRL+X` suivi de `y+entrée`).

Test

Dans votre navigateur, lancer l'URL http://ajob4u.fr/home.html

Vous devriez être redirigé vers https://ajob4u.fr/home.html

5. ANALYSE ET INTERPRETATION DES LOGS D'APACHE

Apache génère des logs en fonction du trafic ou des erreurs qui pourraient survenir lors de la navigation (page introuvable, erreur serveur…). Ce sont des informations essentielles pour le diagnostic, la sécurité et l'analyse du trafic sur le serveur Web.

TYPES DE LOGS

Il y a deux types de logs :

- Logs d'accès (`access.log`) : enregistre toutes les requêtes HTTP reçues par le serveur. Contient les adresses IP du client, la méthode (GET/POST), l'URL, le code de réponse… Utile pour analyser le trafic, repérer les pages les plus visitées, ou détecter des robots.
- Logs d'erreurs (`error.log`) : enregistre les erreurs du serveur, les problèmes de configuration, les erreurs du langage de programmation…
  Indispensable pour déboguer les erreurs 500, les problèmes de vhost, ou les permissions.

PERSONNALISATION PAR VHOST

Les logs d'Apache sont localisés par défaut dans le dossier `/var/log/apache2`.

Chaque Virtual Host peut avoir son propre fichier de logs et son propre chemin de logs. Dans un environnement où vous avez plusieurs sites ou clients, il faut différencier les logs.

Ce paramètre s'effectue dans les fichiers vhost, comme nous l'avez fait précédemment via ces directives :

```
ErrorLog ${APACHE_LOG_DIR}/ajob4u_error.log
CustomLog ${APACHE_LOG_DIR}/ajob4u_access.log combined
```

ANALYSE DES LOGS

Vous devez certainement avoir du contenu qui s'est rempli dans les fichiers de logs.

Logs d'accès

Affichez le fichier de logs d'accès de votre site puis analysez le contenu (codes HTTP, clients…) :

```
$ sudo cat /var/log/apache2/ajob4u_access.log
```

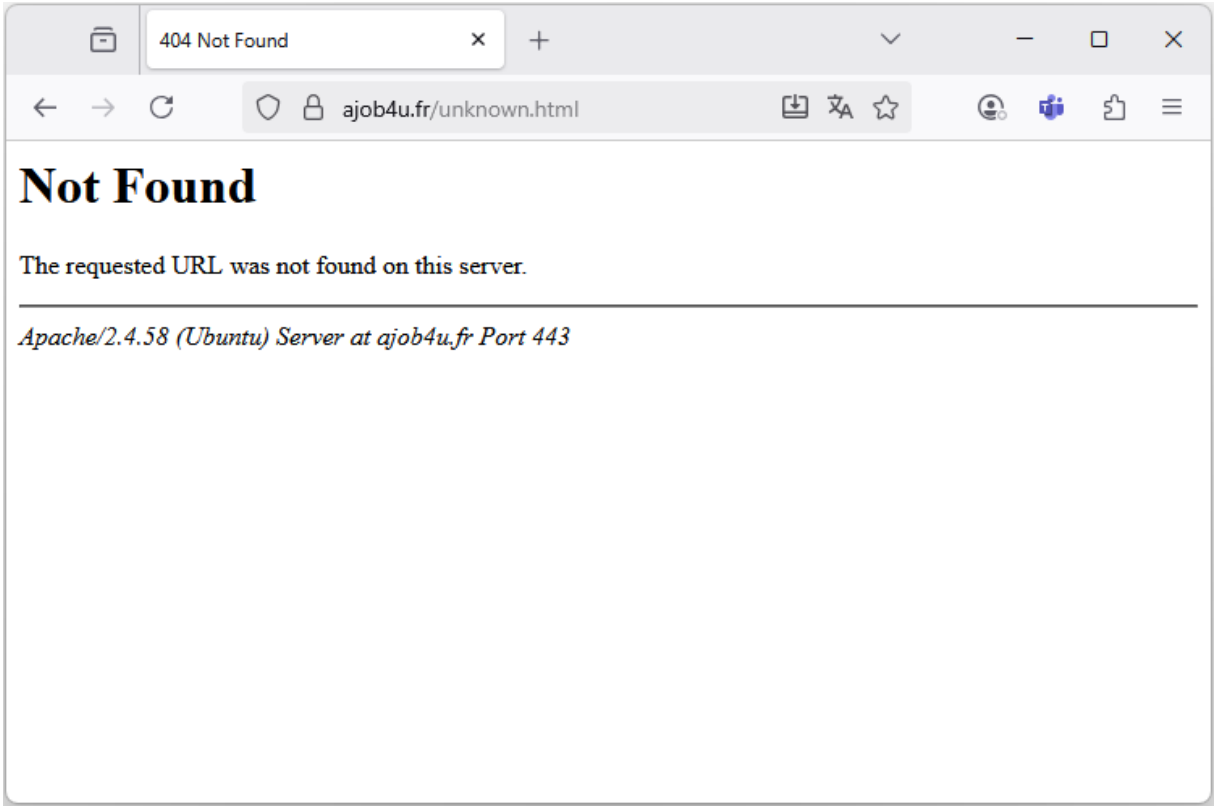
Vous pouvez supprimer ce fichier puis rafraîchir la page de votre navigateur. Une nouvelle entrée devrait apparaître.

Logs d'erreurs

Si tout s'est bien passé jusqu'à présent avec votre serveur Web ou qu'aucune erreur de navigation n'a été faite, ce fichier peut être vide. Nous allons tenter d'accéder à une page qui n'existe pas sur notre serveur Web afin de générer une nouvelle ligne.

Accédez à une page inexistante, par exemple : https://ajob4u.fr/unknown.html

Une page Not Found devrait apparaître dans votre navigateur :



Affichez le fichier d'erreur de votre serveur Web, une nouvelle ligne est apparue :

```
$ sudo cat /var/log/apache2/ajob4u_error.log
```

```
ajob4u.fr:443 ::1 - - [03/Oct/2025:17:12:38 +0200] "GET / HTTP/1.1" 200 756 "-" "Mozilla/5.0 (Windo
ws NT 10.0; Win64; x64; rv:143.0) Gecko/20100101 Firefox/143.0"
ajob4u.fr:443 ::1 - - [03/Oct/2025:17:40:46 +0200] "GET /unknown.html HTTP/1.1" 404 2164 "-" "Mozil
la/5.0 (Windows NT 10.0; Win64; x64; rv:143.0) Gecko/20100101 Firefox/143.0"
```

6. CONCLUSION

Vous avez à présent un environnement Web complet et sécurisé sous Linux qui pourra contenir vos sites Internet.

Vous avez appris à utiliser les redirections et comment analyser les logs Apache.

Votre serveur Web est prêt pour les prochaines séquences !