TP3
LAMP

Sommaire

I/ Installation de LAMP 2
II/ Tests 2
III/ Utilisation 4
1) Installation d'un CMS (Content Management System / WordPress) 6
2) Prise en main de l'administration WordPress 14
3) Créer un premier article WordPress 22
IV/ Apache2 32
1) Analyser de trame 34
2) Configuration Apache 36
3) Scripts CGI simples 44
4) Protection des accès 47
5) Serveurs virtuels 51
6) HTTPS : le http sécurisé 53
V/ Utilisation de MySQL 58
1) Introduction 58
2) Installation 58
3) Fonctionnement 60
4) L'utilitaire MySQL 60
5) Lien avec PHP 63
6) Administration MySQL 66

I/ Installation de LAMP

LAMP est un acronyme pour :

- Linux (le système)
- Apache (le serveur web)
- MySQL (le serveur SQL) ou MariaDB
- PHP (le langage de développement)

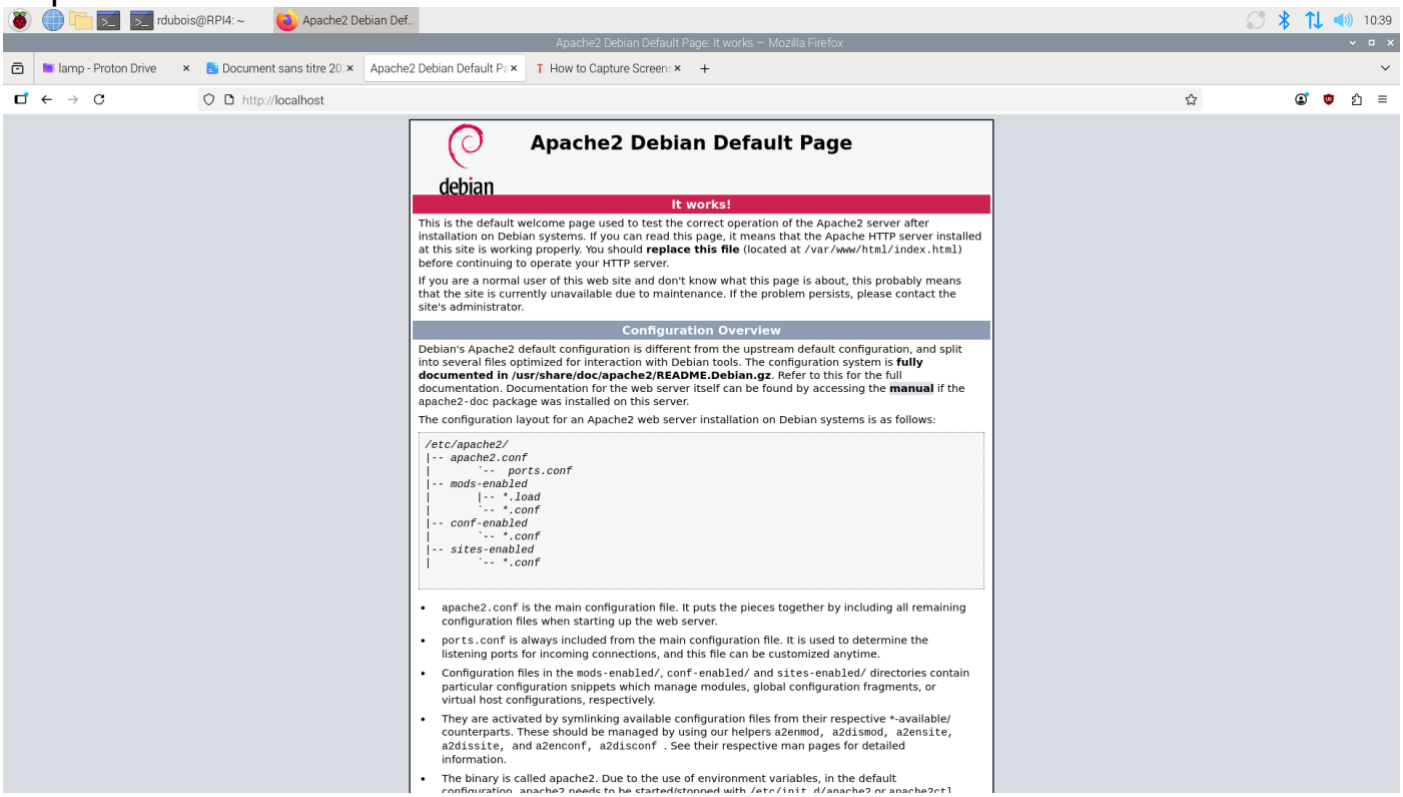
```
sudo apt update
sudo apt upgrade
sudo apt install apache2 php mariadb-server libapache2-mod-php php-mysql
```

Démarrer les services

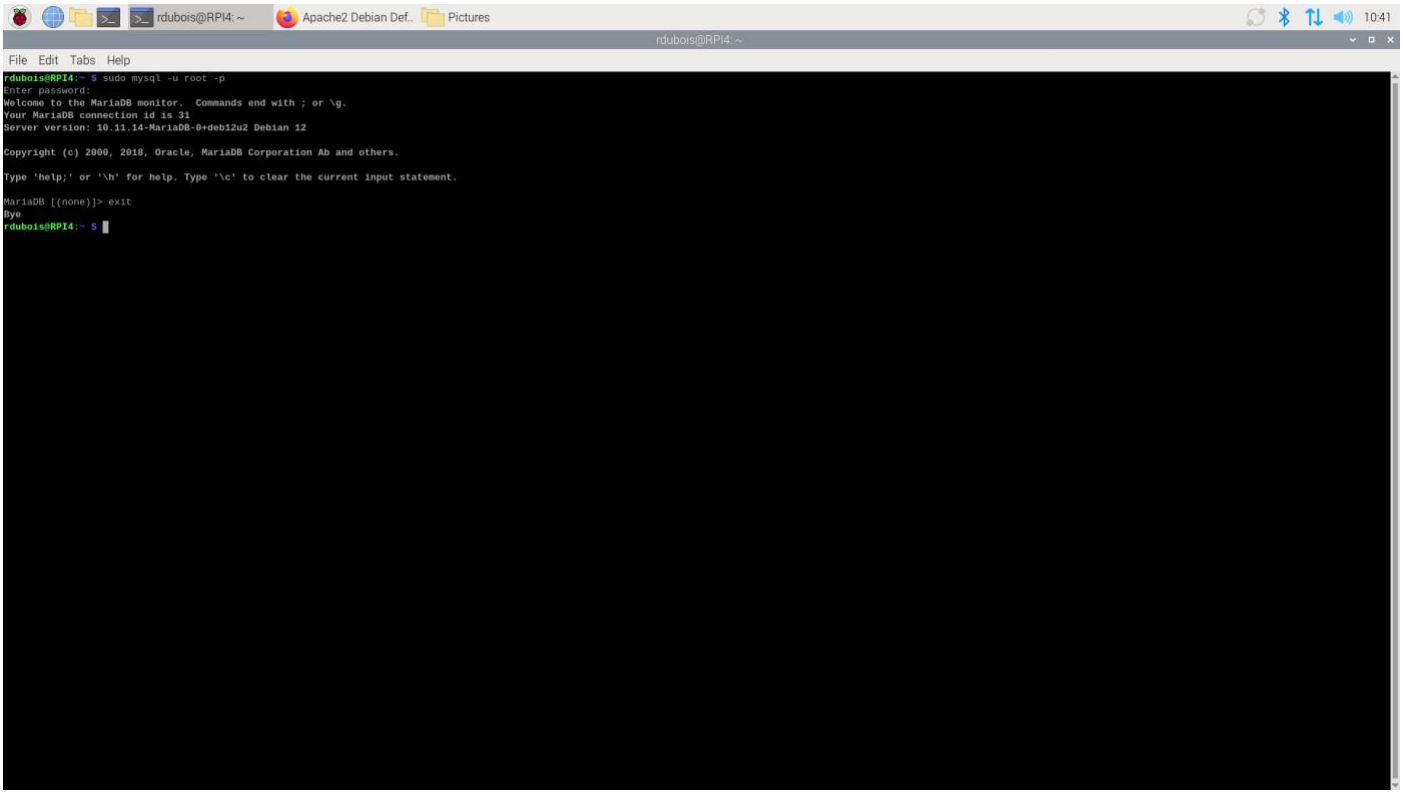
```
systemctl start apache2
systemctl start mysqld
```

II/ Tests

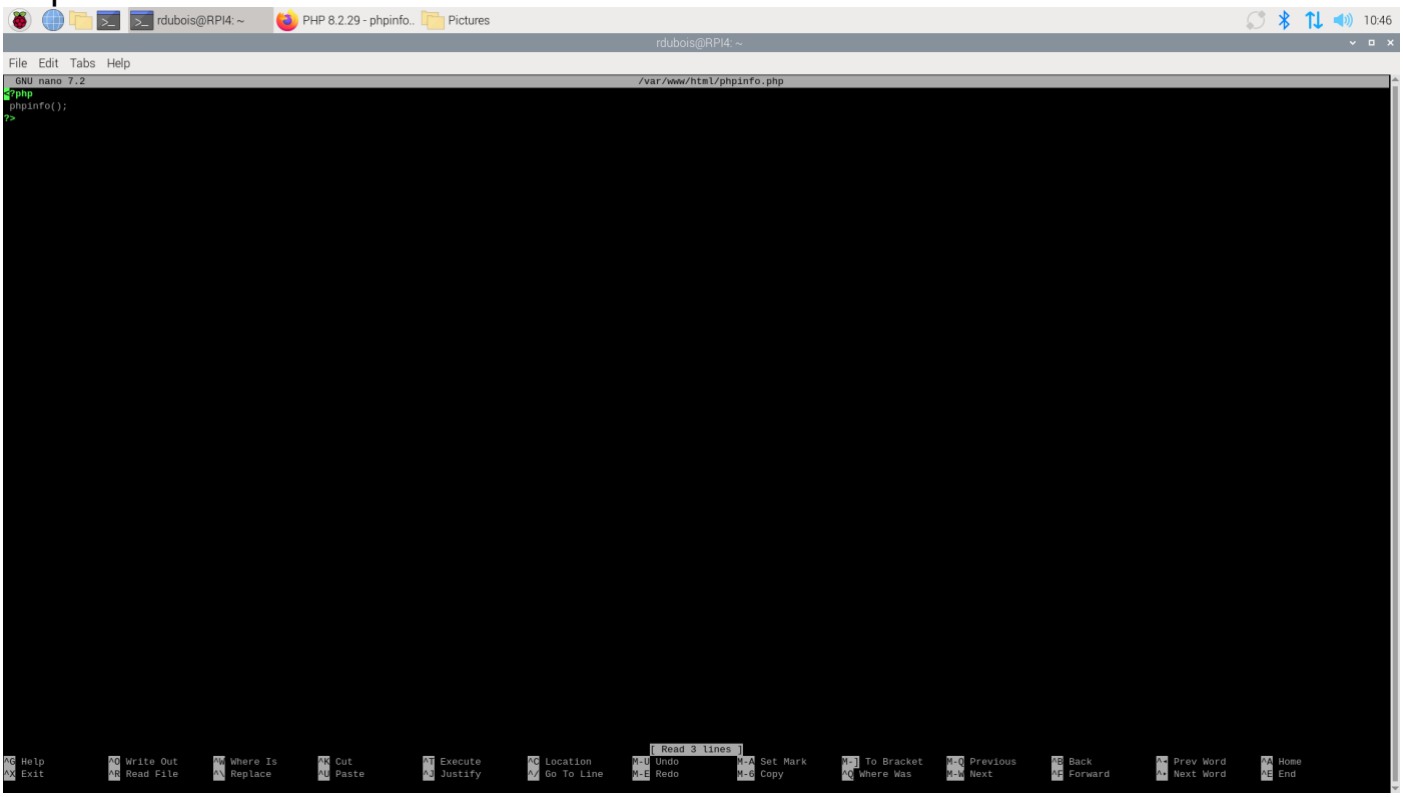
Navigateur : http://localhost It work!

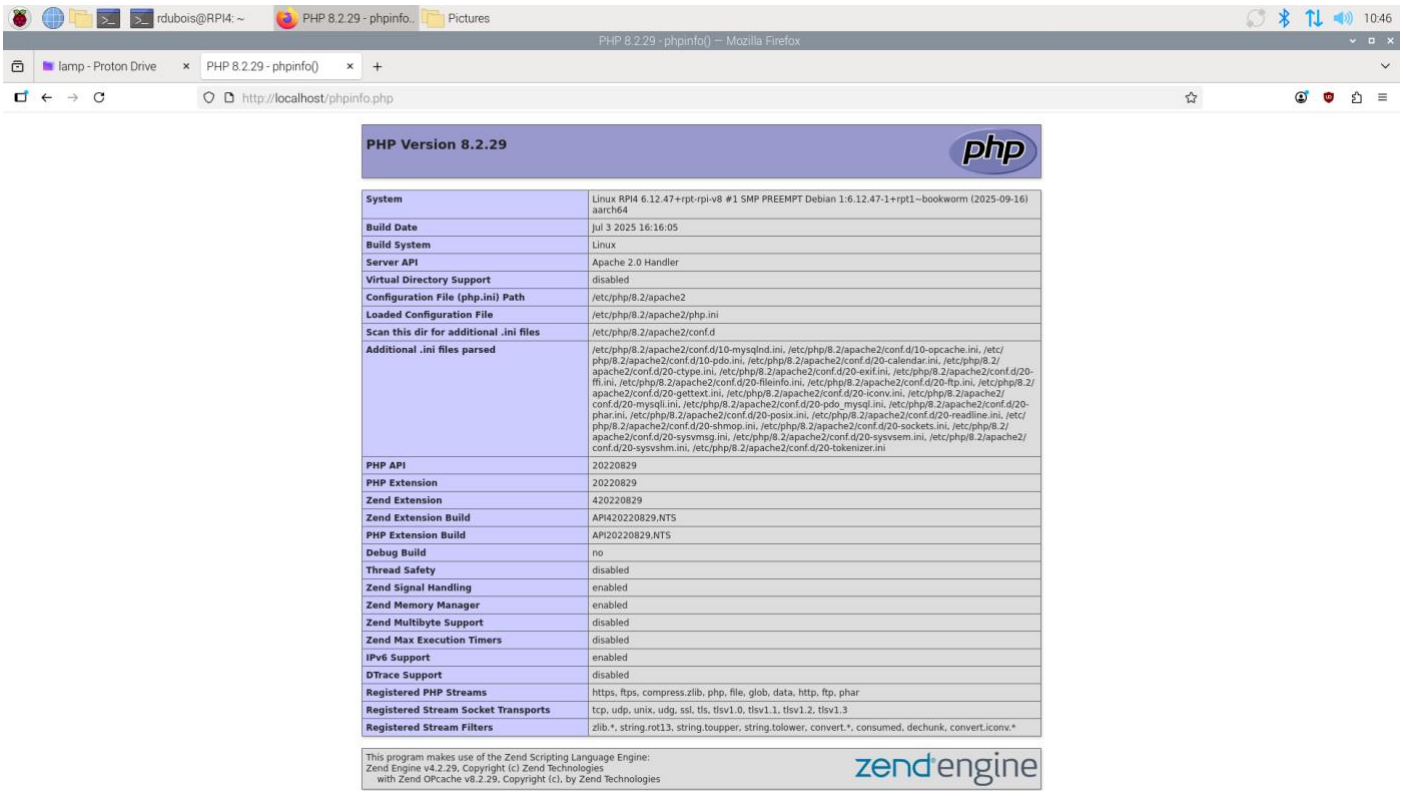


Pour tester le serveur MySQL : sudo mysql -u root -p et exit pour sortir



Pour tester le PHP, on créer un fichier /var/www/html/phpinfo.php



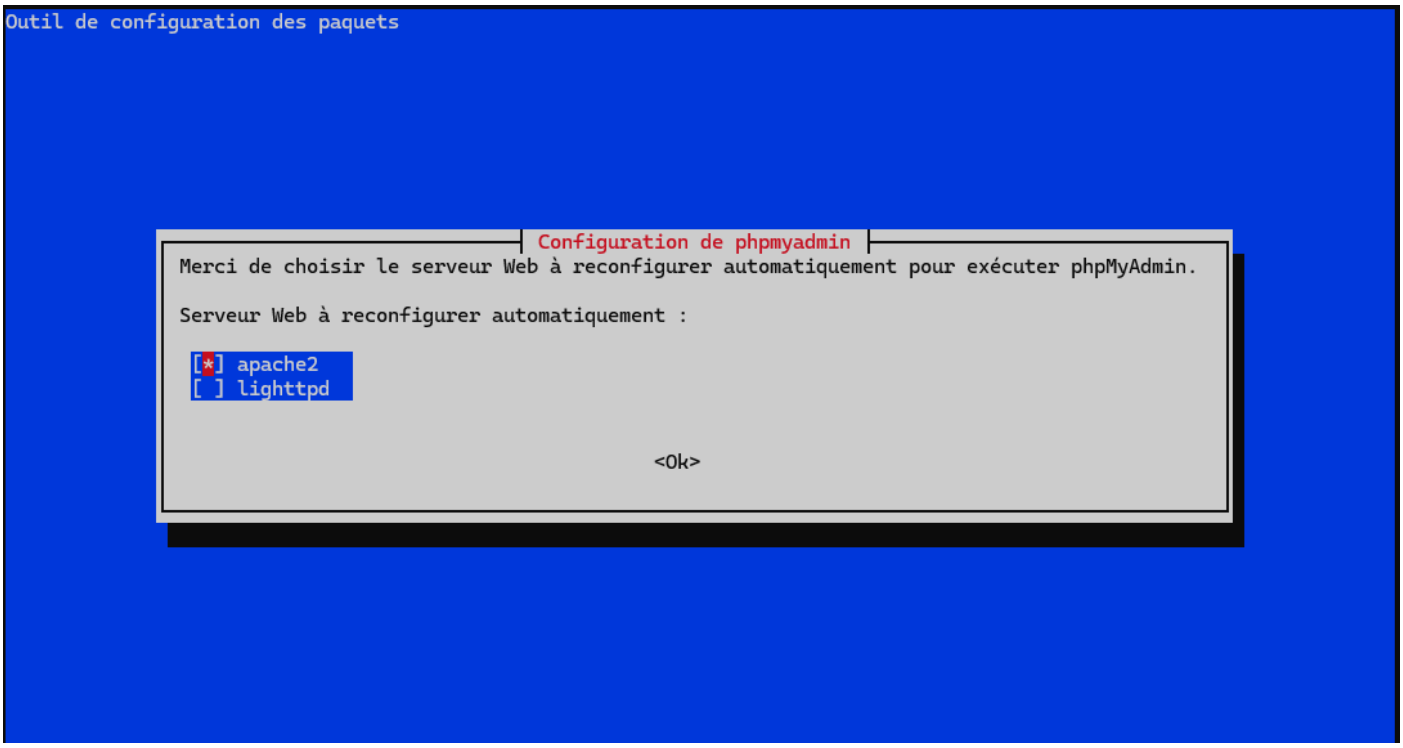


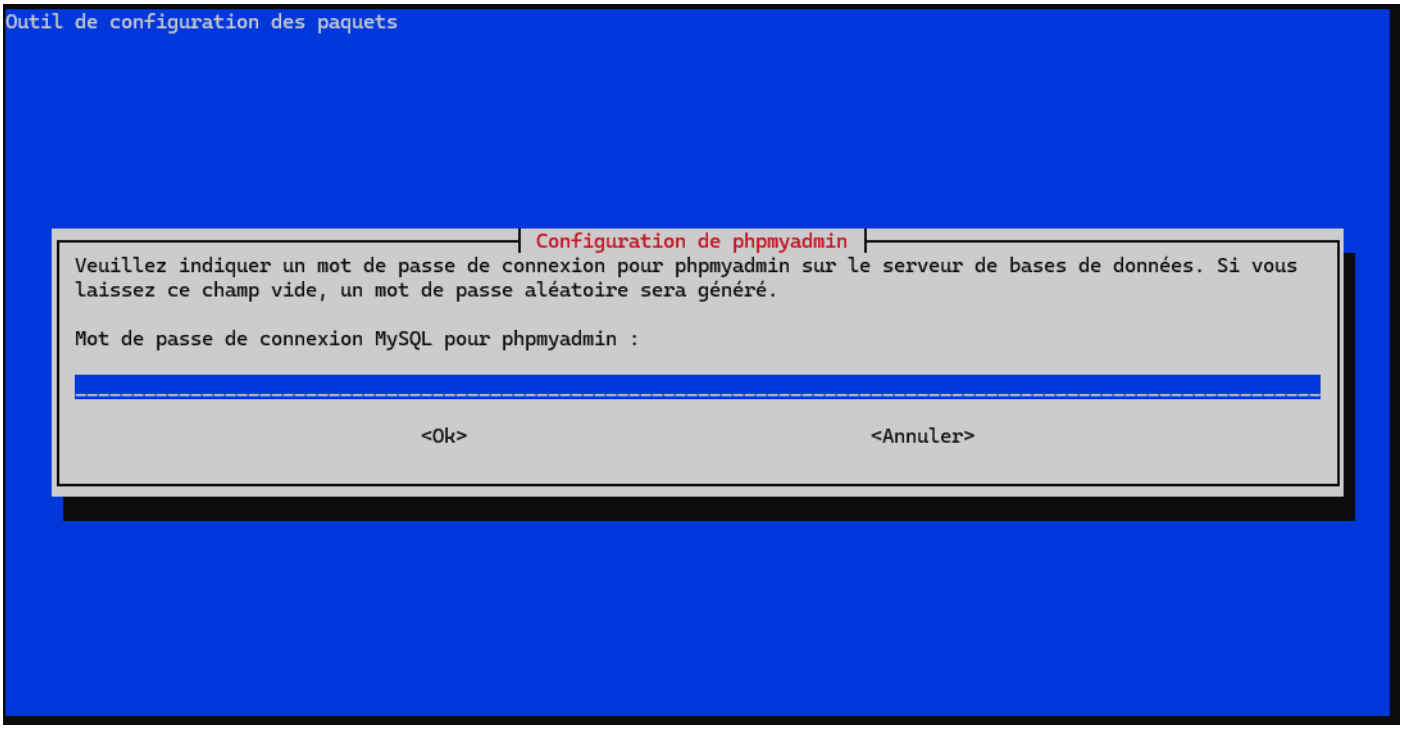
III/ Utilisation

Installer phpmyadmin pour gérer les bases de données graphiquement.

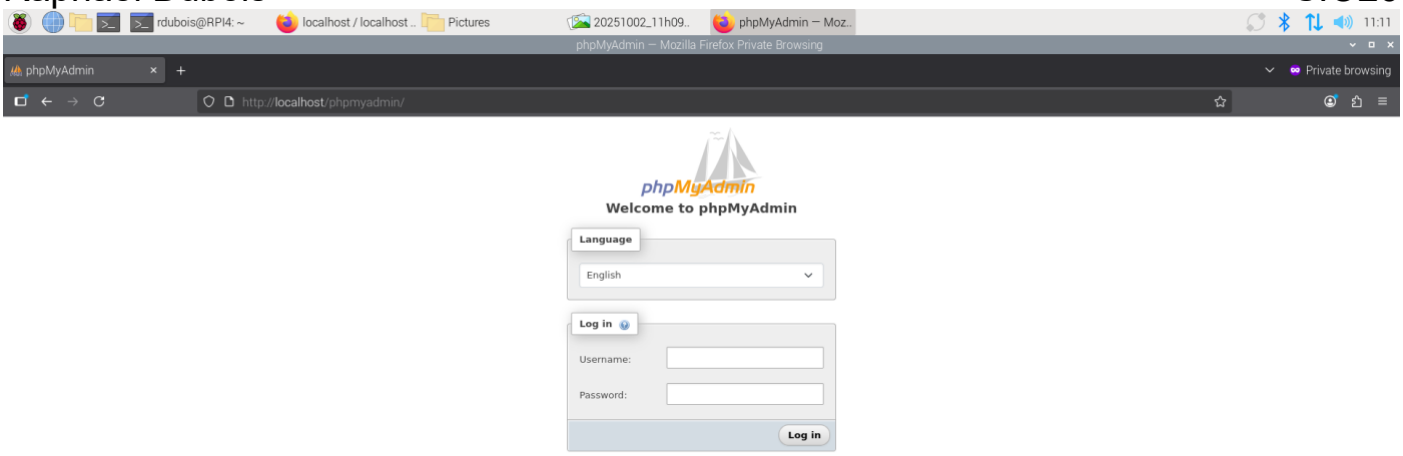
Commande pour l'installer : **sudo apt install phpmyadmin**

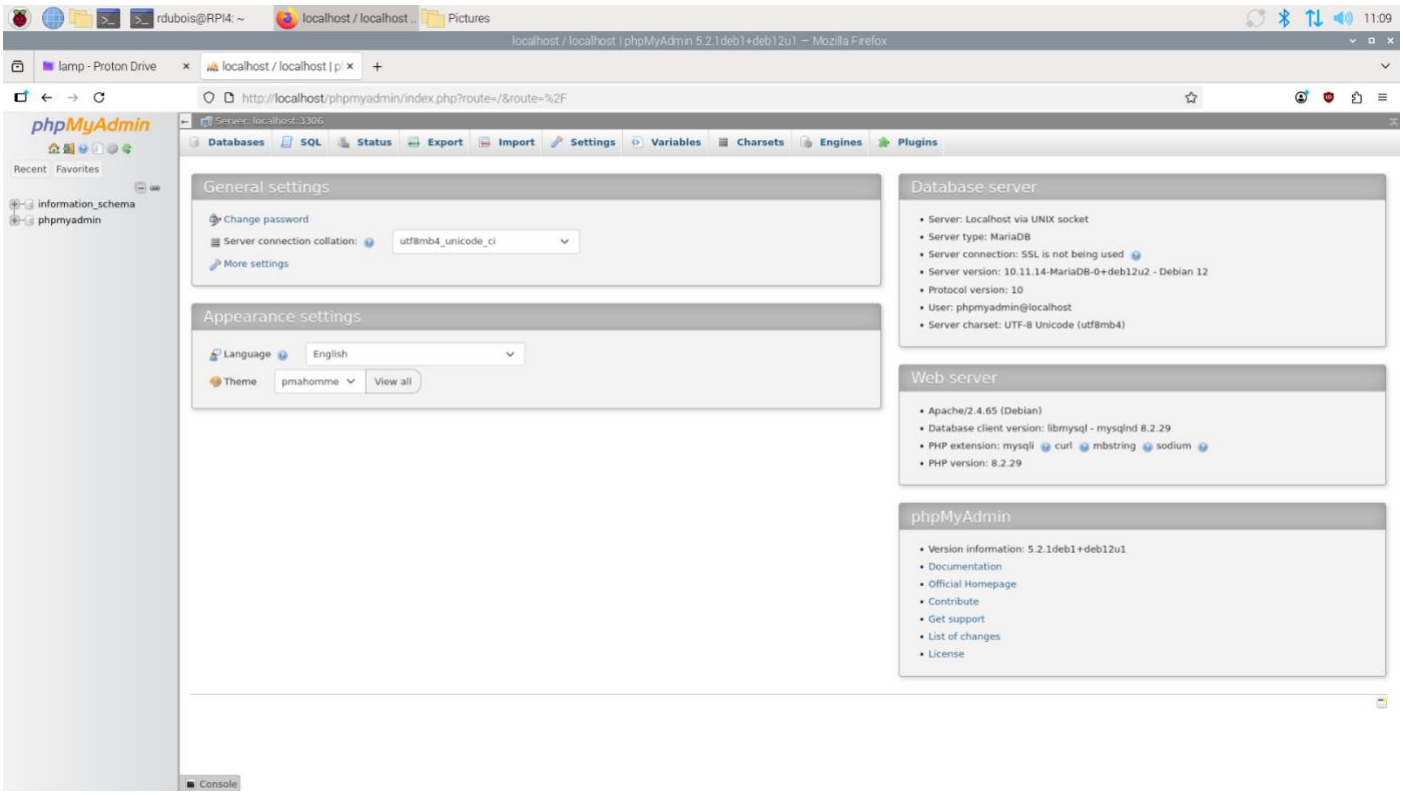
Pendant l'installation, il sera demandé de sélectionner le serveur web actuellement installé (en l'occurrence Apache2), définir un mot de passe pour l'utilisateur « phpmyadmin » :





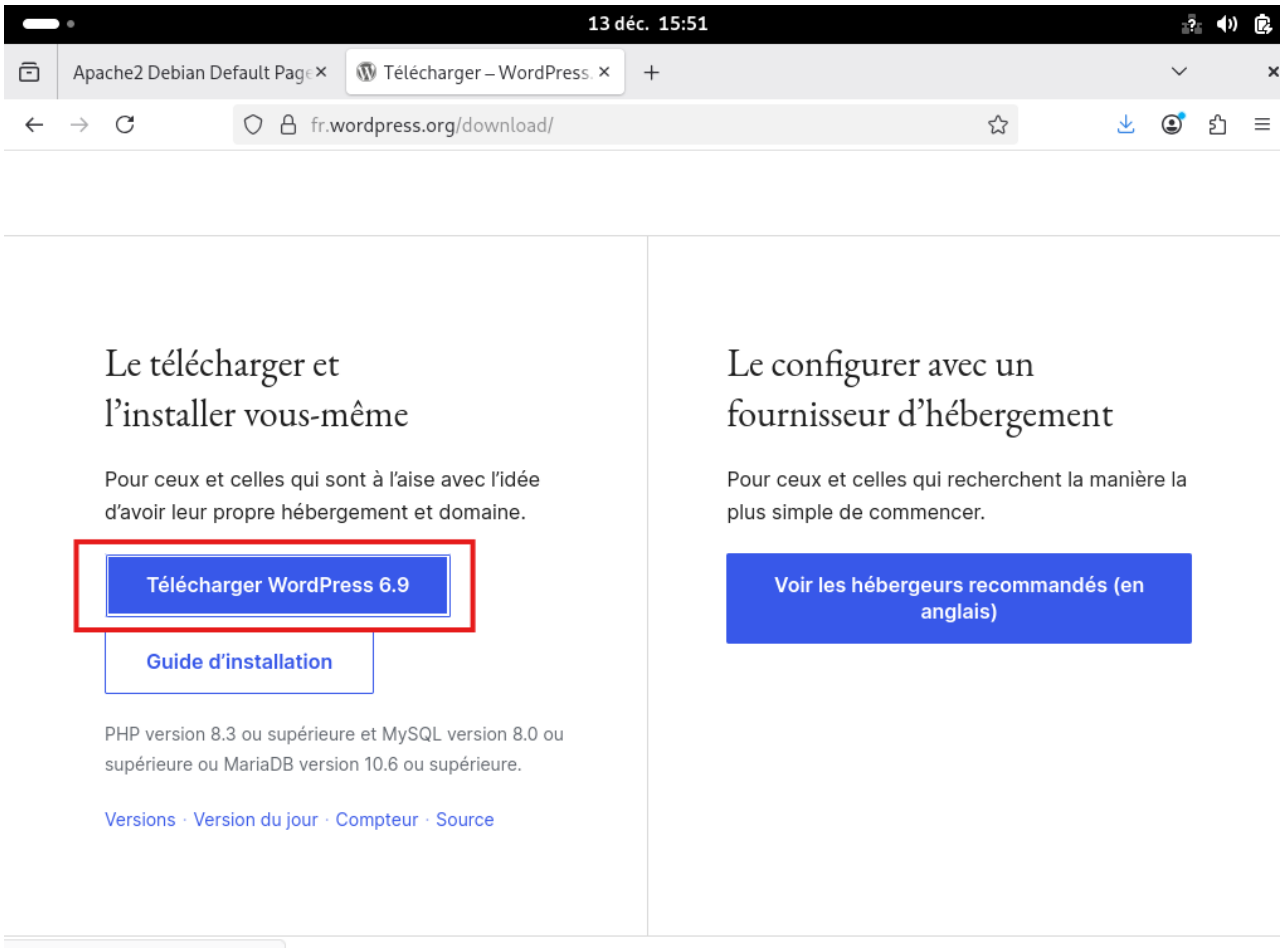
Une fois ceci défini, on peut se rendre sur localhost/phpmyadmin et tenter de s'y connecter



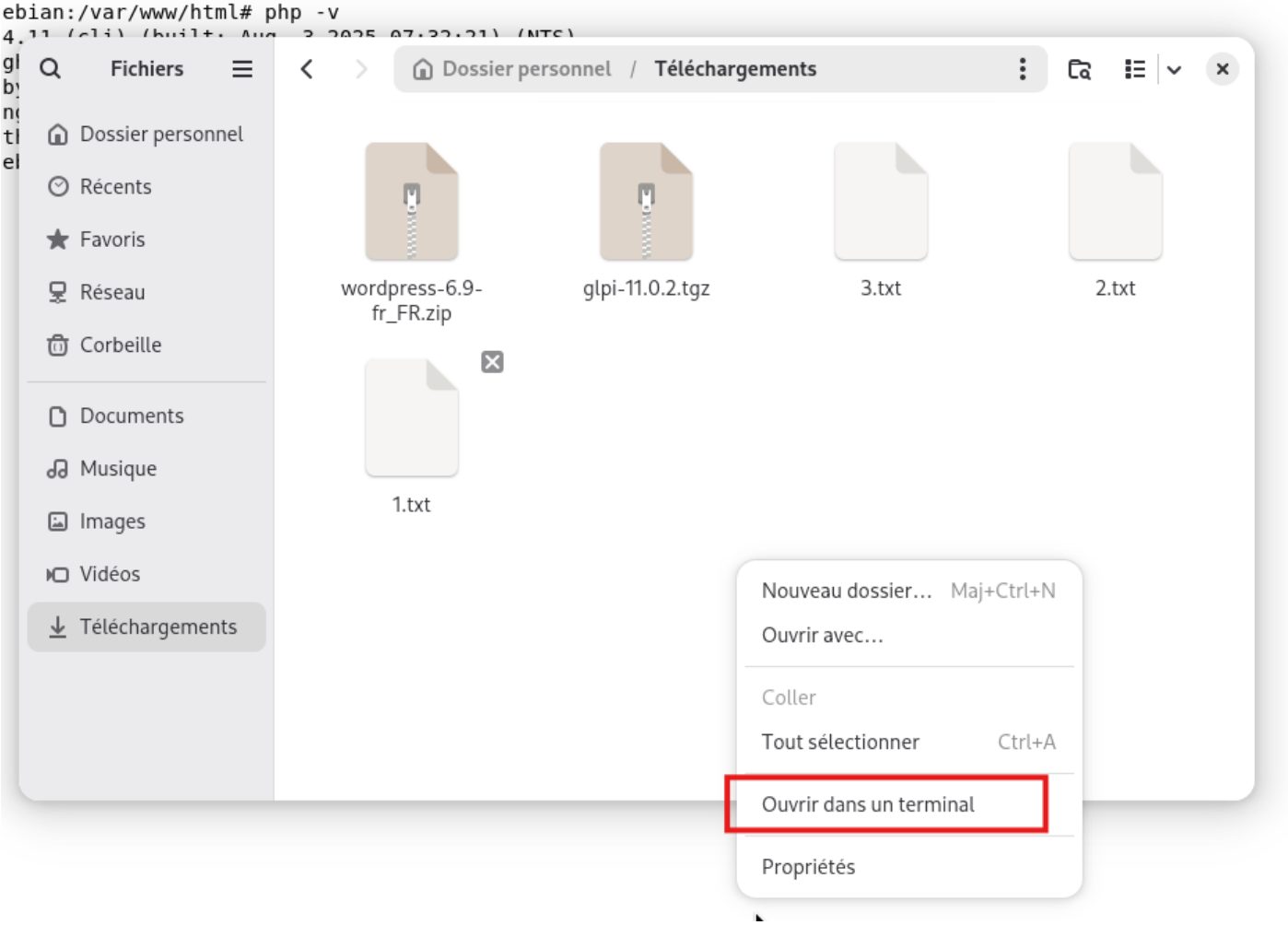


1) Installation d'un CMS (Content Management System / WordPress)

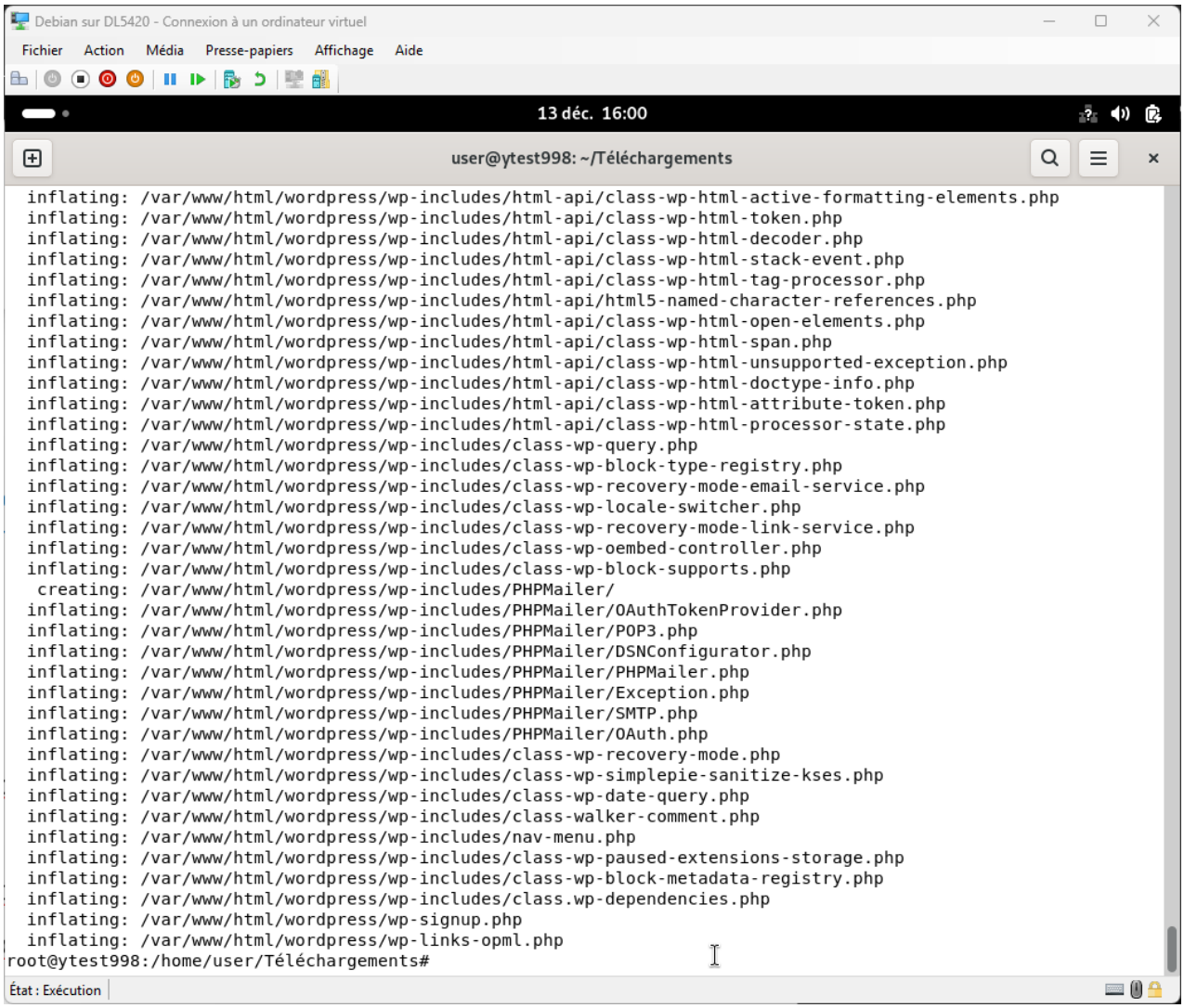
Nous allons donc télécharger et installer WordPress, pour cela, nous allons nous rendre sur la page de téléchargement WordPress (https://fr.wordpress.org/download) :



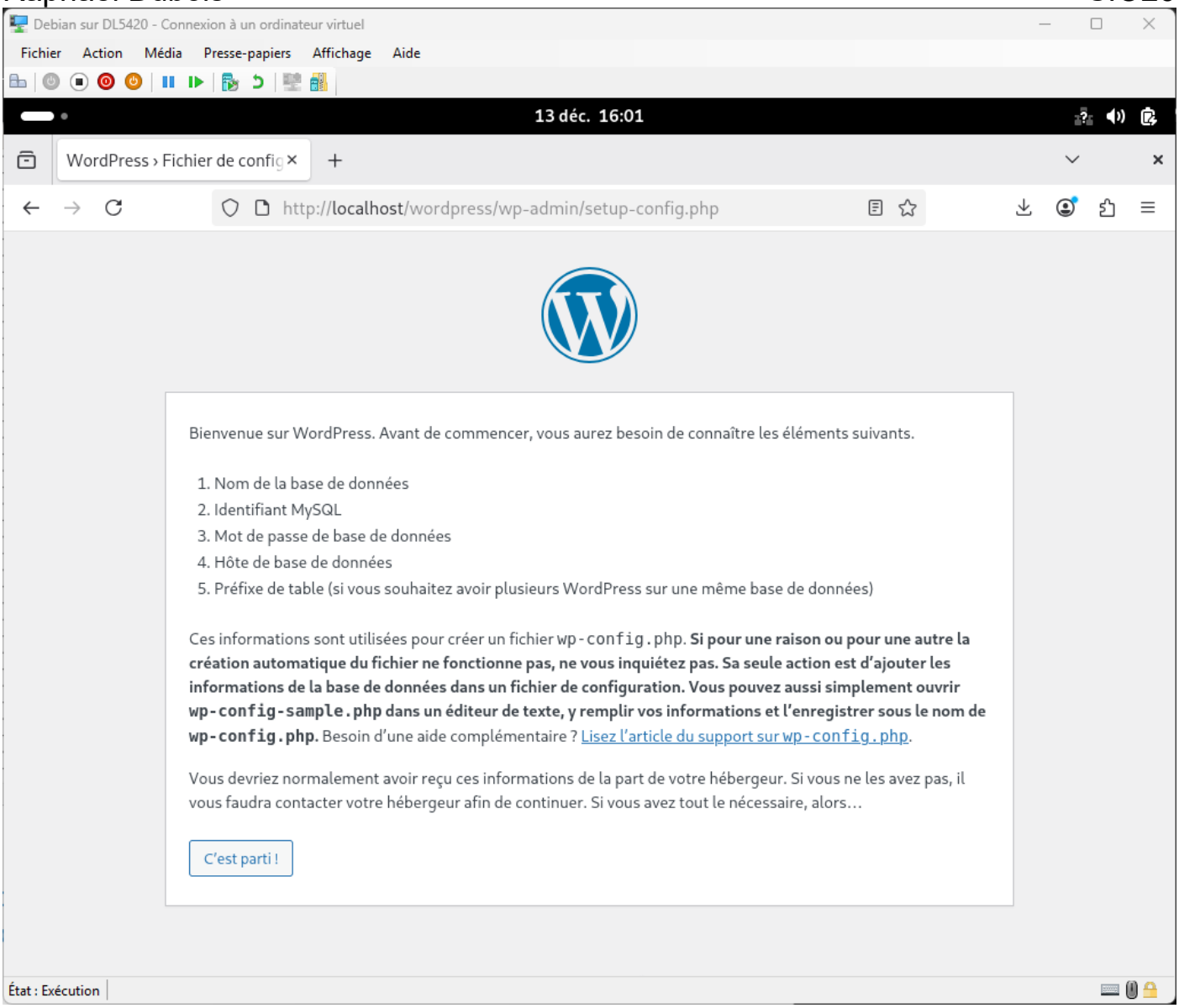
Une fois téléchargé, nous allons ouvrir un terminal dans le dossier « Téléchargement » :



Dans ce terminal, nous allons décompresser l'archive WordPress à la racine de notre site grâce à la commande suivante : **unzip wordpress-6.9-fr_FR.zip -d /var/www/html**



Une fois ceci fait, nous pouvons accéder à la page d'installation via localhost/wordpress sur notre navigateur :



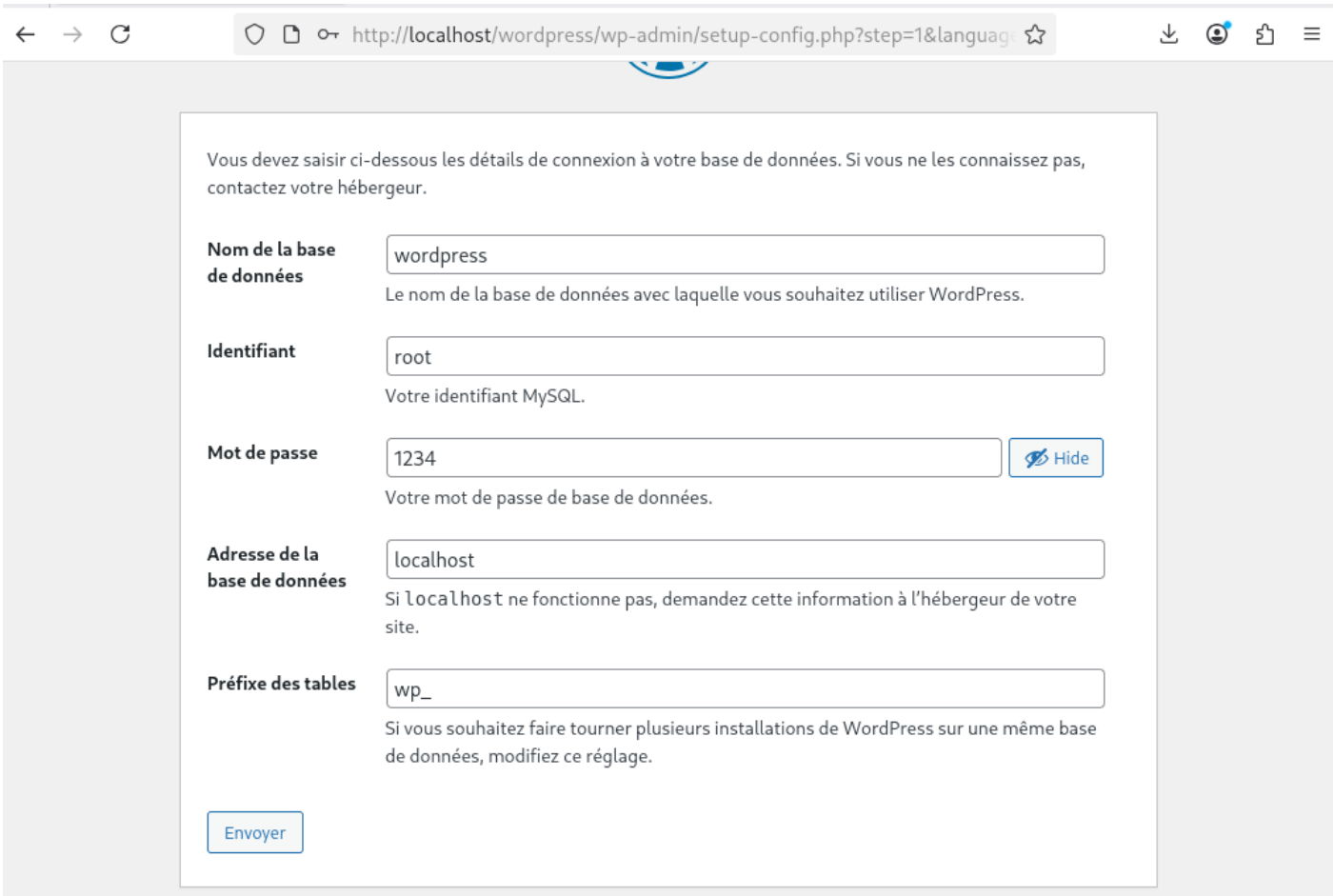
Avant de continuer ici, nous allons créer une base de données WordPress (ou sera stocké toutes les données de notre WordPress).

Pour cela, nous allons entrer les commandes suivantes dans notre terminal :

```
1  sudo mysql -u root -p
2  ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';
3  CREATE DATABASE wordpress;
4  exit
```

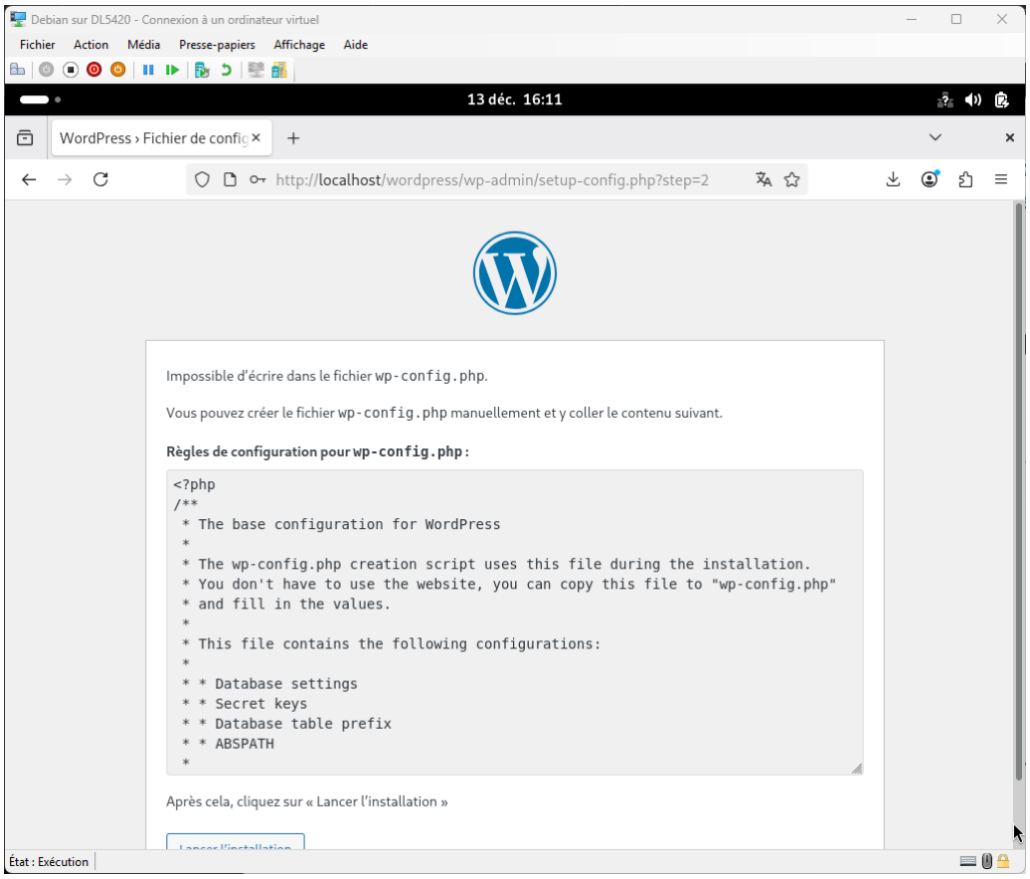
```
user@debian: ~
root@debian:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 35
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

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)]> ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0,005 sec)

MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> exit
Bye
root@debian:/var/www/html#
```

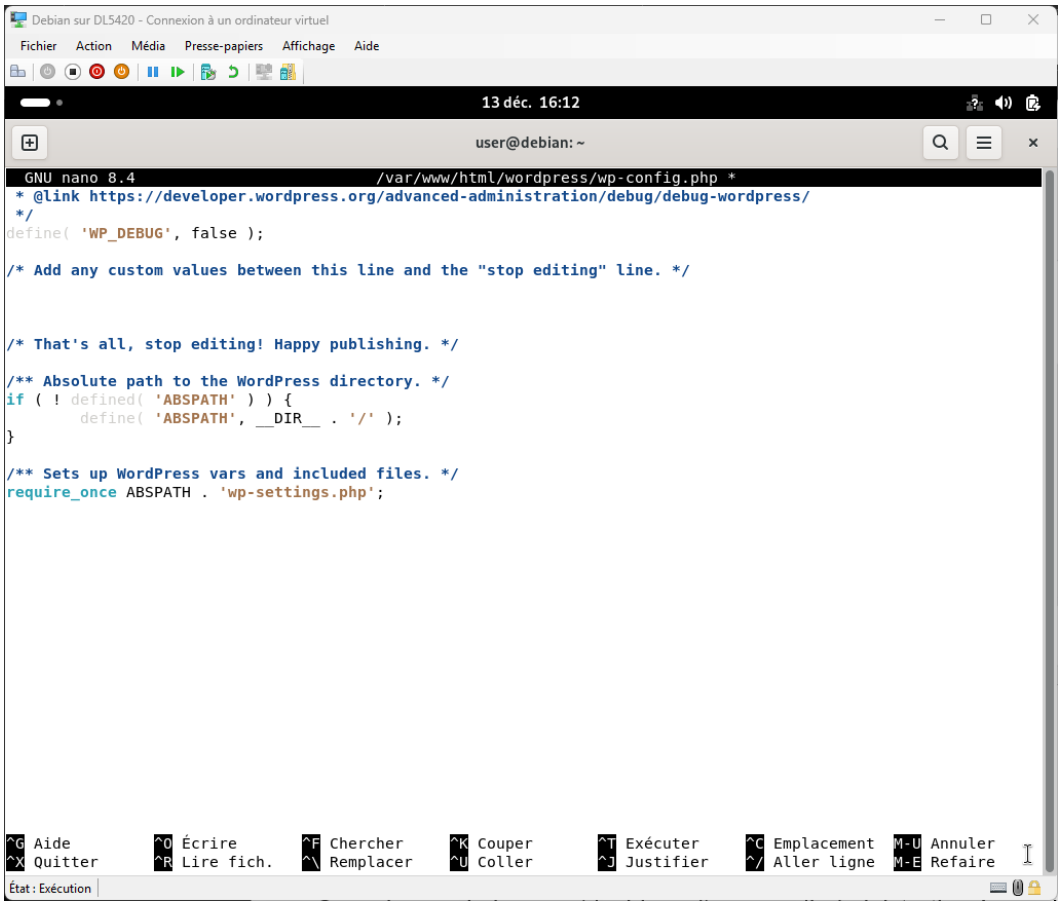
Maintenant, une fois ceci fait, nous pouvons continuer l'installation en y indiquant le nom de notre base de données, identifiants root… :



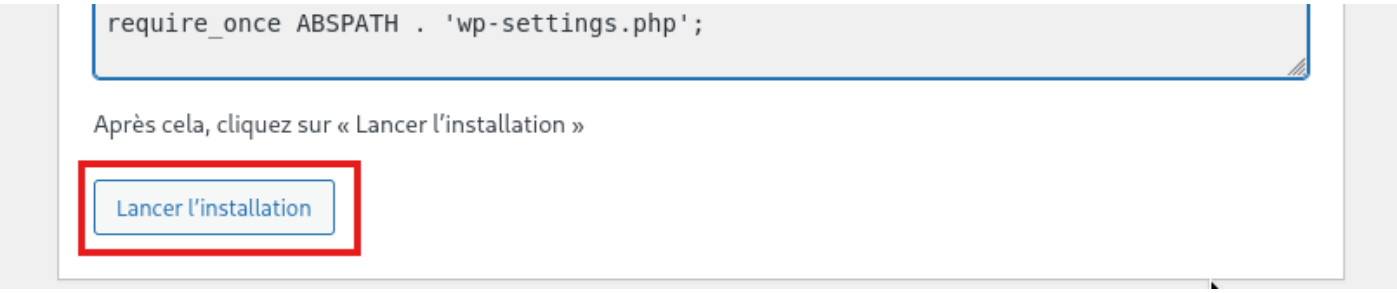
Une fois que nous avons envoyé les informations, WordPress va nous demander de créer un fichier « wp-config.php ». Il suffit de copier-coller le contenu de ce dernier

indiqué sur le site, et de le coller dans un nouveau fichier présent dans notre dossier WordPress :

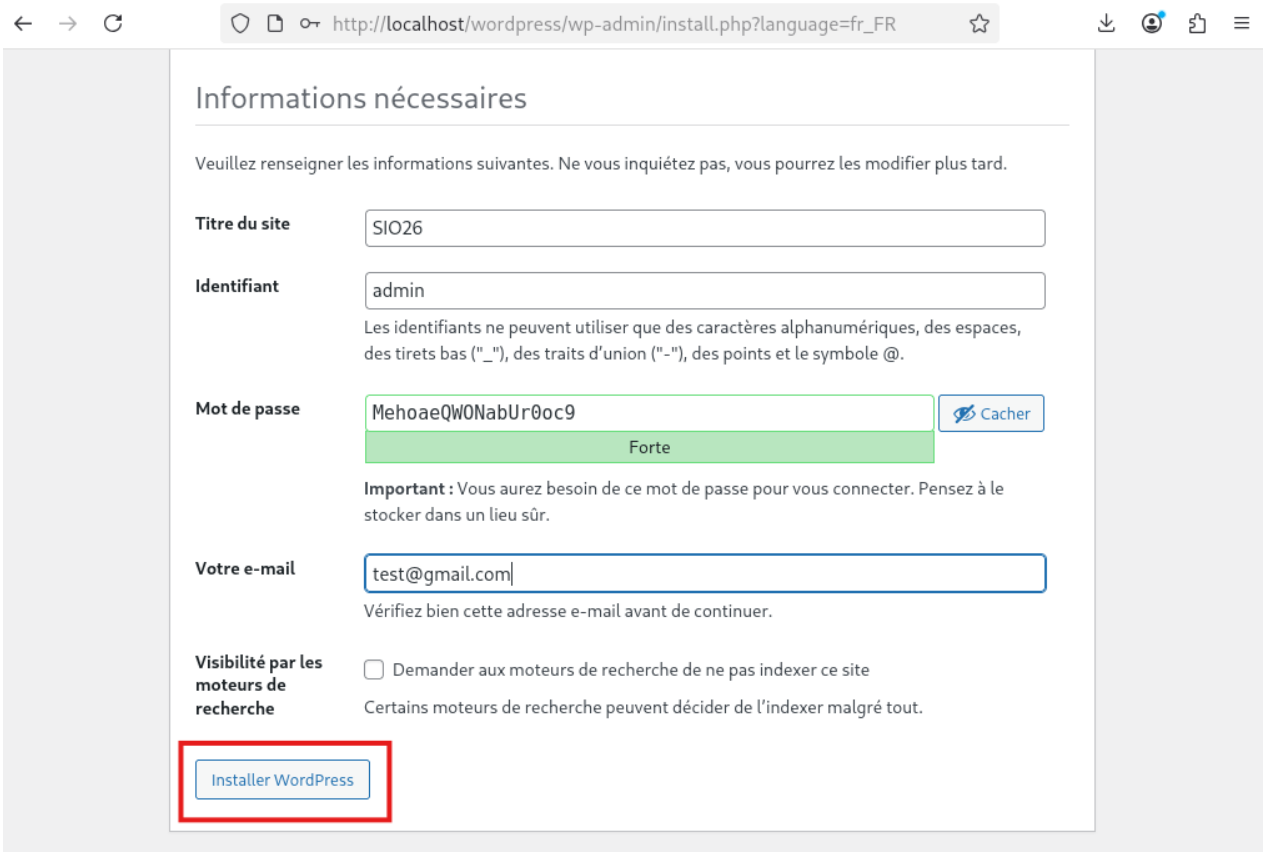




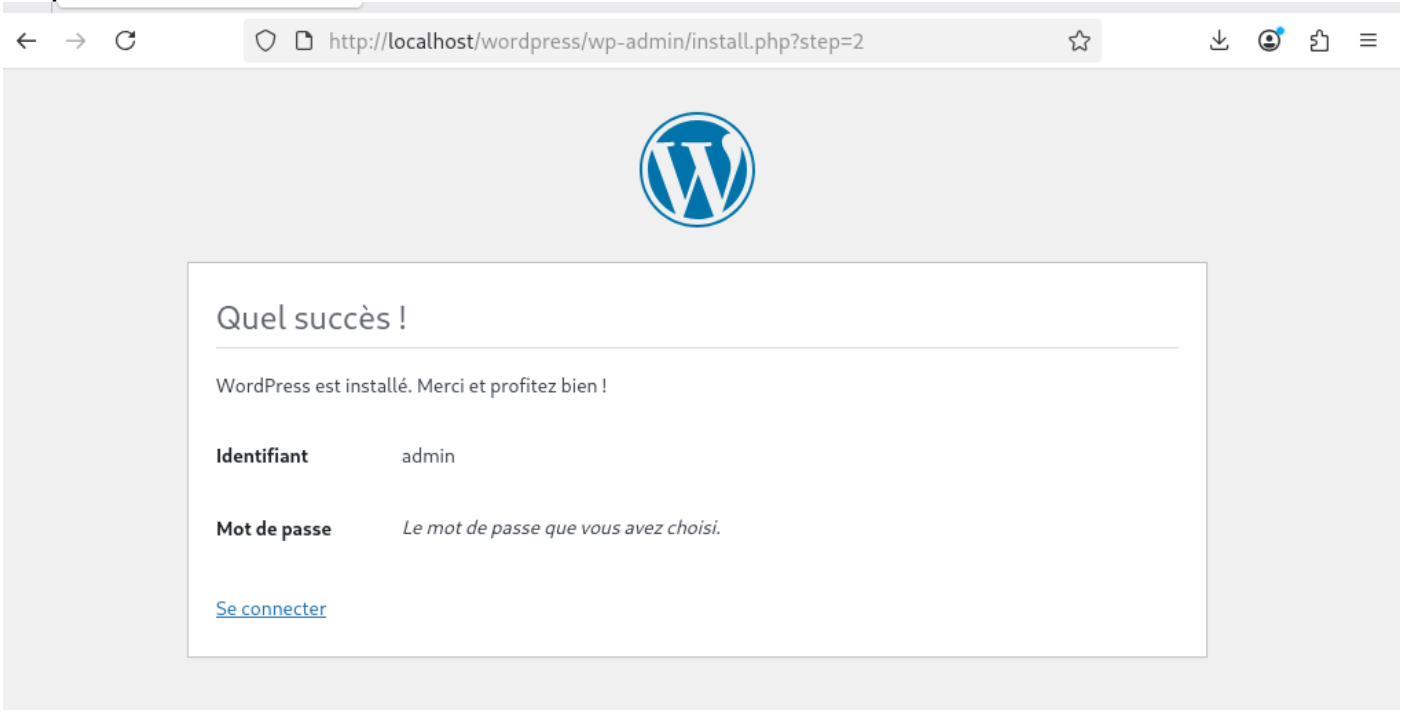
Maintenant, nous pouvons cliquer sur le bouton « Lancer l'installation » :



Maintenant, nous allons définir un titre pour notre site, ainsi qu'un identifiant de connexion pour administrer ce dernier. Une fois toutes les informations indiquées, nous pouvons cliquer sur « Installer WordPress » :

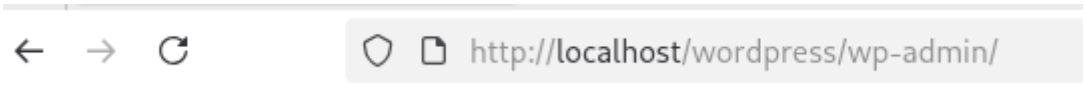


Nous pouvons dès à présent commencer à administrer notre site !

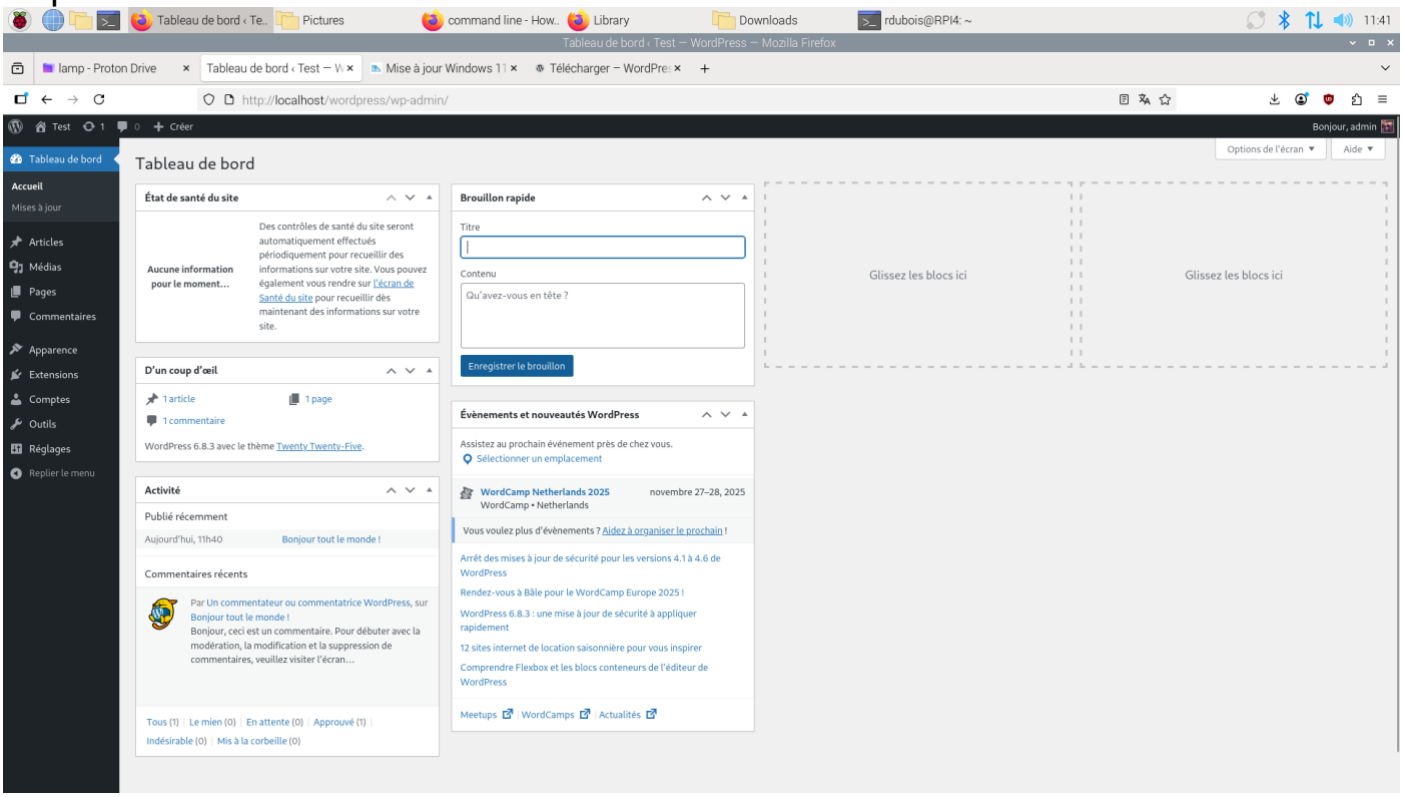


2) Prise en main de l'administration WordPress

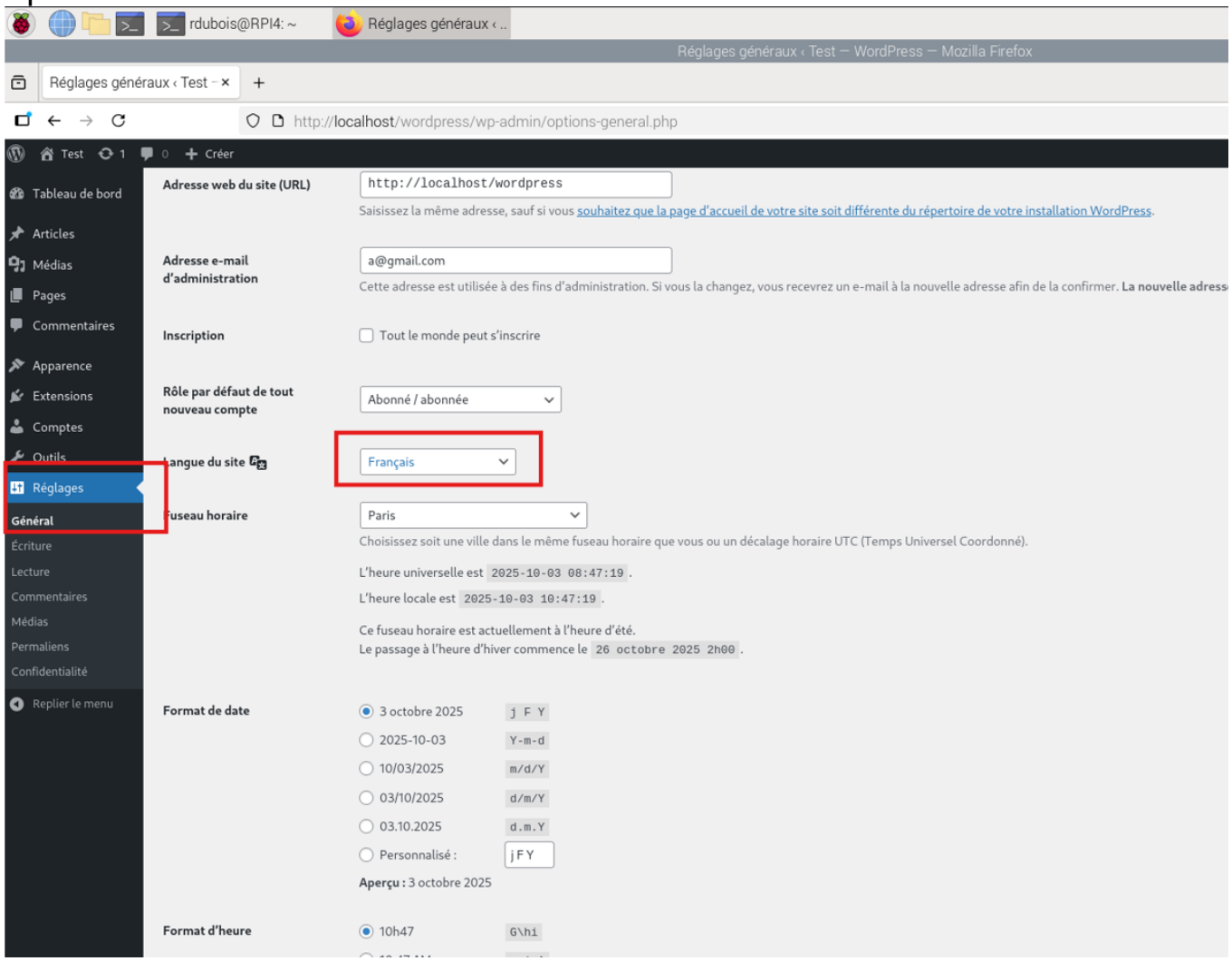
Rajouter wp-admin dans l'URL.



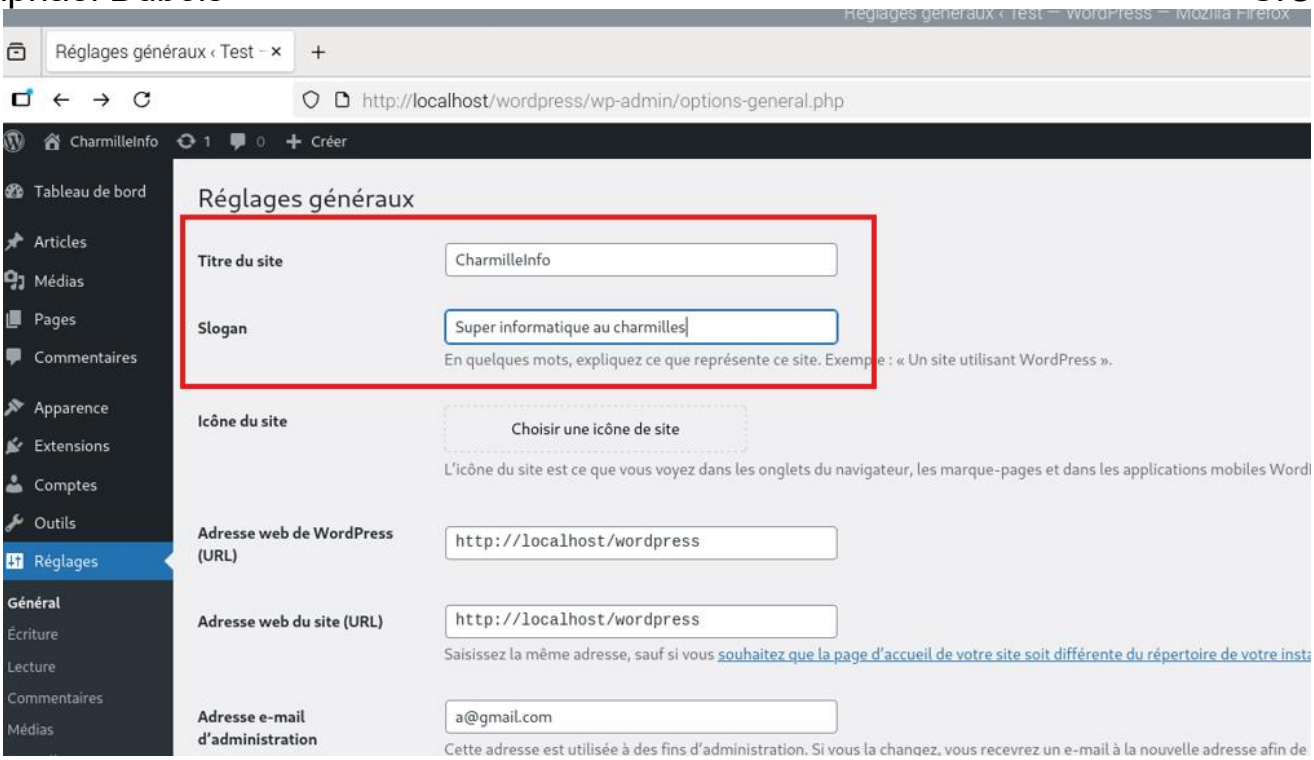
On arrive sur le bureau (dashboard), menu d'administration à gauche.



Choix de la langue dans Settings(Réglages)/Générale



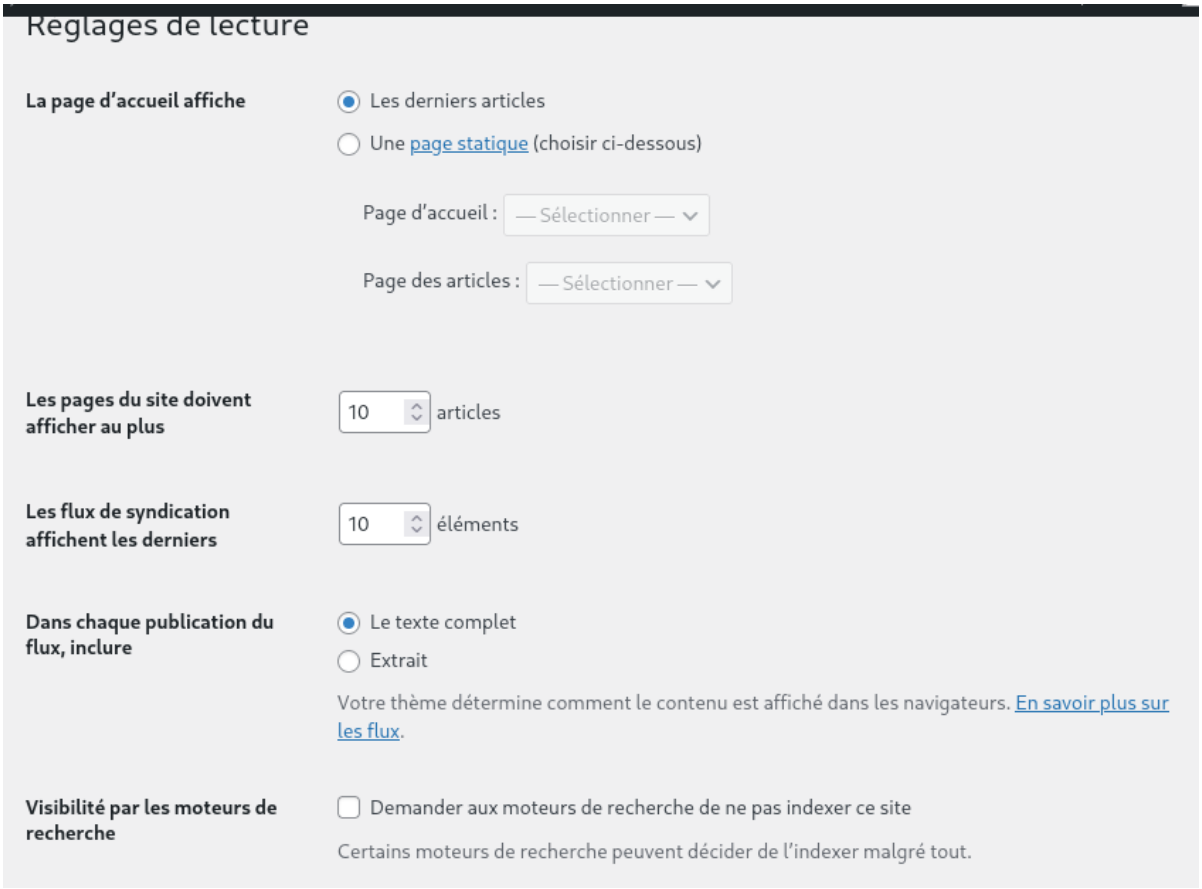
Entrer le titre du site et un slogan (mots clés pour le référencement naturel)



Parcourir les options de lecture et de commentaires.

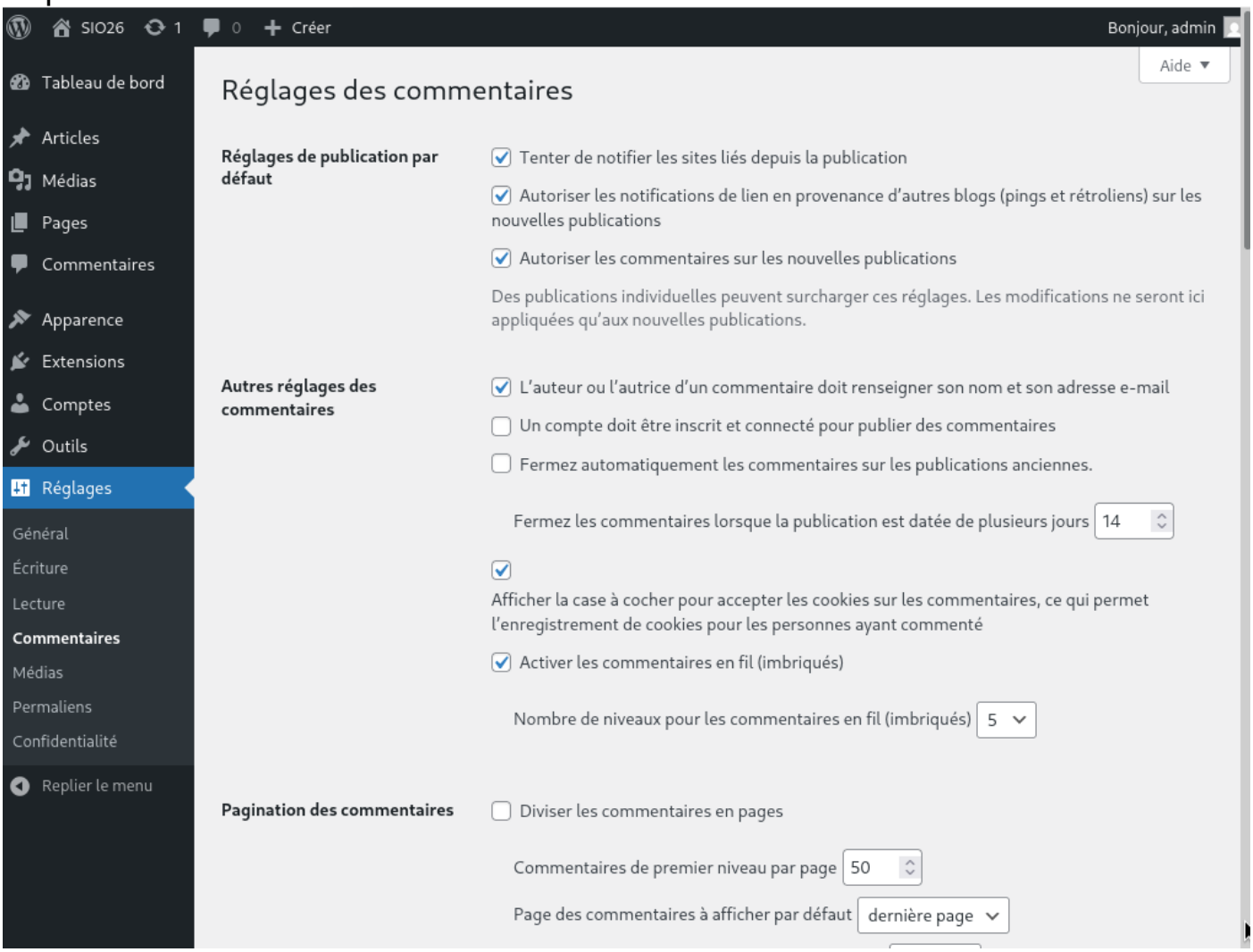
1. Onglet Lecture C'est ici que nous allons définir ce que les visiteurs voient en arrivant :

- **Page d'accueil :** Choisissez d'afficher soit vos **derniers articles** (format blog), soit une **page statique** (format site vitrine/pro).
- **Visibilité :** Il est possible de choisir si oui ou non, notre site doit s'afficher sur Google et les différents moteurs de recherches.



2. Onglet Discussion C'est ici que vous gardez le contrôle sur les échanges :

- **Autorisations :** Décidez si les visiteurs peuvent laisser des commentaires et s'ils doivent s'inscrire pour le faire.
- **Modération :** Choisissez si les commentaires s'affichent directement ou si vous devez les **approuver manuellement** (recommandé pour éviter le spam).
- **Avatar :** Il est aussi possible de sélectionner un avatar par défaut pour les utilisateurs postant des commentaires sur notre site.


SIO26
1
0
Créer
Bonjour, admin
Aide
Tableau de bord
Articles
Médias
Pages
Commentaires
Apparence
Extensions
Comptes
Outils
Réglages
Général
Écriture
Lecture
Commentaires
Médias
Permaliens
Confidentialité
Replier le menu
Réglages des commentaires
Réglages de publication par défaut
Tenter de notifier les sites liés depuis la publication
Autoriser les notifications de lien en provenance d'autres blogs (pings et rétroliens) sur les nouvelles publications
Autoriser les commentaires sur les nouvelles publications
Des publications individuelles peuvent surcharger ces réglages. Les modifications ne seront ici appliquées qu'aux nouvelles publications.
Autres réglages des commentaires
L'auteur ou l'autrice d'un commentaire doit renseigner son nom et son adresse e-mail
Un compte doit être inscrit et connecté pour publier des commentaires
Fermez automatiquement les commentaires sur les publications anciennes.
Fermez les commentaires lorsque la publication est datée de plusieurs jours 14
Afficher la case à cocher pour accepter les cookies sur les commentaires, ce qui permet l'enregistrement de cookies pour les personnes ayant commenté
Activer les commentaires en fil (imbriqués)
Nombre de niveaux pour les commentaires en fil (imbriqués) 5
Pagination des commentaires
Diviser les commentaires en pages
Commentaires de premier niveau par page 50
Page des commentaires à afficher par défaut dernière page

M'envoyer un message lorsque

- [x] Un nouveau commentaire est publié
- [x] Un commentaire est en attente de modération
- [x] N'importe qui peut publier une note

Avant la publication d'un commentaire

- [ ] Le commentaire doit être approuvé manuellement
- [x] L'auteur ou l'autrice d'un commentaire doit avoir déjà au moins un commentaire approuvé

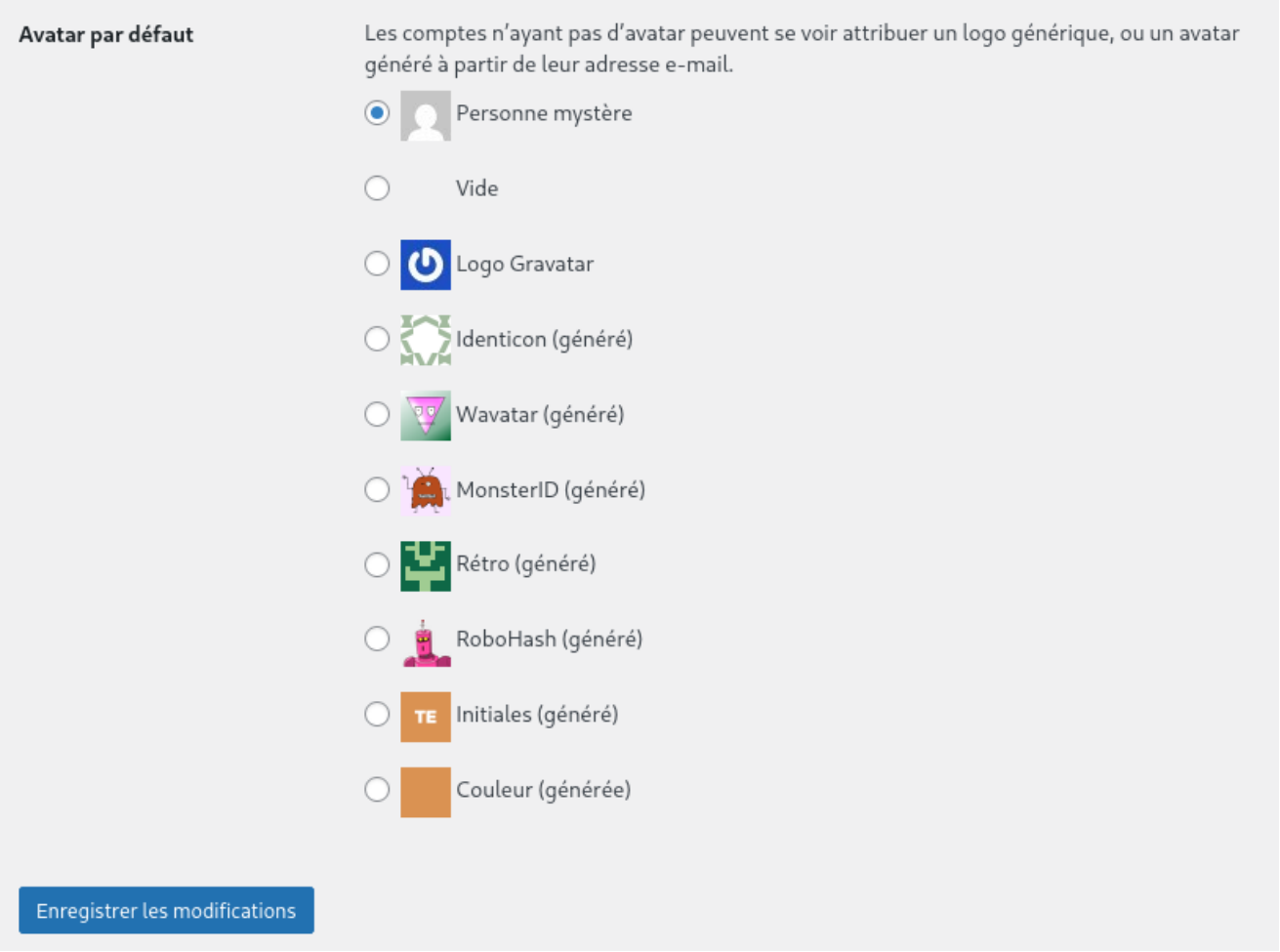
Modération de commentaires

Garder un commentaire dans la file d'attente s'il contient plus de 2 lien(s) (une des caractéristiques typiques d'un commentaire indésirable (spam) est son nombre important de liens)

Lorsqu'un commentaire contient l'un de ces mots dans son contenu, nom d'auteur ou d'autrice, URL, e-mail, adresse IP ou encore la chaîne d'agent utilisateur du navigateur, il sera retenu dans la file de modération. Un seul mot ou une seule adresse IP par ligne. Cette fonction reconnaît l'intérieur des mots, donc « press » suffira pour reconnaître « WordPress ».

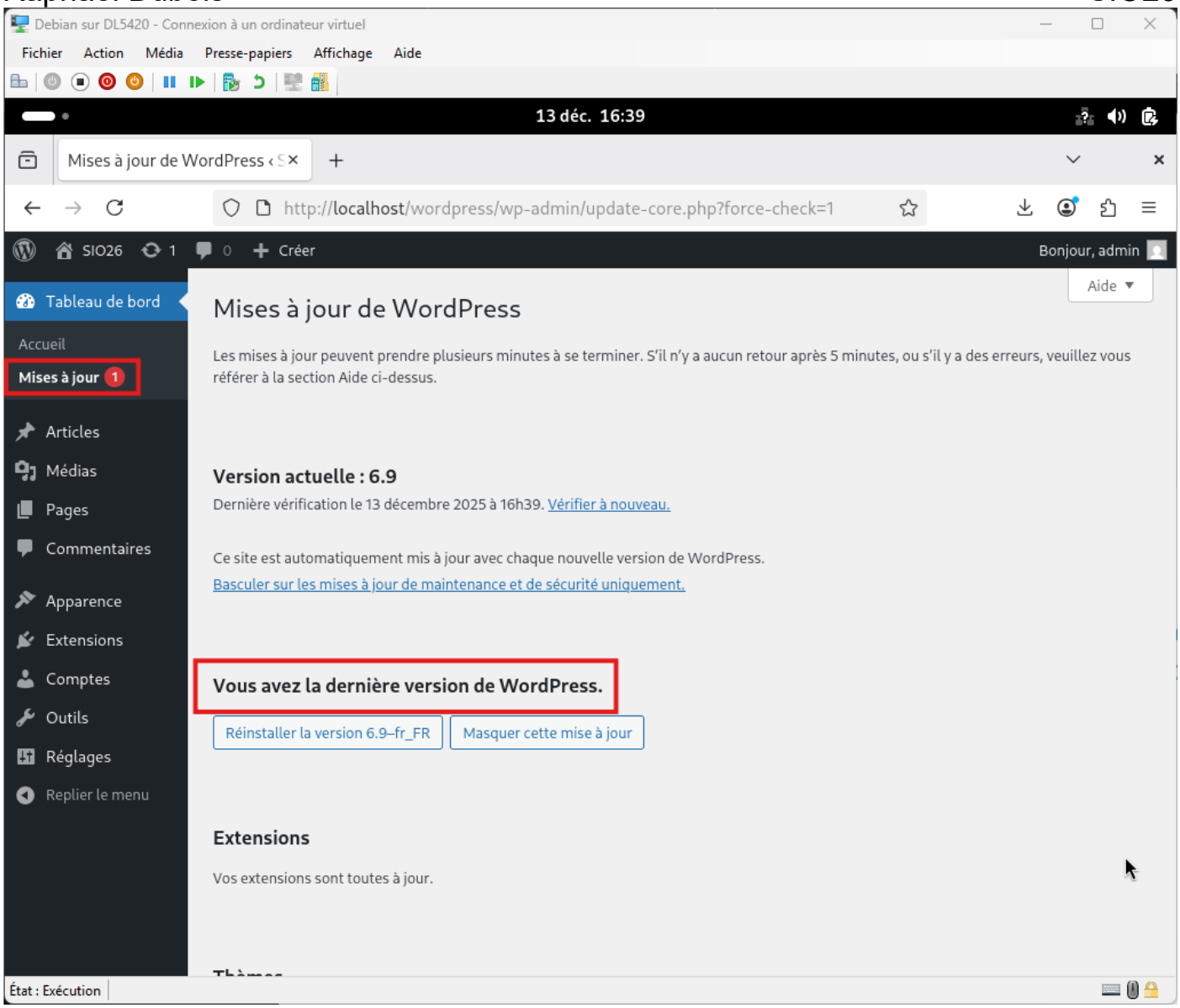
Clés de commentaires non autorisées

Lorsqu'un commentaire contient l'un de ces mots dans son contenu, son nom, son URL, son e-mail, son adresse IP ou encore la chaîne d'agent utilisateur du navigateur, il sera mis à la corbeille. Un seul mot ou une seule adresse IP par ligne. Cette fonction reconnaît l'intérieur des mots, donc « press » suffira pour reconnaître « WordPress ».



Mise à jour

Nous pouvons aussi effectuer les mises à jour de WordPress depuis l'interface d'administration. Pour cela, il suffit de se rendre sur le tableau de bord, puis dans le volet « mises à jour » :



La page nous indique bien que WordPress est bien à jour, à la dernière version.

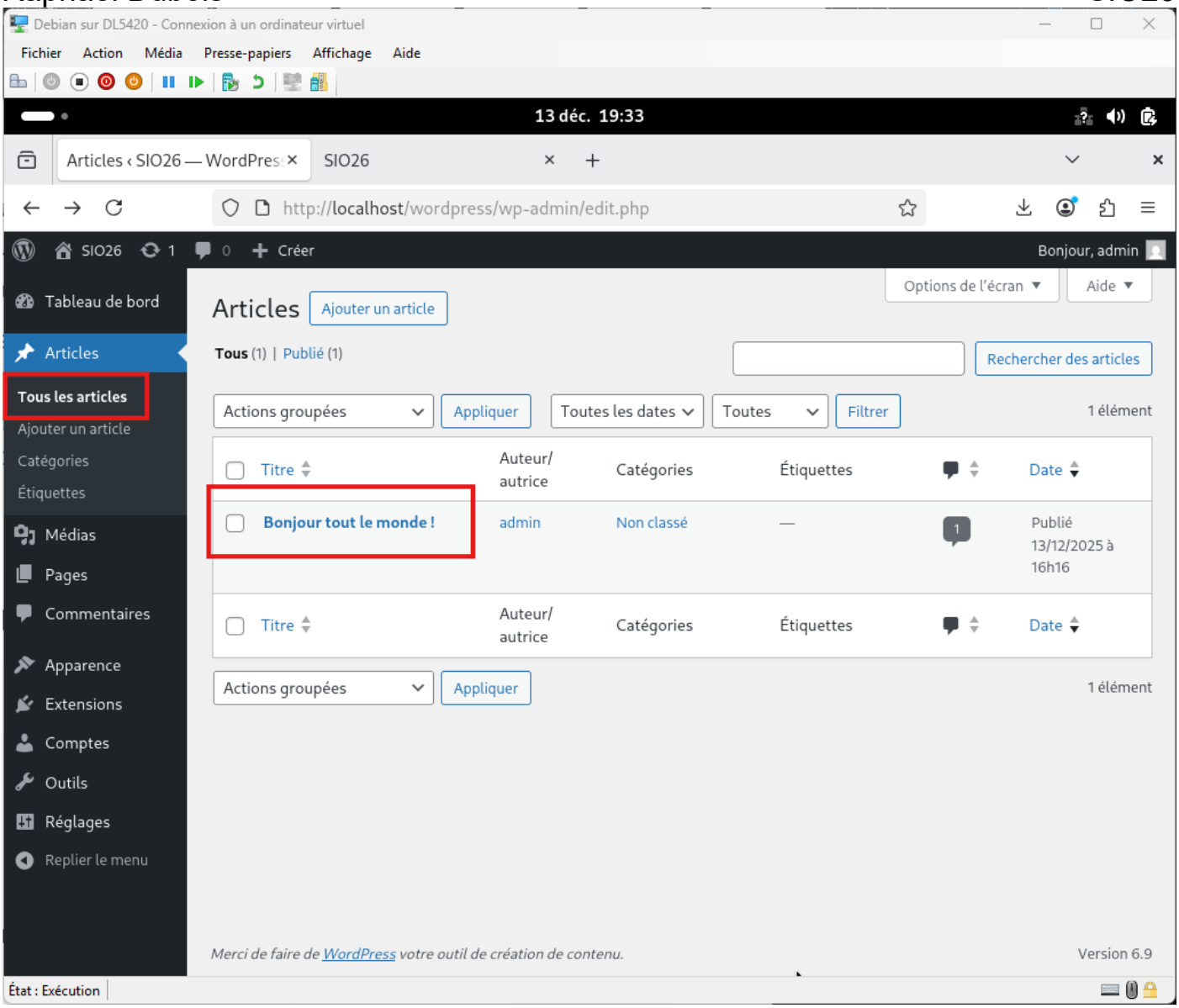
3) Créer un premier article WordPress

Deux types de contenus

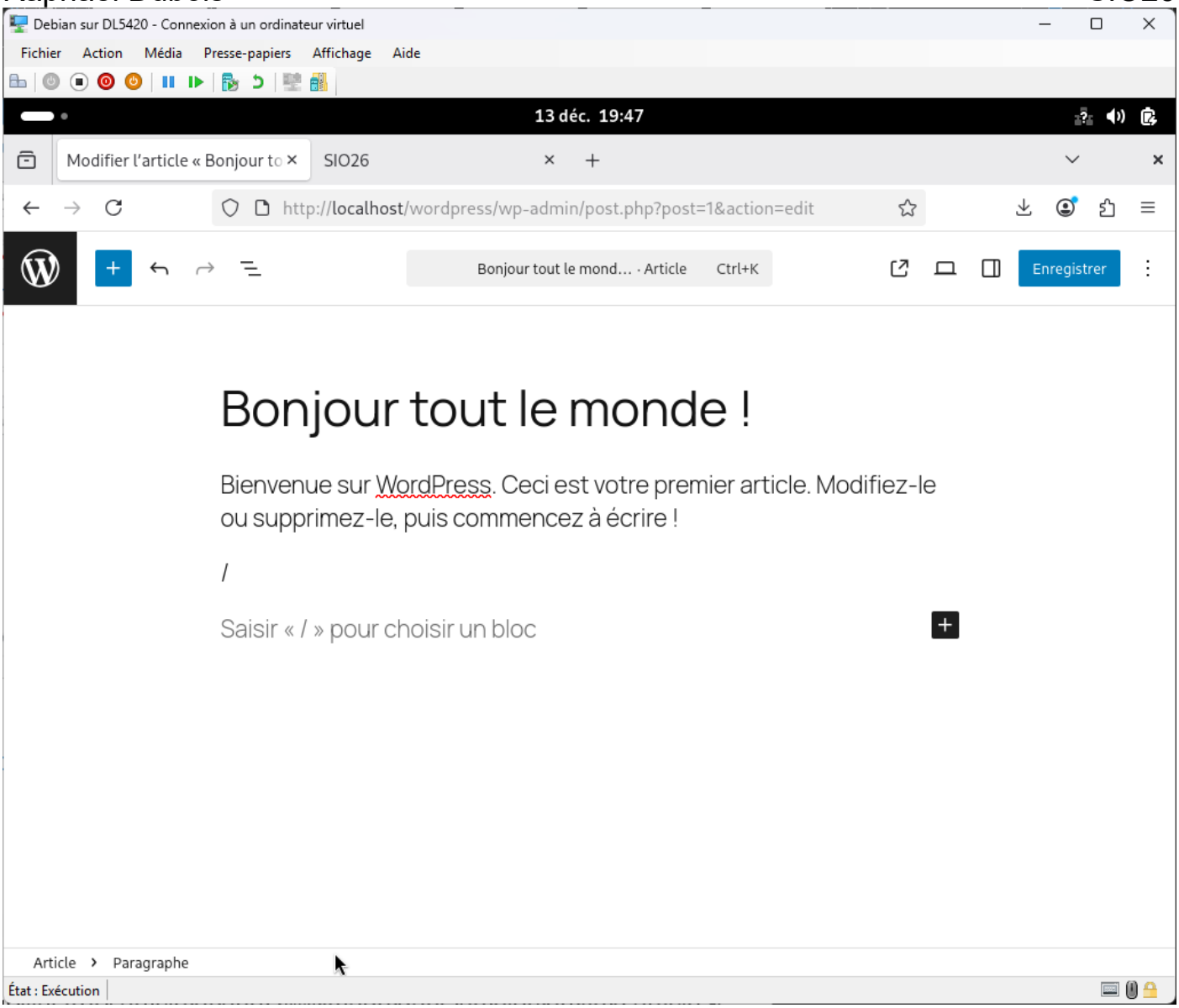
- Les articles : contenus d'actualité, dates, classés en blog.
- Les pages : contenus statiques, non datés, dans le menu ou pied de page

Créer un nouvel article

Nous allons nous rendre dans le panneau d'administration, puis dans « Articles », puis enfin « Tous les articles ». Sur cette page, nous allons cliquer sur « Bonjour tout le monde » étant l'article par défaut :

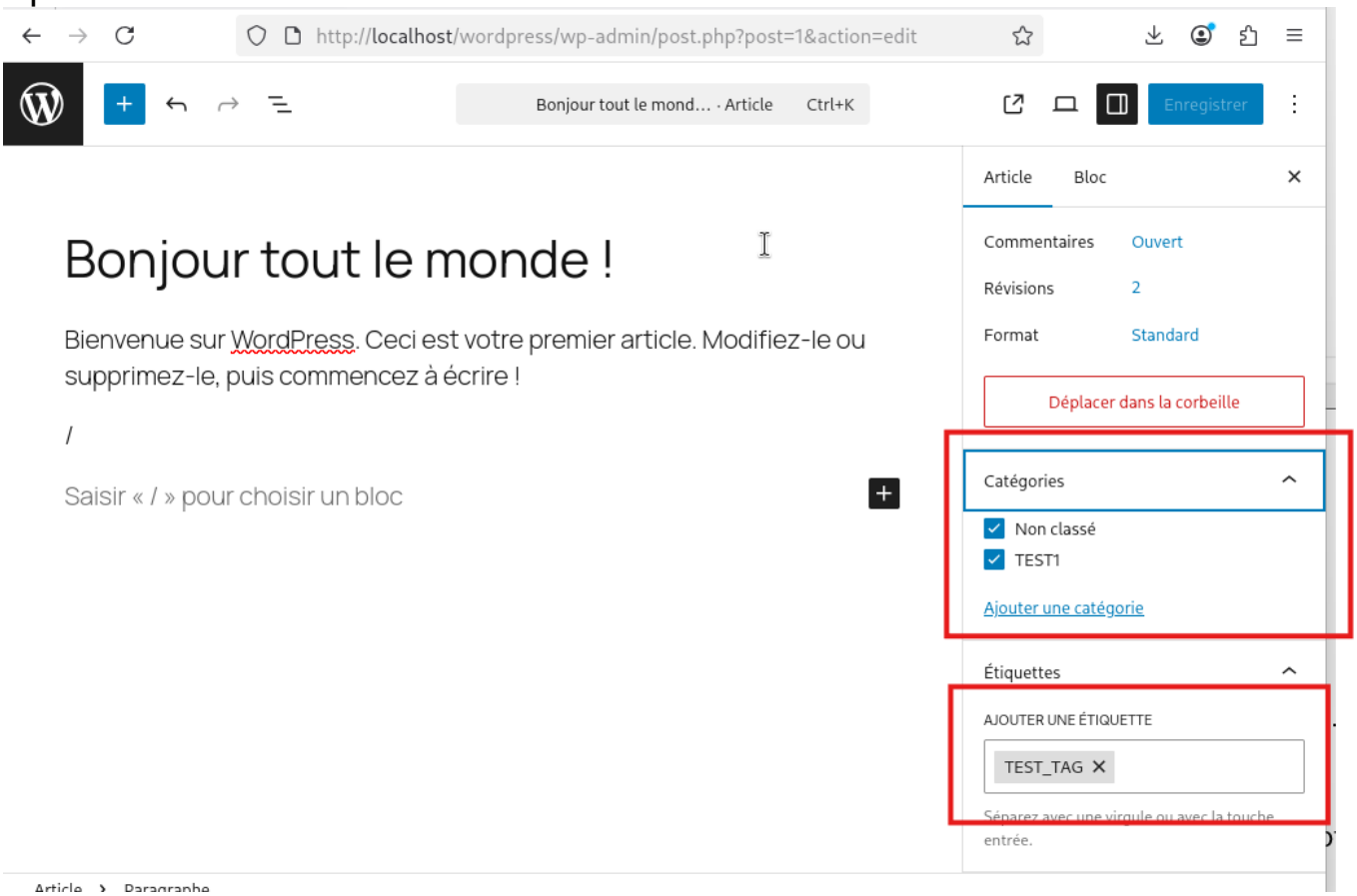


Une fois que l'on a fait ceci, nous pouvons modifier notre article :

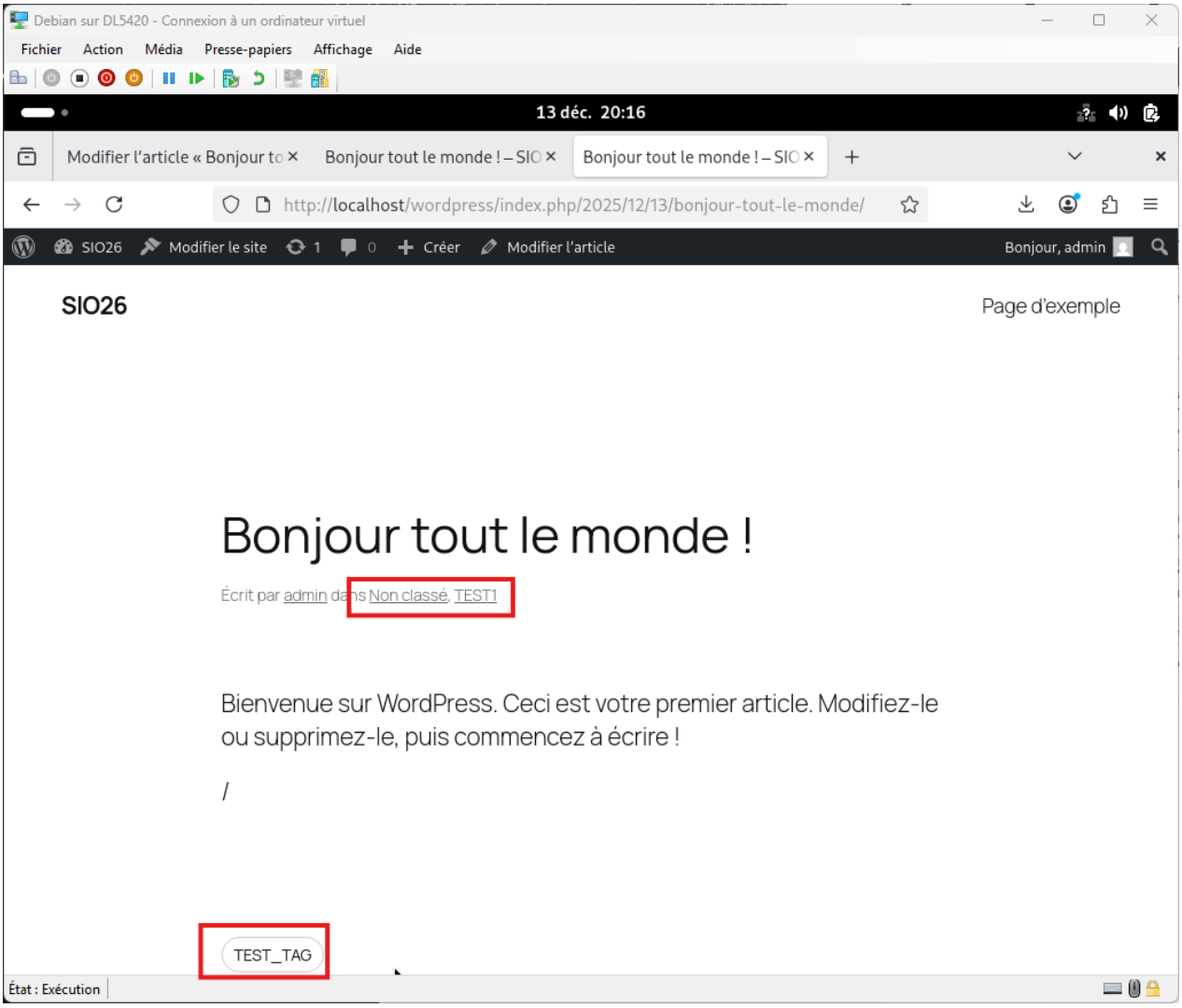


Ajouter une catégorie pour l'article, et des tags

Pour ajouter une catégorie à l'article, il suffit de se rendre dans le volet de droite, dans « Article » puis de descendre un peu jusqu'à catégorie et étiquettes. Ici, il sera possible d'ajouter une catégorie et des étiquettes avec le nom de son choix :

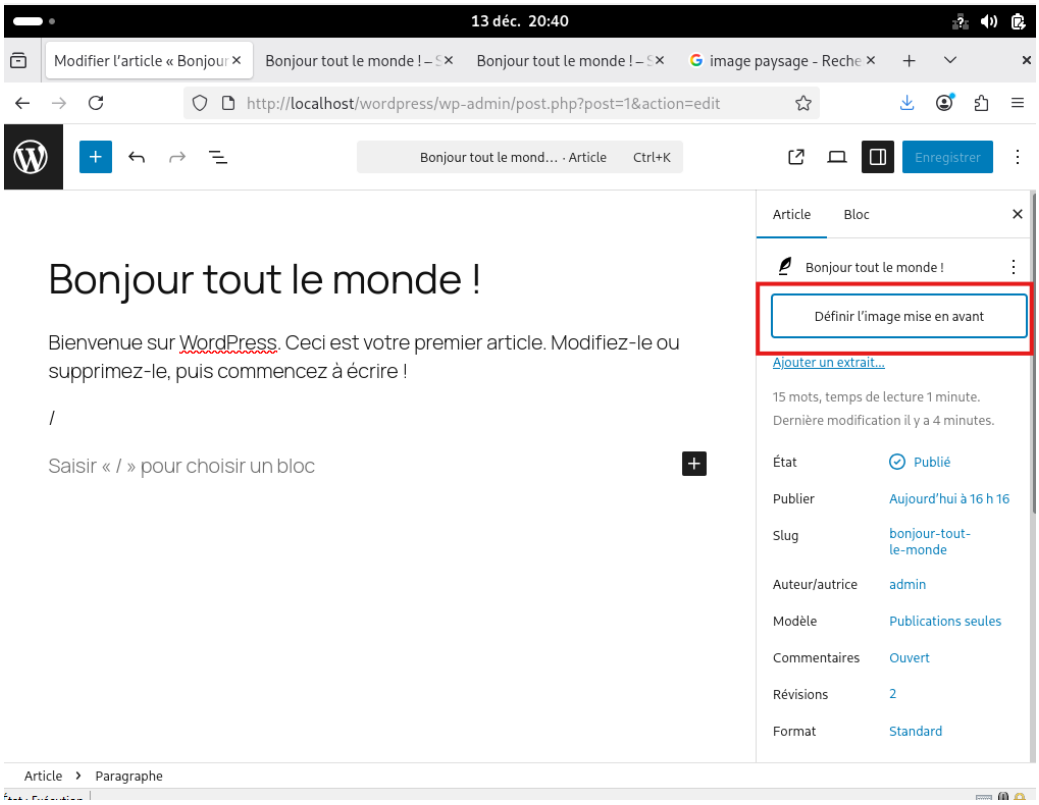


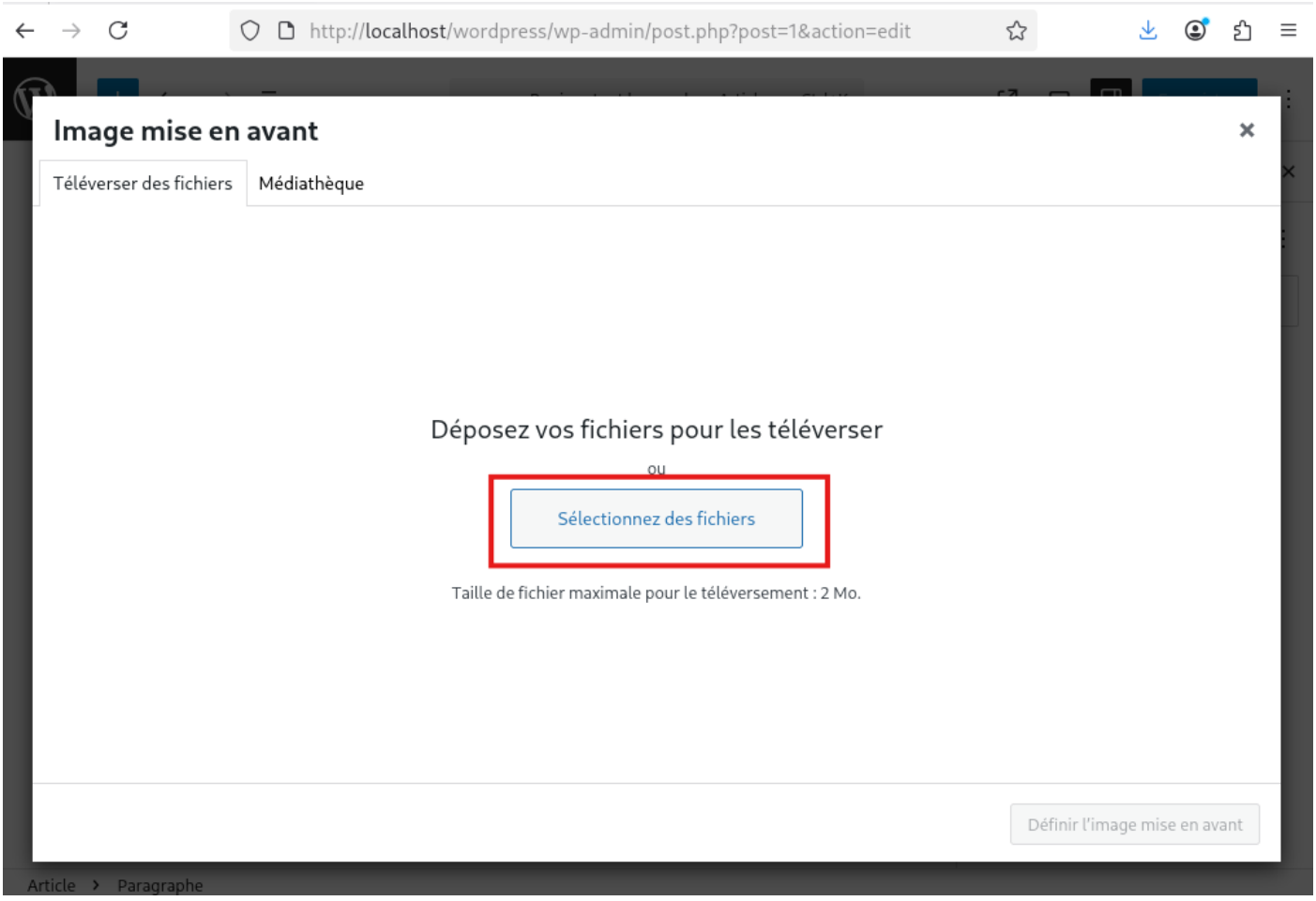
Une fois que nous avons enregistré le site, nous pouvons le consulter pour voir si la catégorie et le tag a bien été appliqué :



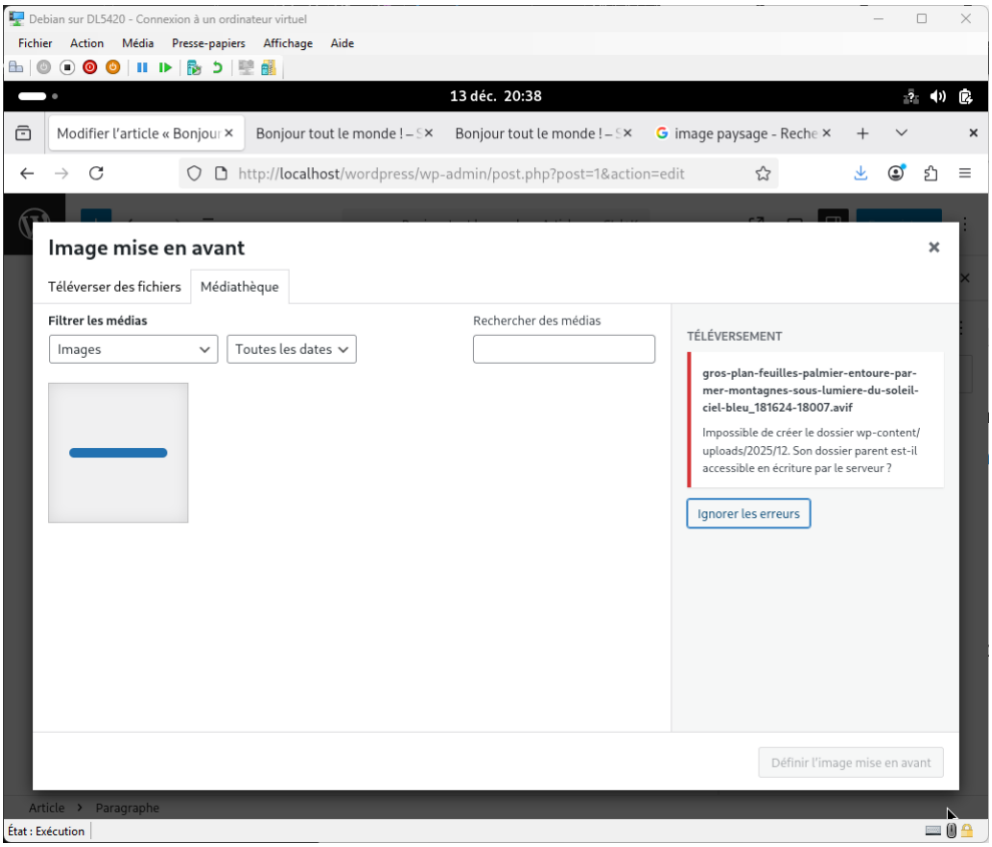
Ajouter une image pour illustrer l'article et une 2ème image dans un bloc en fin d'article.

Pour modifier l'image d'illustration, nous allons nous rendre dans le volet de droite, dans article, puis dans « Définir l'image mise en avant », puis nous allons sélectionner une image précédemment télécharge :





Si lors que l'on essaie de téléverser l'image, on obtient cette erreur :

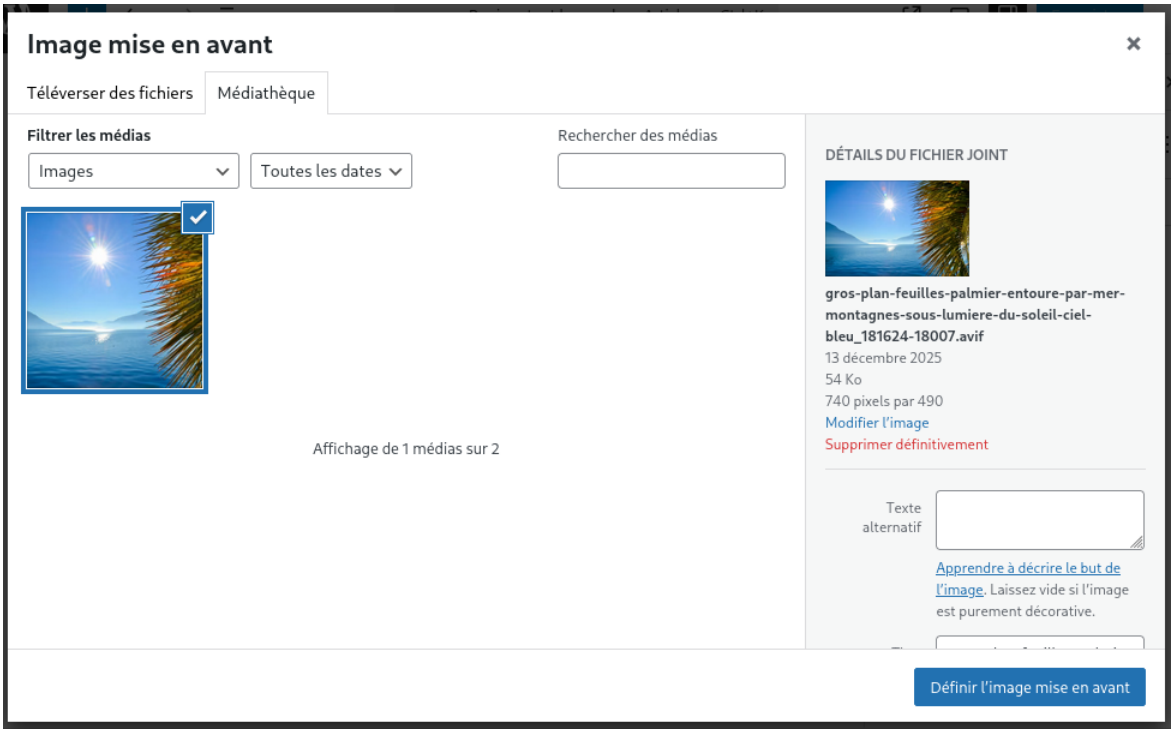


Cela signifie que le dossier wp-content ne peut pas être utilisé par le serveur web, pour régler ceci, il suffit d'executer la commande suivante sur un terminal : **chown -R www-data:www-data /var/www/html/wordpress/wp-content**

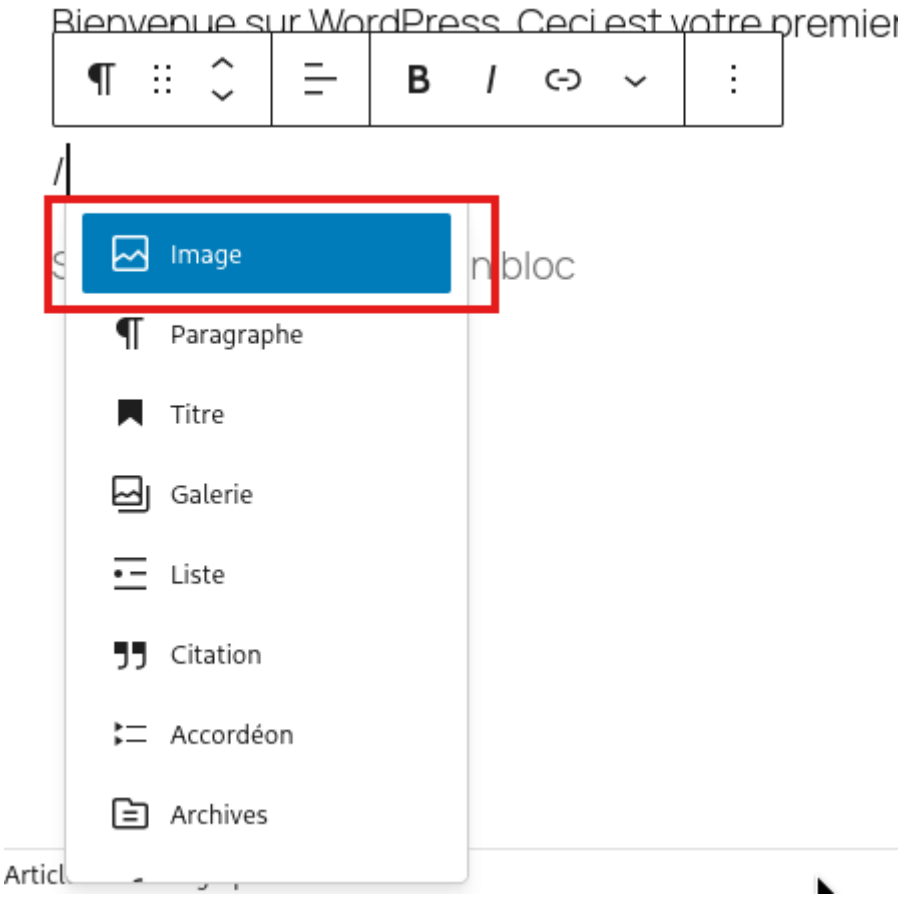
Cette commande va permettre au serveur d'avoir les droits sur le dossier wp-content :

```
root@ytest998:/var/www/html/wordpress# ls
index.php         wp-admin              wp-config-sample.php  wp-links-opml.php  wp-settings.php
license.txt       wp-blog-header.php    wp-content            wp-load.php        wp-signup.php
readme.html       wp-comments-post.php  wp-cron.php           wp-login.php       wp-trackback.php
wp-activate.php   wp-config.php         wp-includes           wp-mail.php        xmlrpc.php
root@ytest998:/var/www/html/wordpress# chown -R www-data:www-data wp-content/
root@ytest998:/var/www/html/wordpress#
```

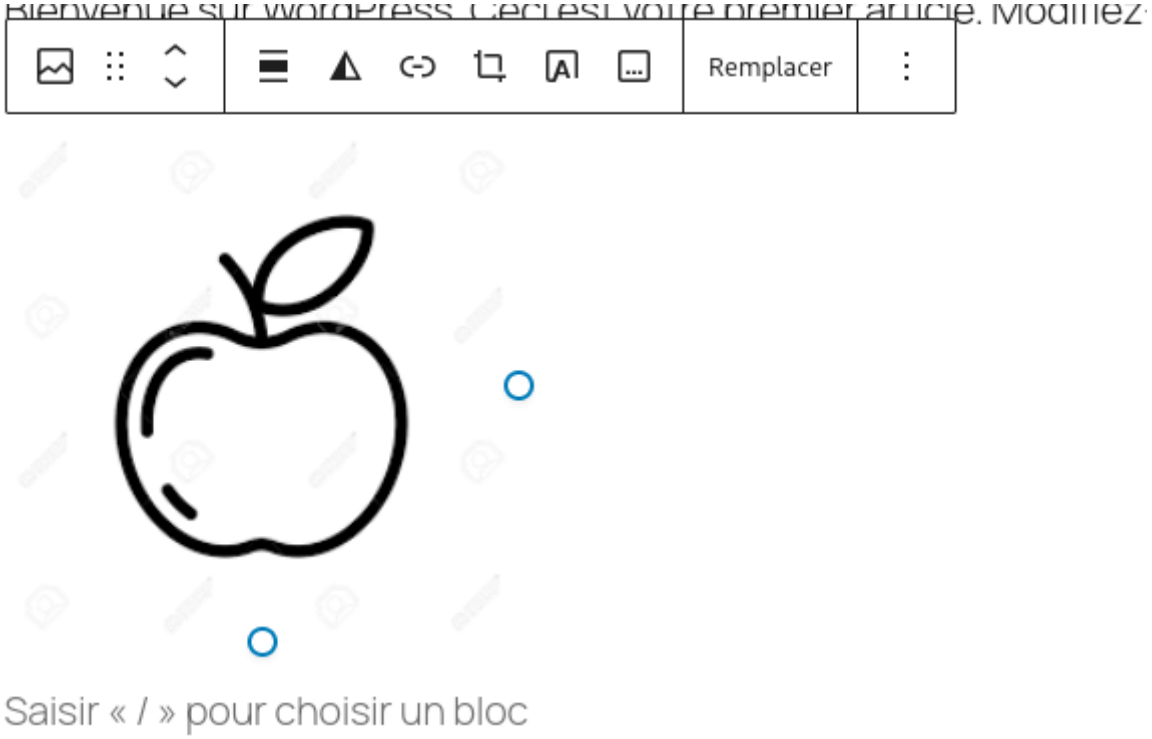
Maintenant, nous pouvons ajouter notre image sur WordPress :



Maintenant, nous allons définir une image dans un bloc, pour créer un bloc, il suffit de faire « / » dans la zone de texte et de sélectionner l'élément que l'on souhaite ajouter (en l'occurrence une image) :

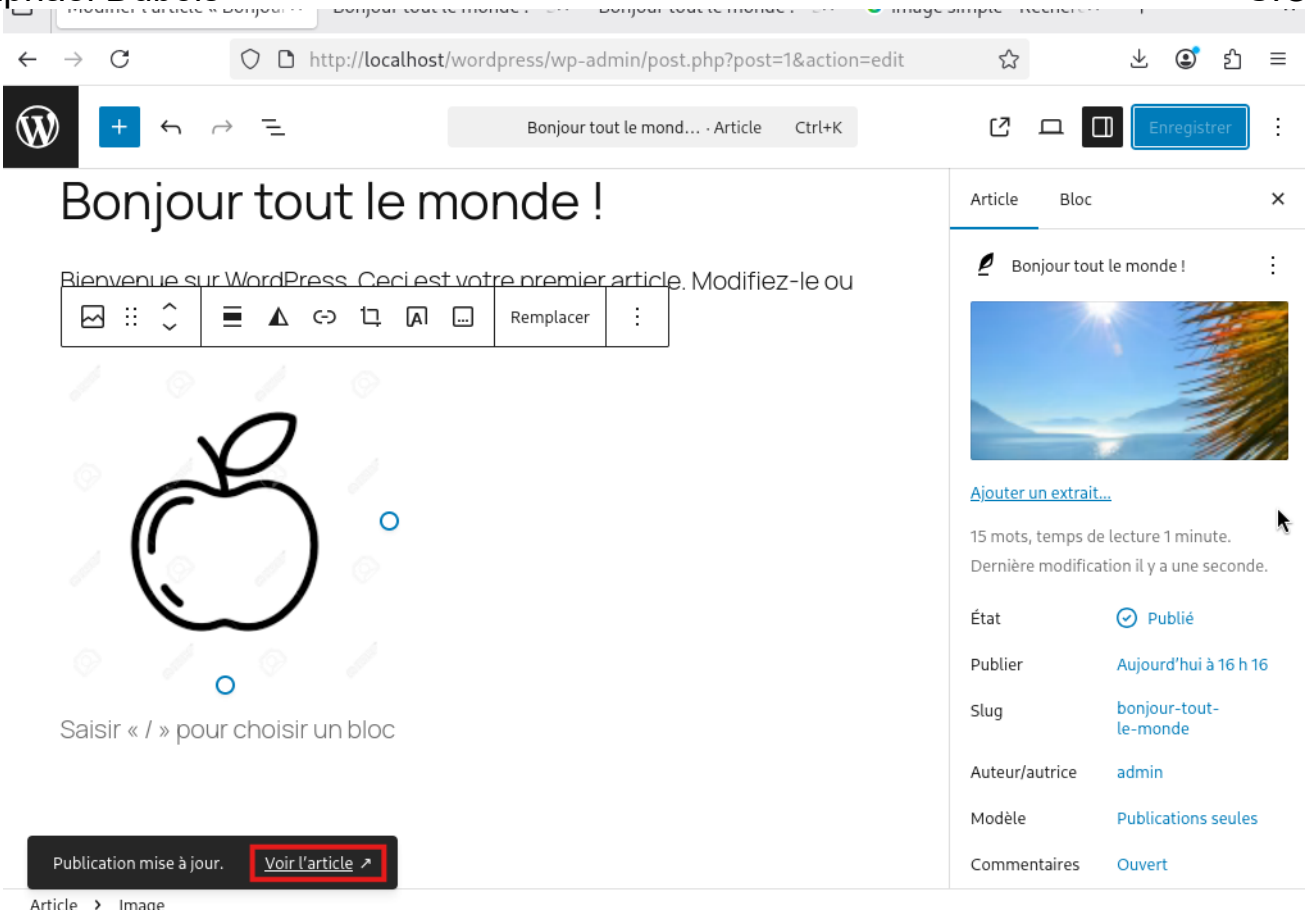


Une fois le bloc sélectionné, nous allons cliquer sur « téléverser », puis nous allons sélectionner notre image. Et enfin, une fois cette dernière sélectionnée, il est possible de la bouger comme bon il nous semble sur la page :



Publier l'article et dans un nouvel onglet de navigateur, vérifier votre site.

Une fois que nos modifications ont été effectuées, nous pouvons cliquer sur « enregistrer » en haut à droite, puis cliquer sur « Voir l'article » :



A présent, nous pouvons observer l'article que nous venons de créer de la vue d'un visiteur du site :



Écrit par admin dans Non classé, TEST1

Bienvenue sur WordPress. Ceci est votre premier article.
Modifiez-le ou supprimez-le, puis commencez à écrire !



TEST_TAG

IV/ Apache2

1) Historique

Internet : interconnexion mondiale de réseaux
1960 au ministère de la Défense américain
WWW : système hypertexte constitué de l'ensemble des pages servies par les serveurs utilisant le protocole http.*
1989 au CERN
1993 : Mosaïc, 1er navigateur convivial
1994 : Mosaïc est transformé en Netscape Navigator
2004 : Mozilla Firefox
1995 : Internet Explorer Microsoft

2) Le protocole HTTP

Déroulement type d'une connexion :

- Connexion du client à un serveur
- Envoi d'une requête au serveur
- Réponse du serveur
- Envoi d'une 2ème requête et fermeture de la connexion
- Deuxième réponse du serveur et fermeture de la connexion

Différentes requêtes possibles :

- GET : Récupération du contenu d'un document
- HEAD : Récupération des en tête seulement
- POST : Envoi de données au serveur
- PUT : Envoi d'un fichier
- DELETE : Suppression d'un fichier
- CONNECT : Accès à un serveur sécurisé HTTPS

Les entêtes, côté client :

- Host : indique au serveur quel hôte est interrogé
- Referer : L'origine du lien
- User-Agent : Le nom du navigateur
- Accept : Le format de fichier accepté
- Accept-language : La langue demandée
- Autorisation : Les informations d'authentification
- Connection : Pour fermer la connexion

Les entêtes, côté serveur :

- Date de la réponse
- Server : Logiciel utilisé (CMS)
- Content-Type : Format des données renvoyées
- Content-Length : Taille totale des fichier renvoyé

Les codes réponses sont classés 5 catégories :

- De 100 ç 199 : Message informatif
- De 200 à 299 : Succès de la requête
- De 300 à 399 : Redirection
  301 : Ressource déplacée à une nouvelle adresse
  307: Ressource momentanément déplacée

- De 400 à 499 : Problème dans la requête
  401 : Requête non autorisée
  404 : Le fichier n'existe pas
  405 : Méthode GET ou POST incorrecte
- 500 à 599 : Erreurs du serveur
  500 : Erreur interne
  503 : Service non disponible

1) Analyser de trame

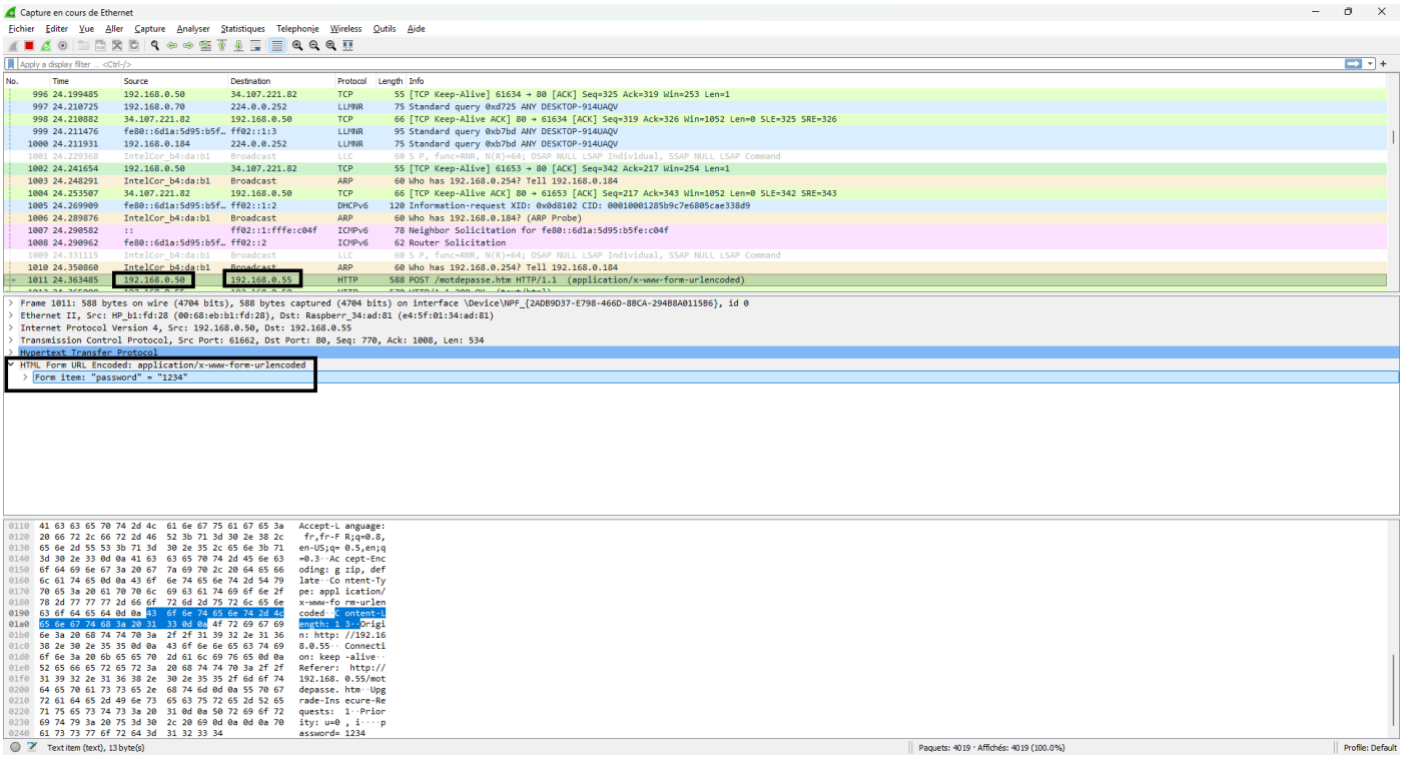
Créer un fichier motdepasse.htm dans /var/www/html :

```
<html>
  <head>
    <title>Test mot de passe</title>
  </head>
  <body>
    <h1>Test mot de passe</h1>
    <form action="motdepasse.htm" method="post">
      <input type="text" id="passe" name="password" />
      <button type="submit">Envoyer</button>
    </form>
  </body>
<html>
```

Maintenant, nous allons lancer WireShark, capturer les requetes http uniquement, puis nous allons aller sur la page que nous venons de créer, sur cette dernière, nous allons entrer un mot de passe et cliquer sur « Envoyer » :



Une fois ceci fait, nous pouvons retourner sur WireShark, trouver notre requête, et dérouler la partie « HTML Form » de notre requête, pour y retrouver le mot de passe que nous avions indiqué :



La foundation Apache

Apache Software Fondation : objectif logiciel libre, 1995

Pas de salariés mais des centaines de contributeurs

Centaine de projet dont HTTP Apache, Ldap Apache Directory (2002), OpenOffice (2011), Spam Assassin (2004)…

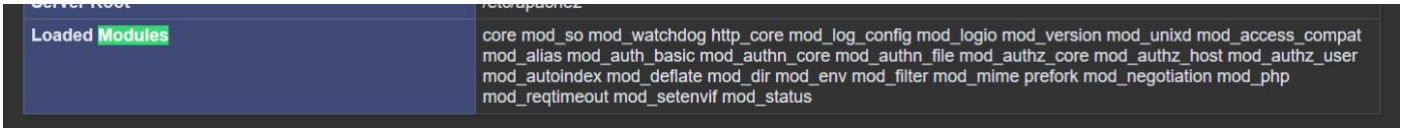
La license Apache est compatible avec le GNU GPL sauf qu'on n'est pas obligé de republier son code sans License Apache

HTTP Apache : V2.4 (2012), conception modulaire

Module : mod_php, mod_sql, mod_ssl

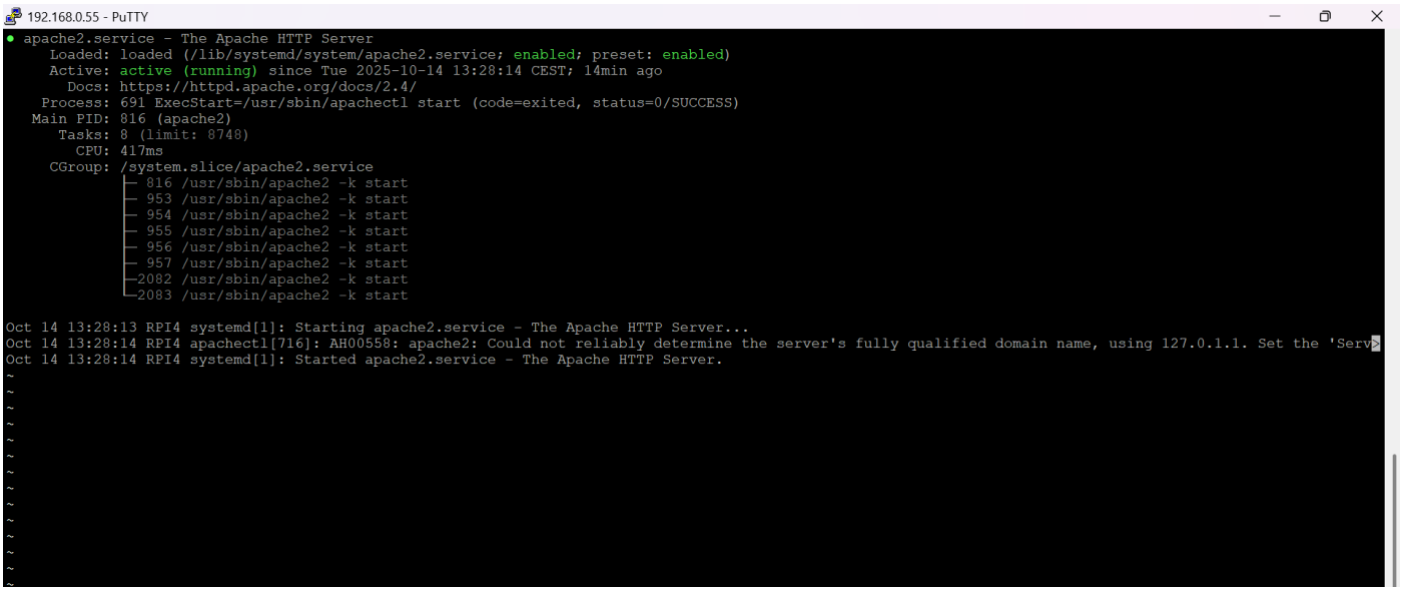
Trouver les modules installés sur votre Apache.

Si on se rend dans le fichier localhost/phpinfo.php précédemment créer, on peut avoir accès à la liste des modules installées



2) Configuration Apache

- Démarrage
  Status du démon ? Erreurs ou Warnings ?



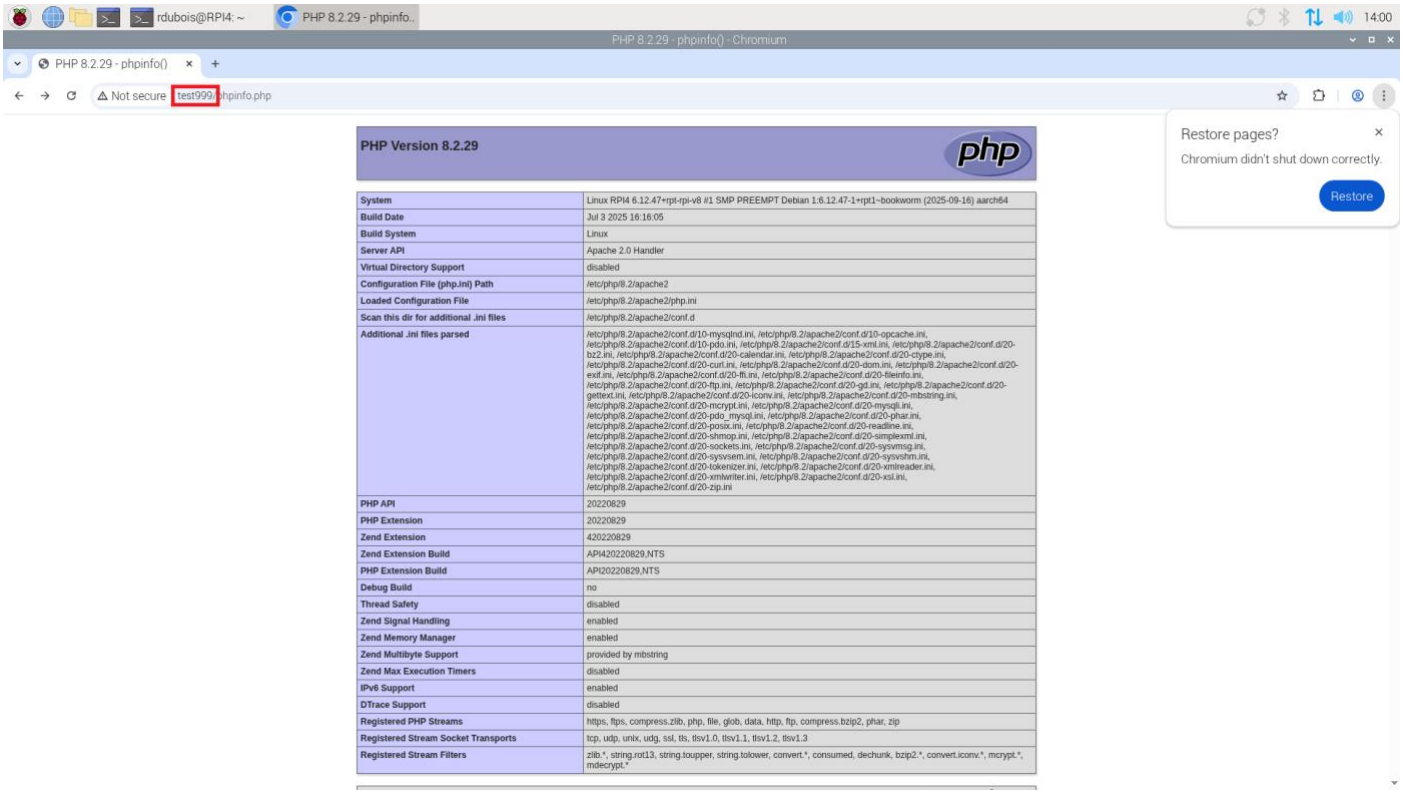
Le démon apache2 est actuellement actif, avec un warning qui nous indique que apache2 ne peut pas complétement déterminer un nom d'hôte.

Mettre un nom de machine avec la commande hostname

Commande hostname :

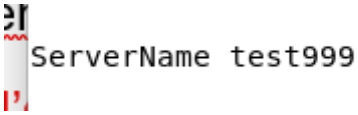


Test avec http://votre_nom



Dans /etc/apache2/apache2.conf, compléter le nom sur la ligne ServerName.

Redémarrer le démon !



Redémarrage du démon avec la commande : **sudo systemctl restart apache2**

Maintenant, si l'on modifie l'hostname du serveur, et qu'on l'essaie d'accéder à http://test999, il sera toujours possible d'y accéder même si le nom d'hôte a été modifié :

```
root@debian:/var/www/html# hostname ytest998
root@debian:/var/www/html#
```



- Dans apache2.conf ou dans conf-enabled

Quelle directive spécifie le port TCP d'écoute ?

Le port d'écoute TCP est défini dans le fichier : /etc/apache2/ports.conf

```
  GNU nano 7.2                                                  ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>
```

Quel est le port par défaut ?

Par défaut, il est défini sur le port 80 pour le HTTP, 443 pour HTTPS

Y a-t-il d'autres fichiers de conf ? Lesquels ?

Oui, l'architecture Apache sous Debian est modulaire. Voici les principaux fichiers et dossiers inclus dans apache2.conf :

- **sites-enabled/*.conf** : Contient les configurations des **VirtualHosts** (les sites actifs). Ce sont généralement des liens symboliques vers sites-available/.
- **conf-enabled/*.conf** : Contient des fragments de configuration globaux (ex: sécurité, jeux de caractères).
- **mods-enabled/*.load et *.conf** : Fichiers pour charger et configurer les **modules** actifs (ex: PHP, SSL, Rewrite).
- **.htaccess** : Fichiers de configuration décentralisés placés directement dans les répertoires du site web (si la directive AllowOverride le permet).

Quelle est la directive qui charge un module ?

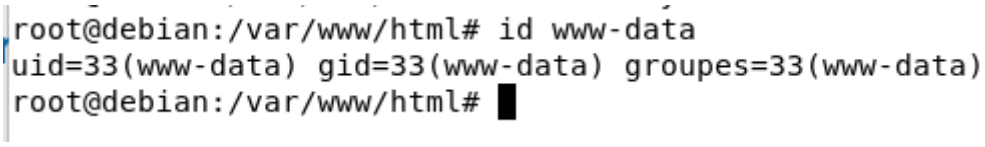
- **La Directive (LoadModule) :** C'est l'instruction technique que le logiciel Apache comprend. Elle doit être écrite dans un fichier de configuration pour que le module fonctionne.
  Exemple dans un fichier : LoadModule rewrite_module /usr/lib/apache2/modules/mod_rewrite.so

- **La Commande (a2enmod) :** C'est un script spécifique aux distributions Debian/Ubuntu/Mint. Il sert à **automatiser** l'écriture ou l'inclusion de la directive LoadModule.

  Nom complet : **A**pache **2 En**able **Mod**ule.

Sous quelle identité (user,group) fonctionne le serveur ? WID et GID ?

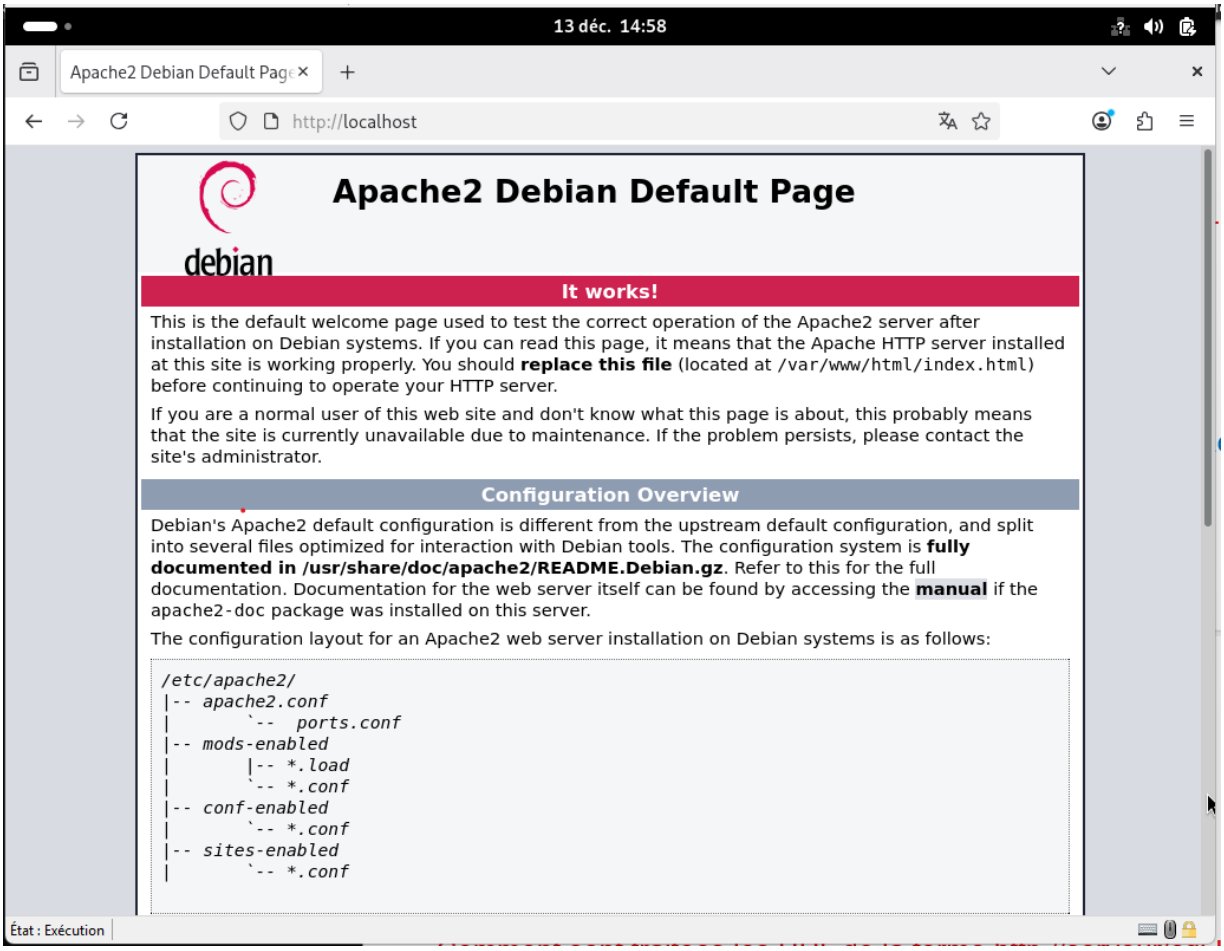
- **Utilisateur (User) :** www-data
- **Groupe (Group) :** www-data
- **UID / GID :** Généralement, l'UID et le GID pour www-data sont **33** (cela peut varier selon l'installation, on peut le vérifier avec la commande id www-data).



Quel est le répertoire racine du site ?

Le répertoire racine du site est le répertoire : /var/www/html accessible via localhost/ sur un navigateur :

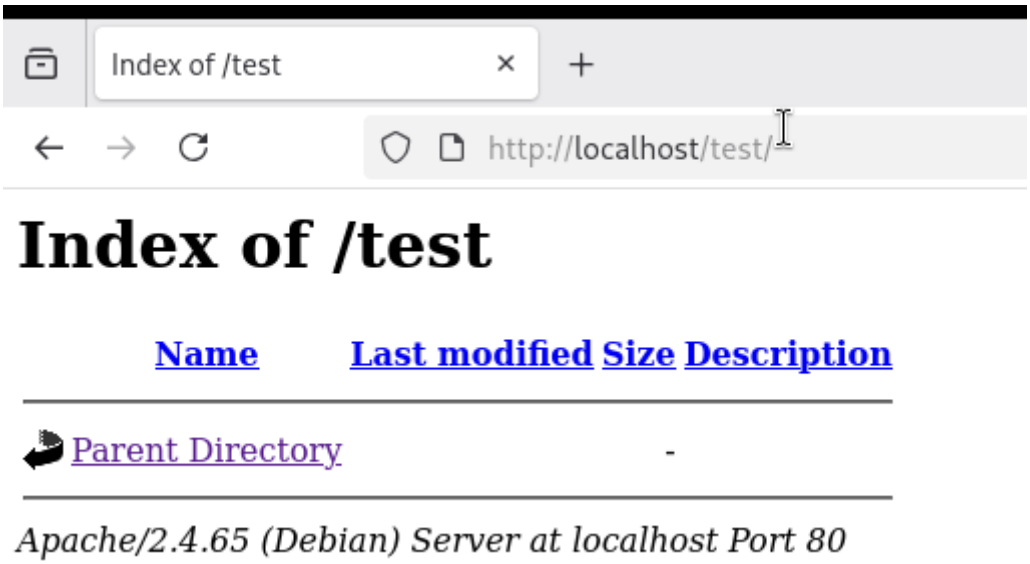
```
root@debian:/etc# cd /var/www/html/
root@debian:/var/www/html# ls
index.html
root@debian:/var/www/html#
```



Quelle page Apache renvoie lorsque l'URL est un dossier ?

Par défaut, Apache renvoie le contenu du dossier (exemple avec le dossier test) :

```
root@debian:/var/www/html# mkdir test
root@debian:/var/www/html# ls
index.html  test
root@debian:/var/www/html#
```



Comment sont traitées les URL de la forme http://serveur/cgi-bin/toto

Le serveur ne renvoie pas le contenu du fichier toto tel quel. Il **exécute** le fichier toto en tant que script (CGI) sur le serveur, puis renvoie le résultat (la sortie standard du script) au client. Cela permet d'exécuter des scripts Perl, Python ou Shell.

Quels sont les fichiers de logs générés et où ?

Par défaut, les logs sont stockés dans le répertoire **/var/log/apache2/**.

Il y a deux types principaux :

1. **access.log** : Enregistre toutes les requêtes reçues (qui, quand, quelle page, code réponse).
2. **error.log** : Enregistre les erreurs du serveur (démarrage, arrêt, bugs de scripts, fichiers introuvables).

Dans quel fichier de log peut-on constater le démarrage ?

Pour ouvrir le fichier de log concernant le démarrage d'apache2, nous devons utiliser la commande journalctl -u apache2, on peut constater que le serveur a bien redémarrer après l'exécution de la commande :

```
root@RPI4:/var/log/apache2# systemctl restart apache2
root@RPI4:/var/log/apache2# journalctl -u apache2
Oct 14 14:40:02 RPI4 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 14 14:40:03 RPI4 systemd[1]: Started apache2.service - The Apache HTTP Server.
Oct 16 10:17:02 RPI4 systemd[1]: Reloading apache2.service - The Apache HTTP Server...
Oct 16 10:17:02 RPI4 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
Oct 16 10:42:53 RPI4 systemd[1]: Stopping apache2.service - The Apache HTTP Server...
Oct 16 10:42:53 RPI4 systemd[1]: apache2.service: Deactivated successfully.
Oct 16 10:42:53 RPI4 systemd[1]: Stopped apache2.service - The Apache HTTP Server.
Oct 16 10:42:53 RPI4 systemd[1]: apache2.service: Consumed 1.143s CPU time.
Oct 16 10:42:53 RPI4 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 16 10:42:53 RPI4 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@RPI4:/var/log/apache2#
```

Accéder à la page d'accueil depuis une autre machine

A-t-on une trace dans les logs ? Où ?

Oui, on a une trace dans le fichier /var/log/apache2/access.log, on peut voir quel IP c'est connecté, sur quel page, à quel heure avec son user agent…

```
root@RPI4:/var/log/apache2# cat access.log
192.168.0.179 - - [16/Oct/2025:10:22:47 +0200] "GET / HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:144.0) Gecko/20100101 Firefox/144.
0"
192.168.0.179 - - [16/Oct/2025:10:22:47 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 200 6040 "http://192.168.0.55/" "Mozilla/5.0 (Windows NT 10.0; Win64; x6
4; rv:144.0) Gecko/20100101 Firefox/144.0"
192.168.0.179 - - [16/Oct/2025:10:22:47 +0200] "GET /favicon.ico HTTP/1.1" 404 490 "http://192.168.0.55/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:144.0
) Gecko/20100101 Firefox/144.0"
192.168.0.179 - - [16/Oct/2025:10:31:49 +0200] "GET / HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:144.0) Gecko/20100101 Firefox/144.
0"
192.168.0.179 - - [16/Oct/2025:10:31:50 +0200] "GET /favicon.ico HTTP/1.1" 404 490 "http://192.168.0.55/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:144.0
) Gecko/20100101 Firefox/144.0"
root@RPI4:/var/log/apache2#
```

- Accès aux dossiers

Créer un dossier /var/www/html/test et mettre dedans le fichier test.htm :

```
<html>
  <head>
    <title>Test</title>
  </head>
  <body>
    <h1>Ceci est un test !</h1>
  </body>
</html>
```

Création du dossier et du fichier grâce aux commandes suivantes

mkdir /var/www/html/test

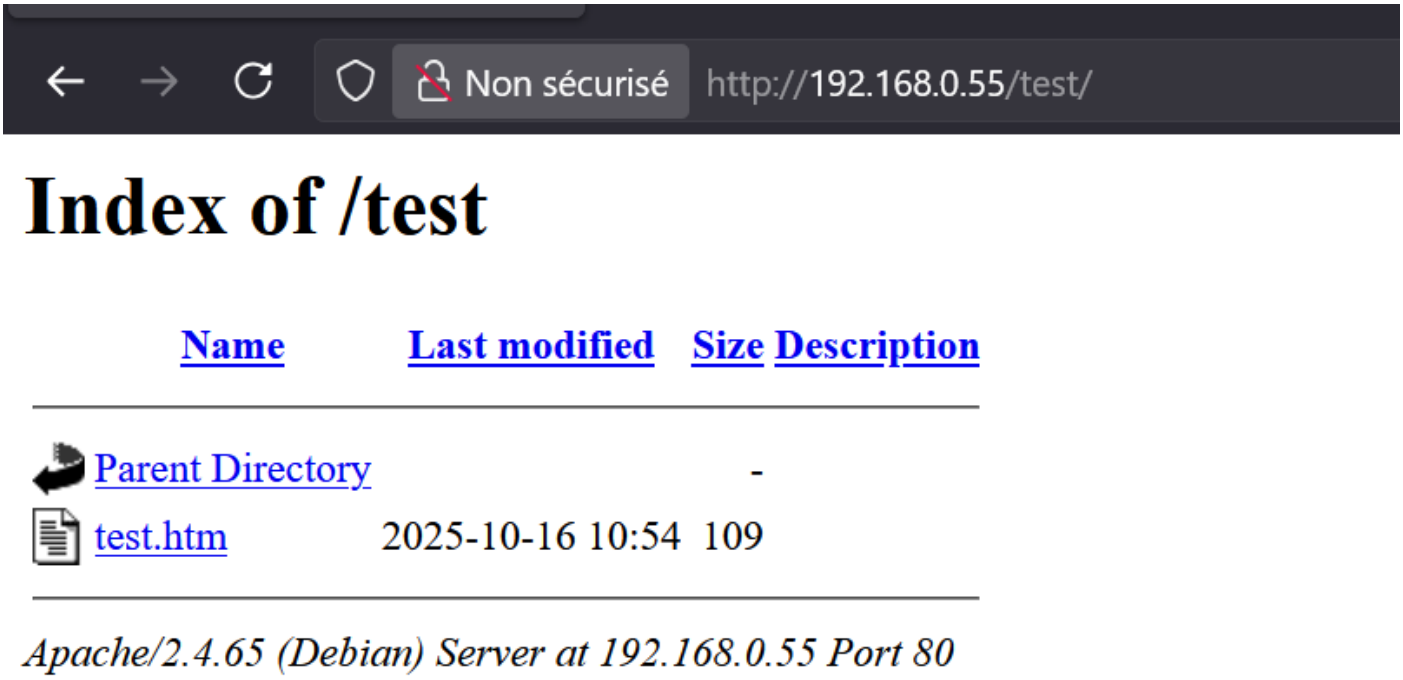
nano /var/www/html/test/test.htm

Accéder au fichier test.htm dans un navigateur.



Accéder au dossier test (sans le test.htm)

Qu'obtient-on ?



On obtient le contenu du dossier test

Renommer test.htm en index.htm et actualiser le navigateur

Qu'obtient-on ? Pourquoi ?

Si on modifie le fichier test.htm en index.htm, et qu'on se rend dans le répertoire test, le contenu du répertoire ne sera plus listé et la page sera automatiquement affiché, étant donné que le fichier index est la page par défaut que le site doit charger.



Ceci est un test !

Pour lister (ou pas) le dossier test, on peut spécifier dans le fichier apache 2.conf :

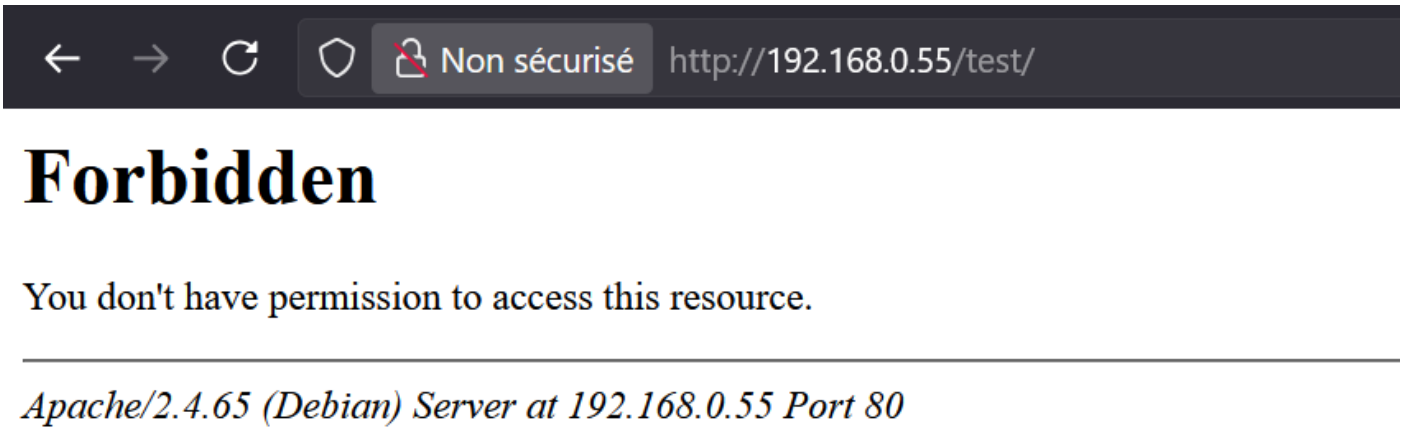
<Directory /var/www/html/test>

Options -Indexes

</Directory>

Qu'est-ce que ça change ?

Si on modifie la configuration d'apache2 avec ces option, et qu'on modifie le fichier index.htm en test.htm, et qu'on essaie d'accéder à nouveau au répertoire, on obtient une erreur « 403 Forbidden » ; ce qui signifie que le serveur ne nous autorise pas à lister les fichiers présents dans le dossier test, ce qui est important à activer pour plus de sécurité sur notre serveur.



3) Scripts CGI simples

1. Ecrire un script en shell bash nommé date.cgi qui renvoie la date et l'heure dans une page html :

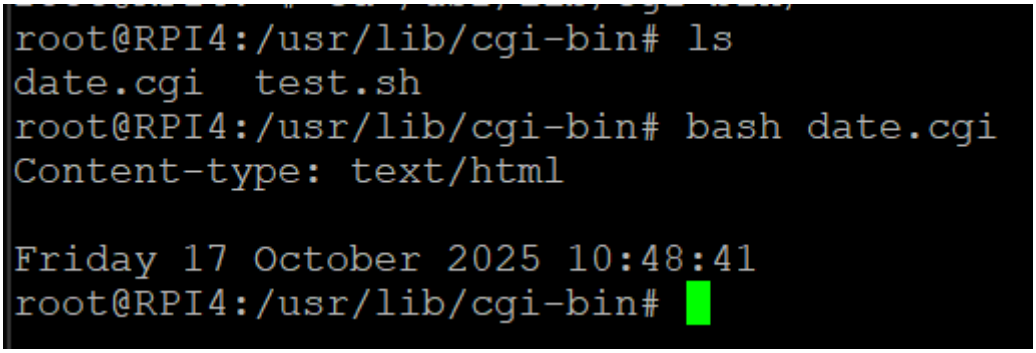
#!/bin/bash

echo "Content-Type: text/html"

```
echo ""

date +"%A %d %B %Y %T"
```

Ou enregistrez-vous le script ?

Le script doit être enregistré dans le répertoire suivant : /usr/lib/cgi-bin



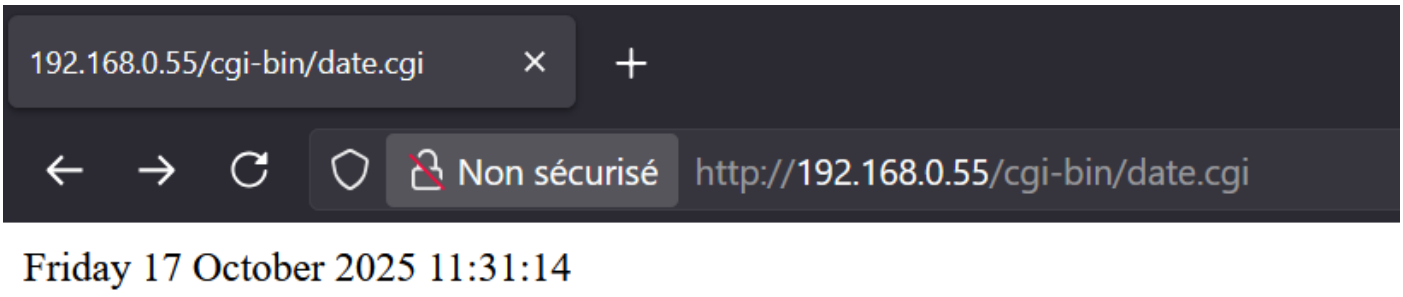
Faut-il configurer quelque chose ?

Il faut activer le module cgi sur notre serveur grâce à la commande suivante : sudo a2enmod cgi

Une fois cette commande lancé, il faut redémarrer notre service apache2 afin que cgi soit actif : systemctl restart apache2

Une fois ceci fait, nous devons aussi rendre le script exécutable grâce à la commande suivante : chmod +x /usr/lib/cgi-bin/date.cgi

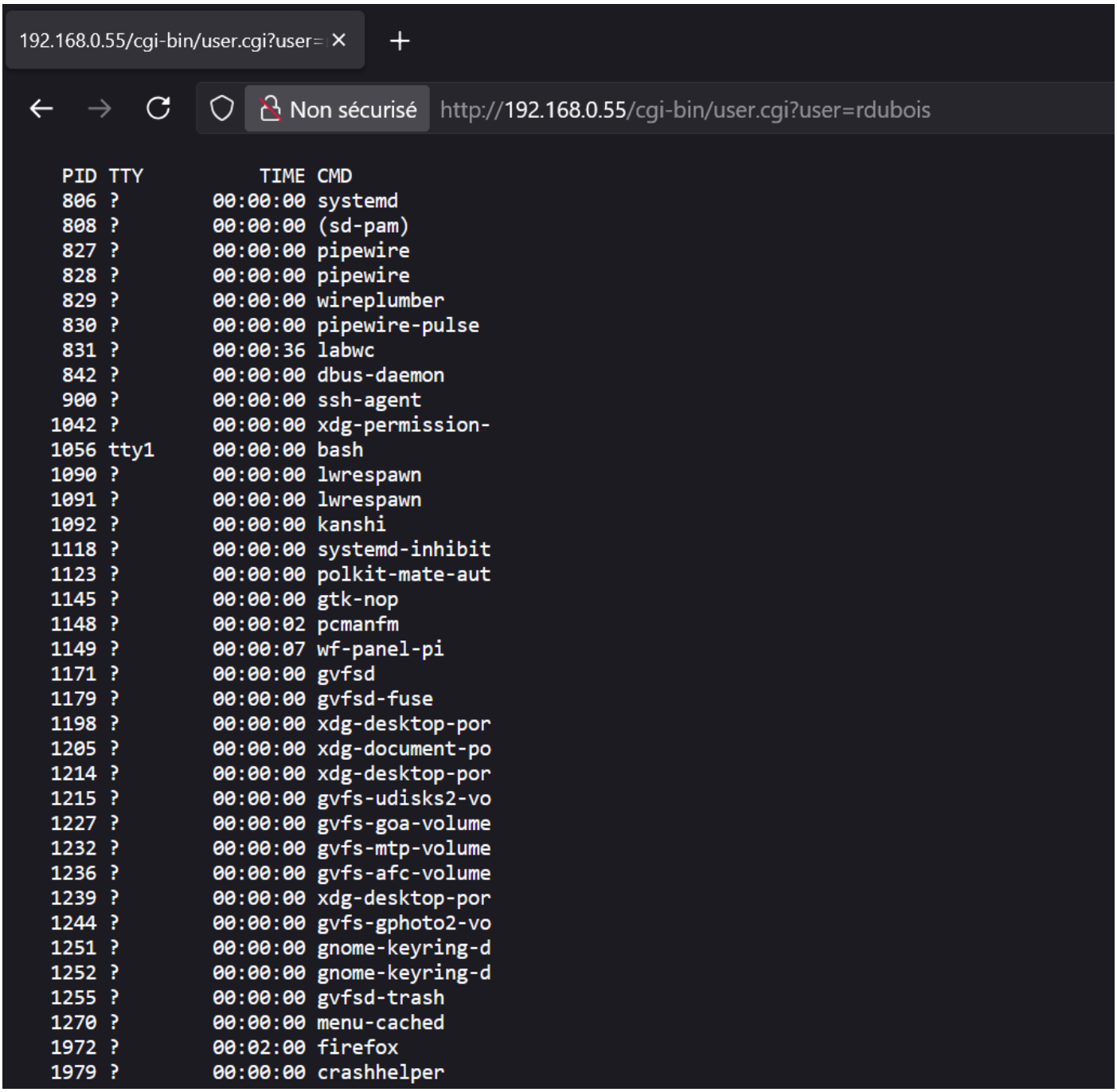
Une fois que nous avons tout configurer, nous pouvons se rendre sur ip/cgi-bin/date.cgi pour vérifier que notre script fonctionne dans apache !



2. Ecrire un CGI qui affiche la liste des processus appartenant à un utilisateur donné : user.cgi

#!/bin/bash

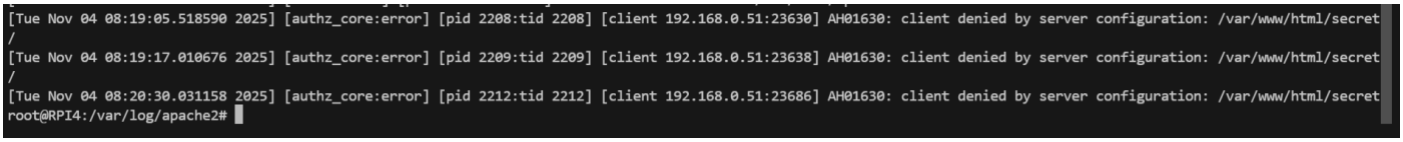
```
echo "Content-type: text/plain"

echo ""


user=$(echo "$QUERY_STRING" | grep -oP '(?<=user=)\w+')

ps -u $user
```




```
  PID TTY          TIME CMD
  806 ?        00:00:00 systemd
  808 ?        00:00:00 (sd-pam)
  827 ?        00:00:00 pipewire
  828 ?        00:00:00 pipewire
  829 ?        00:00:00 wireplumber
  830 ?        00:00:00 pipewire-pulse
  831 ?        00:00:36 labwc
  842 ?        00:00:00 dbus-daemon
  900 ?        00:00:00 ssh-agent
 1042 ?        00:00:00 xdg-permission-
 1056 tty1     00:00:00 bash
 1090 ?        00:00:00 lwrespawn
 1091 ?        00:00:00 lwrespawn
 1092 ?        00:00:00 kanshi
 1118 ?        00:00:00 systemd-inhibit
 1123 ?        00:00:00 polkit-mate-aut
 1145 ?        00:00:00 gtk-nop
 1148 ?        00:00:02 pcmanfm
 1149 ?        00:00:07 wf-panel-pi
 1171 ?        00:00:00 gvfsd
 1179 ?        00:00:00 gvfsd-fuse
 1198 ?        00:00:00 xdg-desktop-por
 1205 ?        00:00:00 xdg-document-po
 1214 ?        00:00:00 xdg-desktop-por
 1215 ?        00:00:00 gvfs-udisks2-vo
 1227 ?        00:00:00 gvfs-goa-volume
 1232 ?        00:00:00 gvfs-mtp-volume
 1236 ?        00:00:00 gvfs-afc-volume
 1239 ?        00:00:00 xdg-desktop-por
 1244 ?        00:00:00 gvfs-gphoto2-vo
 1251 ?        00:00:00 gnome-keyring-d
 1252 ?        00:00:00 gnome-keyring-d
 1255 ?        00:00:00 gvfsd-trash
 1270 ?        00:00:00 menu-cached
 1972 ?        00:02:00 firefox
 1979 ?        00:00:00 crashhelper
```

4) Protection des accès

- Créer un répertoire secret à la racine du site et placez dedans une page d'accueil. Configurer Apache pour que vous puissiez accéder à cette page uniquement depuis le serveur.
  Quel est le code d'erreur renvoyé par Apache lorsqu'on tente d'accéder depuis une autre machine ? Qu'observe-t-on dans les logs ?



- On souhaite protéger l'accès au sous-dossier /var/www/html/prive Il devra être accessible qu'à un ensemble limité de comptes utilisateurs Apache (différents des utilisateurs Linux). Une requête sur ce dossier provoquera l'affichage d'une page d'authentification.

Création de dossier prive et de sa page d'accueil.

```
root@debian:/var/www/html# mkdir prive
root@debian:/var/www/html# cd prive/
root@debian:/var/www/html/prive# echo prive > index.html
root@debian:/var/www/html/prive#
```

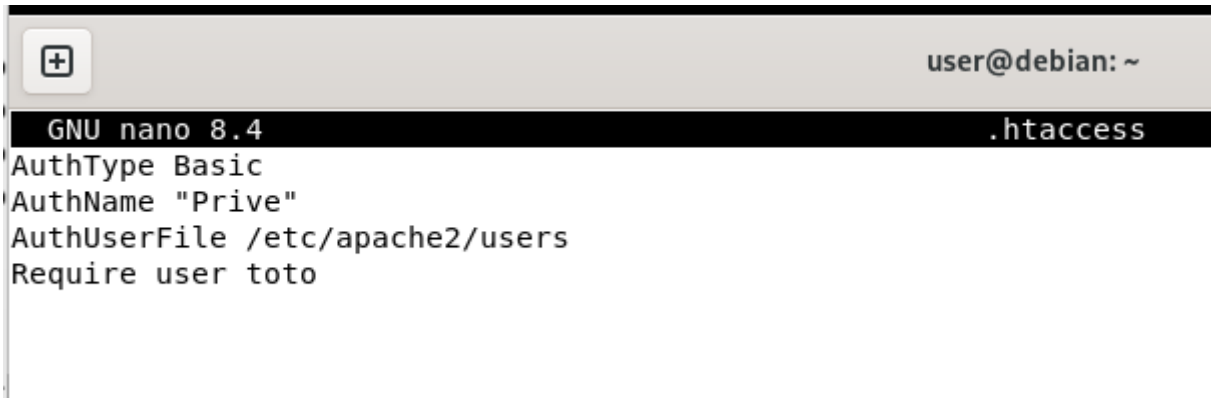
Test de la page privé avant modification des permissions d'accès :



Création de fichier .htaccess :

```
AuthType Basic

AuthName "Prive"

AuthUserFile /etc/apache2/users # Fichier des utilisateurs
```

```
Require user toto
```



Création de l'utilisateur Apache toto :

```
cd /etc/apache2

htpasswd -c users toto
```

```
root@debian:/etc/apache2# htpasswd -c users toto
New password:
Re-type new password:
Adding password for user toto
root@debian:/etc/apache2#
```

Vérifie-le fichier /etc/apache2/users

L'utilisateur et le mot de passe (chiffré) est bien présent dans le fichier users :

```
root@debian:/etc/apache2# cat users
toto:$apr1$WiCiYUxj$0Z/7BNv3am2o9faJlYMKX1
root@debian:/etc/apache2#
```

Cela ne fonctionne pas !

Rajouter dans le fichier de configuration d'Apache :

<Directory /var/www/html/prive>

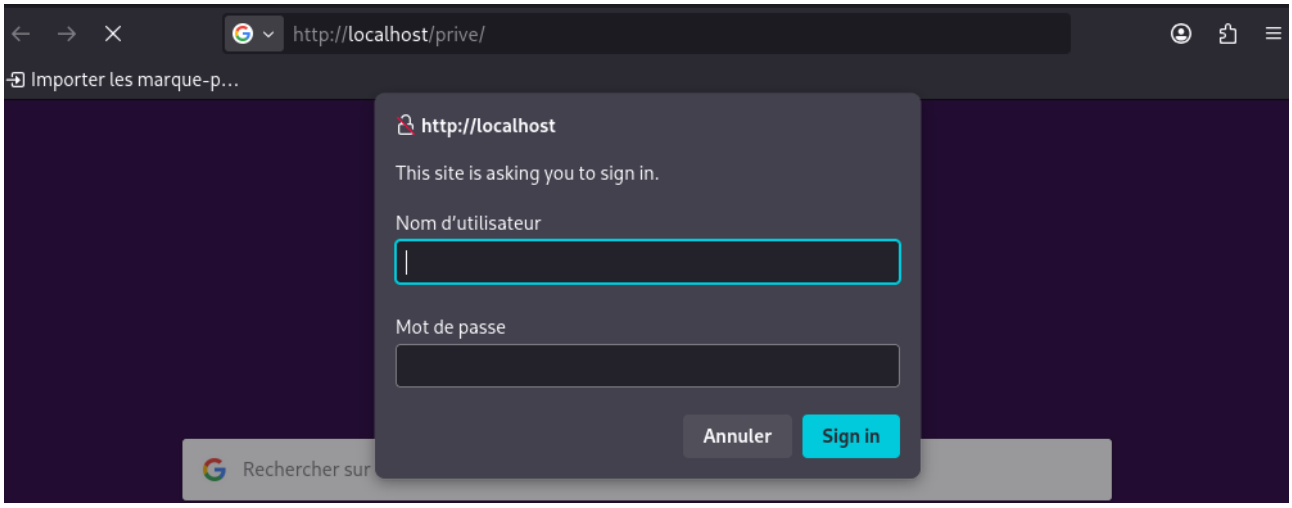
AllowOverride AuthConfig

Options -Indexes

</Directory>

Après avoir rajouté ceci, il ne faut pas oublier de redémarrer le démon Apache2 : **sudo systemctl restart apache2**

Maintenant, lors que l'on va tenter de se connecter sur le page /prive, nous auront une boite de dialogue nous demandant un utilisateur et un mot de passe :



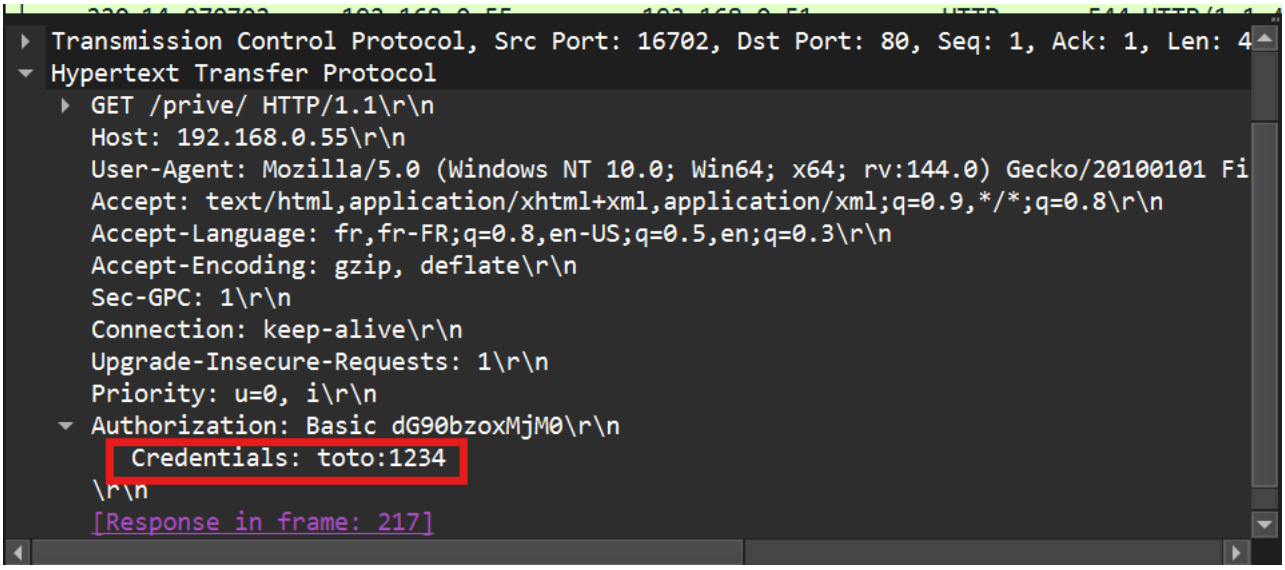
Il suffit d'entrer notre nom d'utilisateur toto avec son mot de passe pour accéder à la page privée :



prive

Capturer le mot de passe. Est-il en clair ?

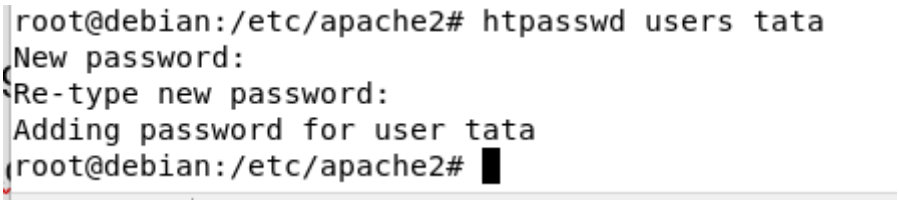
En utilisant Wireshark pour regarder la trame, on remarque que le mot de passe est affiché en clair dans la requête :



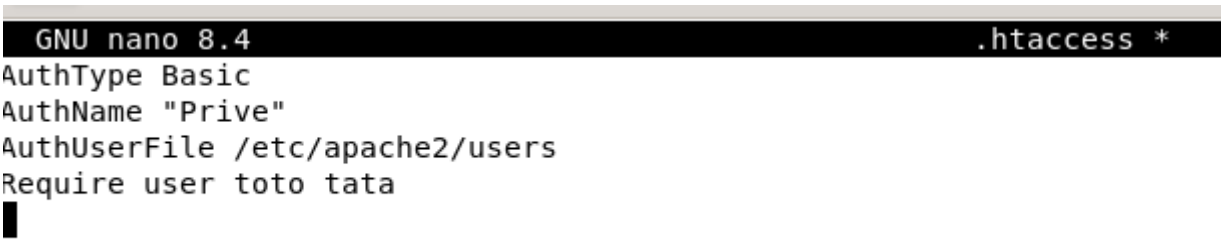
Rajouter un utilisateur tata.

Pour rajouter un utilisateur tata sans écraser toto, il suffit de lancer la même commande que précédemment mais sans l'option « -c » qui signifie create (créer le fichier) suivi du nom d'utilisateur que l'on souhaite créer :

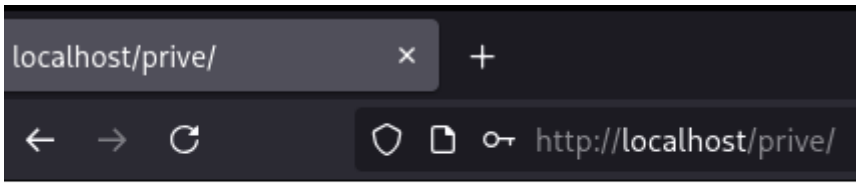
```
cd /etc/apache2

htpasswd users tata
```



Une fois ceci fait, nous allons modifier notre fichier .htaccess pour y inclure l'utilisateur « tata » sur la ligne « Require user » :



Maintenant nous pouvons tenter de se connecter avec l'utilisateur tata et constater que la connexion fonctionne :



prive

5) Serveurs virtuels

Pour héberger plusieurs sites sur un seul serveur, on utilise des VirtualHosts.

Dans cette partie, nous allons créer un VirtualHost pour site1 et site2

Premièrement, nous allons créer 2 dossiers sur notre serveur (site1 et site2), puis nous allons y créer un fichier index.html avec comme contenu, le nom du site (site1 et site2) :

```
root@debian:/var/www/html# mkdir site1
root@debian:/var/www/html# mkdir site2
root@debian:/var/www/html# echo site1 > site1/index.html
root@debian:/var/www/html# echo site2 > site2/index.html
root@debian:/var/www/html#
```

Maintenant, nous allons modifier notre fichier apache2.conf pour y inclure les différents hôtes virtuels :

<VirtualHost localhost>

DocumentRoot /var/www/html/site1

ServerName site1.localhost

</VirtualHost>

<VirtualHost localhost>

DocumentRoot /var/www/html/site2

ServerName site2.localhost

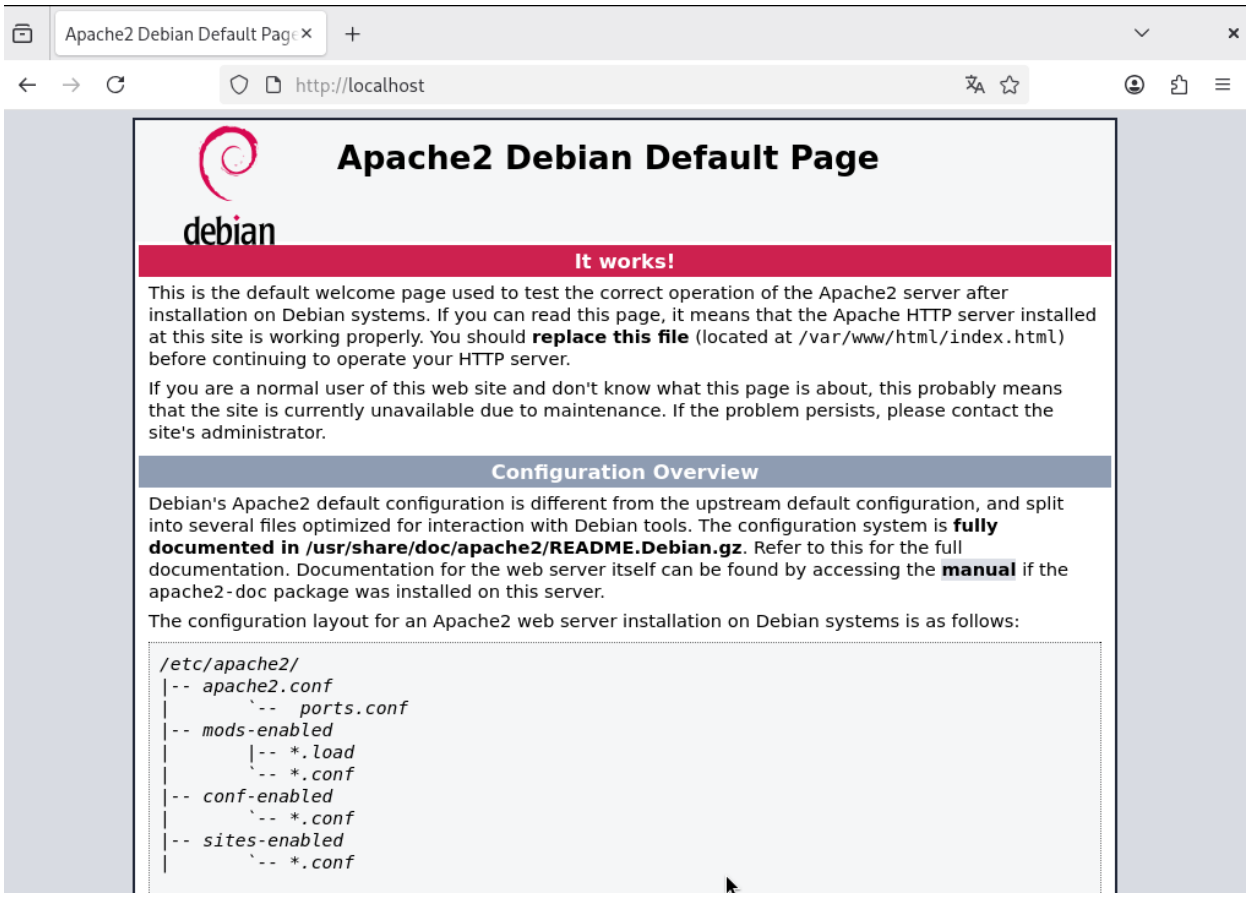
</VirtualHost>

```
ServerName localhost # Site par défaut

<VirtualHost localhost> # Site 1
        DocumentRoot /var/www/html/site1
        ServerName site1.localhost
</VirtualHost>

<VirtualHost localhost> # Site 2
        DocumentRoot /var/www/html/site2
        ServerName site2.localhost
</VirtualHost>
```

Après ceci, ne pas oublier de redémarrer le démon : **sudo systemctl restart apache2**

Maintenant, nous pouvons vérifier nos différents sites en y tapant l'URL dans notre navigateur :

Site par défaut :



Site1 :



Site2 :



6) HTTPS : le http sécurisé

Par défaut, apache utilise le protocole http sur le port 80

Les mots de passe transitent donc en clair !

Le HTTPS rajoute une couche sécurisée SSL au http, par défaut sur le port 443. On utilise le certificat par défaut d'Apache2 ; On active le module SSL :

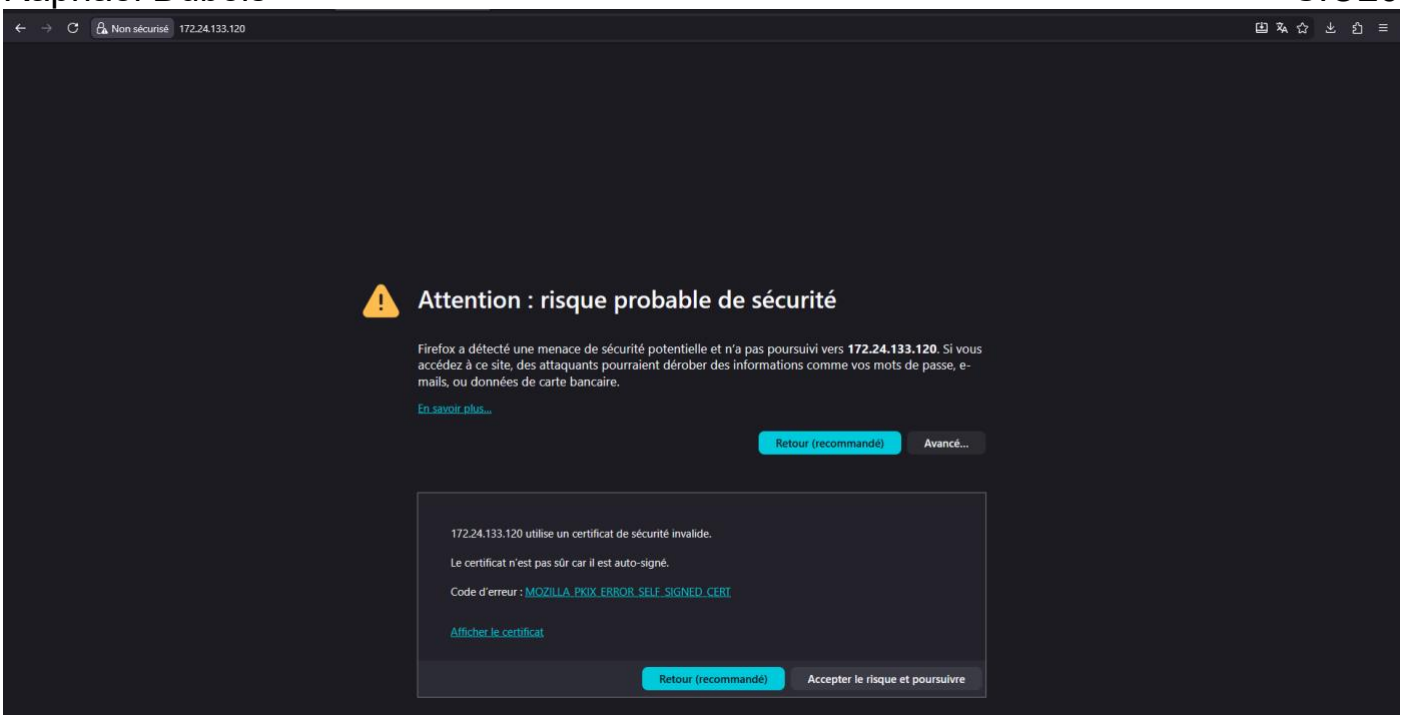
a2enmod ssl

a2ensite default-ssl

systemctl restart apache2

```
root@debian:/etc/apache2# /usr/sbin/a2enmod ssl
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 self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:/etc/apache2# /usr/sbin/a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian:/etc/apache2# systemctl restart apache2
root@debian:/etc/apache2#
```

Vérification du site :



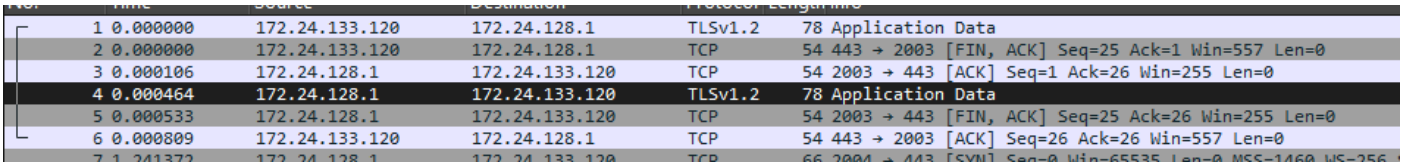
Cet avertissement signifie que le certificat est auto-signé, ce qui est normal dans notre cas (le serveur a auto signé un certificat). Il suffit de cliquer sur Accepter le risque et poursuivre

Nous sommes maintenant connectés en HTTP**S** sur notre site :



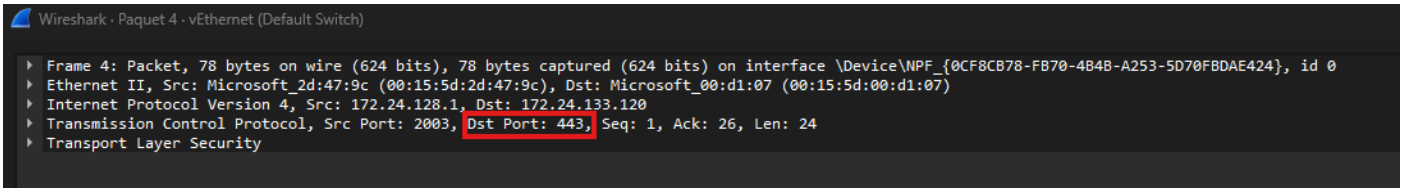
- Vérifier que le site fonctionne en sécuriser et contrôler le chiffrement avec WireShark.

Si l'on observe les requêtes les requêtes dans WireShark, nous remarquons que nous avons plus de requêtes http, mais des requêtes TLS en direction du serveur Apache2 :



1	0.000000	172.24.133.120	172.24.128.1	TLSv1.2	78 Application Data
2	0.000000	172.24.133.120	172.24.128.1	TCP	54 443 → 2003 [FIN, ACK] Seq=25 Ack=1 Win=557 Len=0
3	0.000106	172.24.128.1	172.24.133.120	TCP	54 2003 → 443 [ACK] Seq=1 Ack=26 Win=255 Len=0
4	0.000464	172.24.128.1	172.24.133.120	TLSv1.2	78 Application Data
5	0.000533	172.24.128.1	172.24.133.120	TCP	54 2003 → 443 [FIN, ACK] Seq=25 Ack=26 Win=255 Len=0
6	0.000809	172.24.133.120	172.24.128.1	TCP	54 443 → 2003 [ACK] Seq=26 Ack=26 Win=557 Len=0

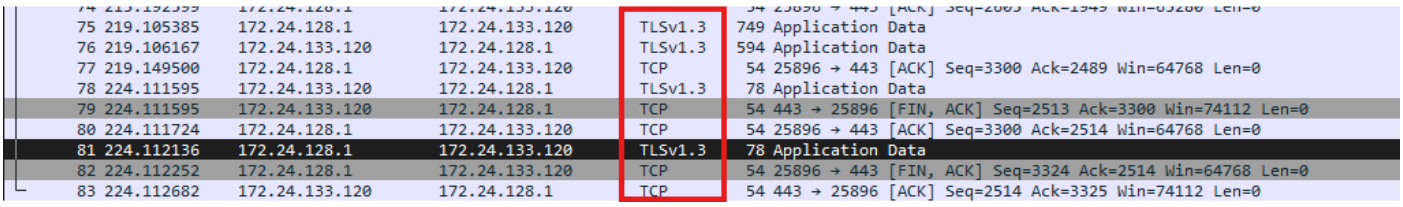
Si l'on décide de regarder la requête de plus près, on peut remarquer que le port cible est le port 443, étant le port par défaut du protocole HTTPS :



Si on essaie de taper le mot de passe sur la page motdepasse.htm, il ne sera pas possible de voir les données (et donc le mot de passe) étant donné que les données entre le serveur et le client sont chiffrées par un certificat :



Requêtes TLS (et donc chiffrées) :



75	219.105385	172.24.128.1	172.24.133.120	TLSv1.3	749 Application Data
76	219.106167	172.24.133.120	172.24.128.1	TLSv1.3	594 Application Data
77	219.149500	172.24.128.1	172.24.133.120	TCP	54 25896 → 443 [ACK] Seq=3300 Ack=2489 Win=64768 Len=0
78	224.111595	172.24.133.120	172.24.128.1	TLSv1.3	78 Application Data
79	224.111595	172.24.133.120	172.24.128.1	TCP	54 443 → 25896 [FIN, ACK] Seq=2513 Ack=3300 Win=74112 Len=0
80	224.111724	172.24.128.1	172.24.133.120	TCP	54 25896 → 443 [ACK] Seq=3300 Ack=2514 Win=64768 Len=0
81	224.112136	172.24.128.1	172.24.133.120	TLSv1.3	78 Application Data
82	224.112252	172.24.128.1	172.24.133.120	TCP	54 25896 → 443 [FIN, ACK] Seq=3324 Ack=2514 Win=64768 Len=0
83	224.112682	172.24.133.120	172.24.128.1	TCP	54 443 → 25896 [ACK] Seq=2514 Ack=3325 Win=74112 Len=0

- Où se trouve le certificat

Pour trouver l'emplacement du certificat, il suffit d'ouvrir le fichier default-ssl.conf dans sites-available et de regarder la ligne « SSLCertificateFile » pour avoir l'emplacement du certificat :

```
#   A self-signed (snakeoil) certificate can be created by installing
#   the ssl-cert package. See
#   /usr/share/doc/apache2/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key
```

- Le site fonctionne suivant deux modes. Supprimer le mode http
  a2dissite 000-default

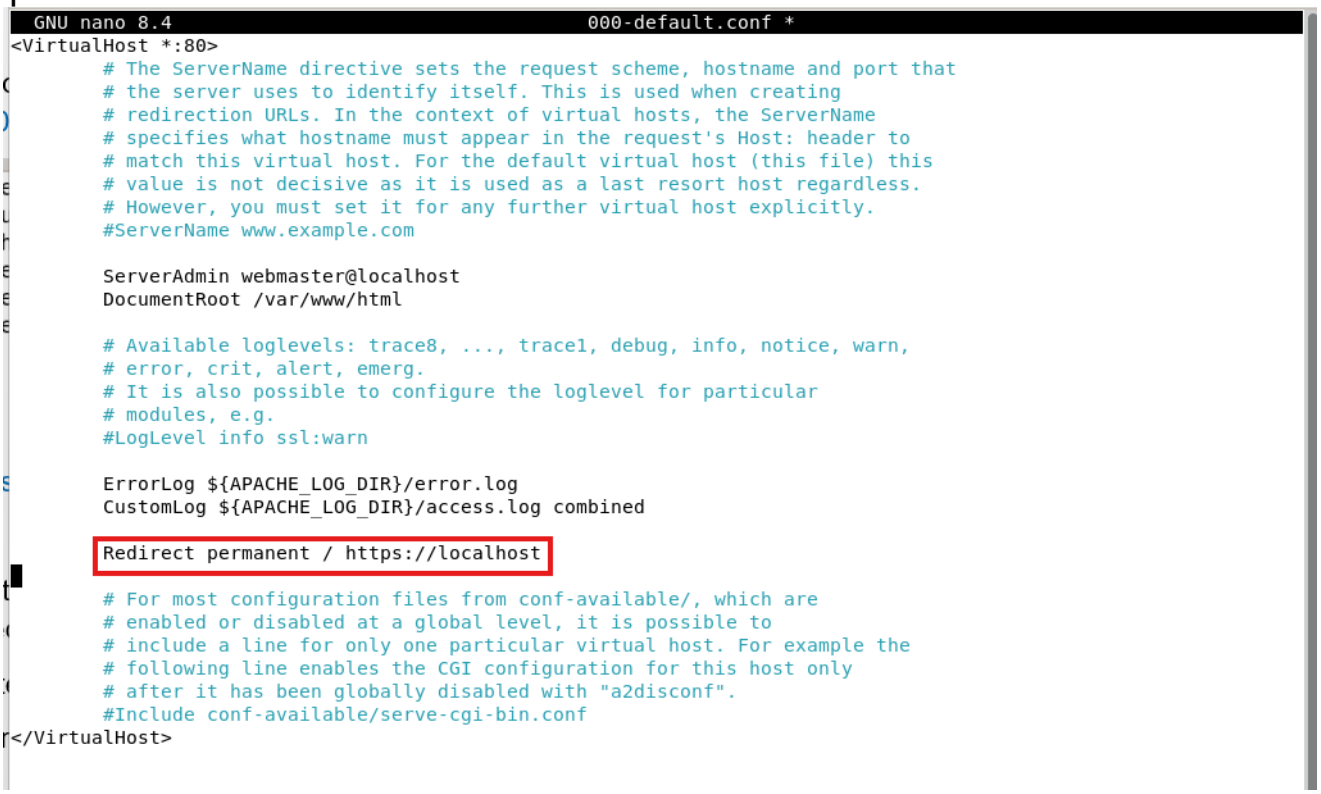
```
root@debian:/etc/apache2/sites-available# /usr/sbin/a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian:/etc/apache2/sites-available# systemctl restart apache2
root@debian:/etc/apache2/sites-available#
```

Le site a été désactivé avec succès

Si on veut que toutes les requêtes soient redirigées automatiquement en HTTPS, il faut faire une redirection permanente dans le fichier

/etc/apache2/sites-available/000-default.conf

Redirect permanent / https://localhost

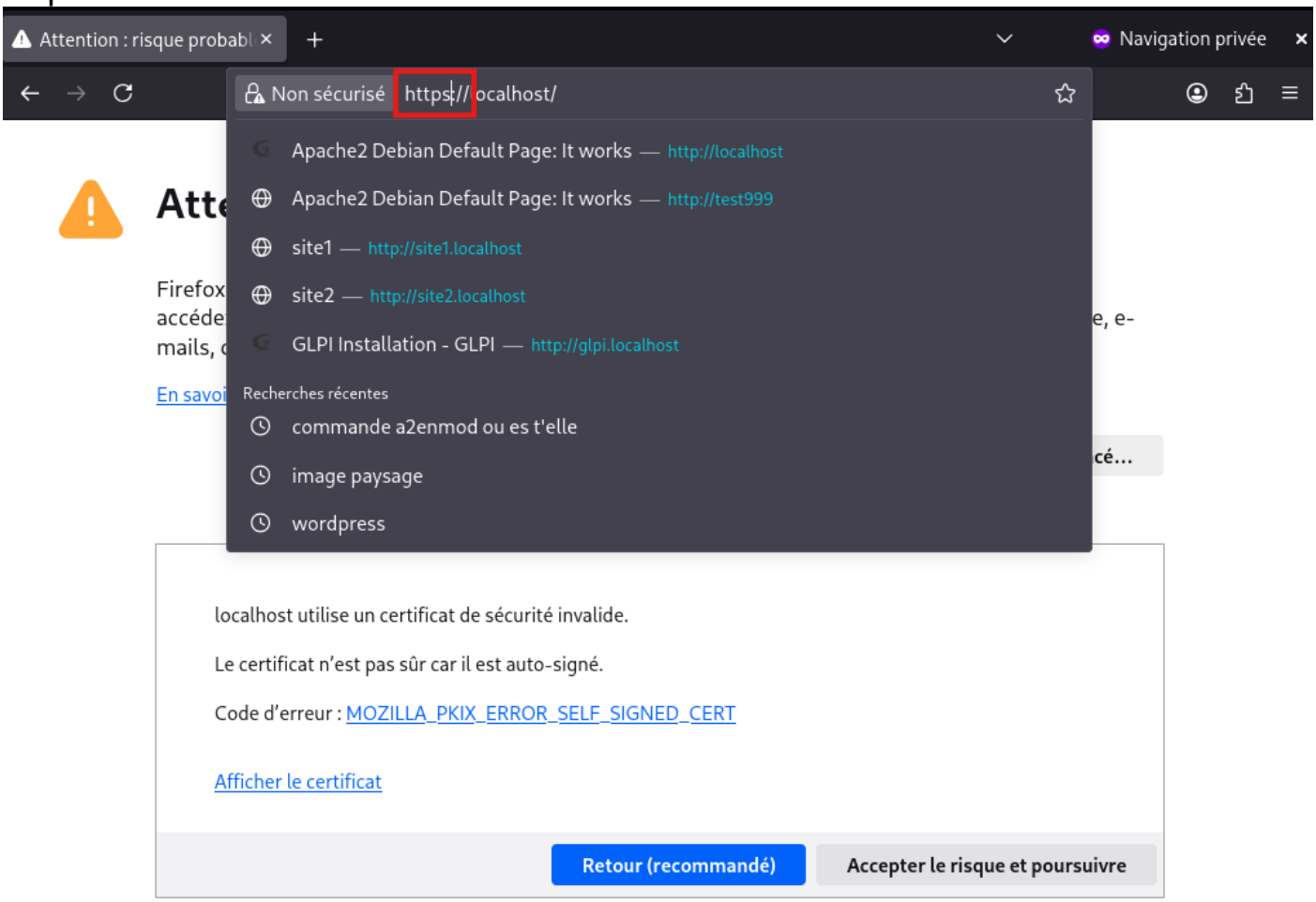


Remettre le site 000-default et tester

Activation du site avec la commande a2ensite et redémarrage du démon apache2 :

```
root@debian:/etc/apache2/sites-available# /usr/sbin/a2ensite 000-default
Enabling site 000-default.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian:/etc/apache2/sites-available# systemctl restart apache2
```

La redirection c’est fait avec succès vers https://localhost:



V/ Utilisation de MySQL

1) Introduction

MySql (Mariadb) est un SGBD (Système de gestion de base de données) libre, complémentaire de PHP pour créer des pages dynamiques : elles sont fabriquées à la demande.

Postgresql est un autre SGBD libre. Oracle ou Microsoft SQL Server sont propriétaire.

2) Installation

Déjà faite au début du TP (installation de LAMP avec APT) :

```
sudo apt update

sudo apt upgrade

sudo apt install apache2 php mariadb-server libapache2-mod-php
php-mysql
```

Vérification avec phpinfo :

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 8.4.11
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	Off	Off
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	/var/run/mysqld/mysqld.sock	/var/run/mysqld/mysqld.sock
mysqli.default_user	*no value*	*no value*
mysqli.local_infile_directory	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.rollback_on_cached_plink	Off	Off

mysqlnd

mysqlnd	enabled
Version	mysqlnd 8.4.11
Compression	supported
core SSL	supported
extended SSL	supported
Command buffer size	4096
Read buffer size	32768
Read timeout	86400
Collecting statistics	Yes
Collecting memory statistics	No
Tracing	n/a
Loaded plugins	mysqlnd,debug_trace,auth_plugin_mysql_native_password,auth_plugin_mysql_clear_password,auth_plugin_caching_sha2_password,auth_plugin_sha256_password
API Extensions	mysqli,pdo_mysql

Le status du serveur SQL est bien défini sur « enabled ».

3) Fonctionnement

Mysql gère les fichiers de la base de données, les fonctions de protection et de sécurité, et fournit des fonctions de programmation (comme **php**)

Le processus mysqld est le serveur qui a le droit d'interroger la base.

4) L'utilitaire MySQL

mysql -u root -p

```
root@debian:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 674
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/Maria
DB/Server

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)]>
```

C'est l'éditeur SQL. Commande utile : help

```
MariaDB [(none)]> help

General information about MariaDB can be found at
http://mariadb.org

List of all client commands:
Note that all text commands must be first on line and end with ';'
?         (\?) Synonym for `help'.
charset   (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
edit      (\e) Edit command with $EDITOR.
ego       (\G) Send command to MariaDB server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to MariaDB server.
help      (\h) Display this help.
nopager   (\n) Disable pager, print to stdout.
notee     (\t) Don't write into outfile.
nowarning (\w) Don't show warnings after every statement.
pager     (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
costs     (\Q) Toggle showing query costs after each query
rehash    (\#) Rebuild completion hash.
sandbox   (\-) Disallow commands that access the file system (except \P without an argument and \e).
source    (\.) Execute an SQL script file. Takes a file name as an argument.
status    (\s) Get status information from the server.
system    (\!) Execute a system shell command.
tee       (\T) Set outfile [to_outfile]. Append everything into given outfile.
use       (\u) Use another database. Takes database name as argument.
warnings  (\W) Show warnings after every statement.

For server side help, type 'help contents'

MariaDB [(none)]>
```

Nous allons créer la base Films et y donner tous les droits à l'utilisateur adminFilms :

create database Films;

grant all privileges on Films.* to adminFilms@localhost

identified by '1234';

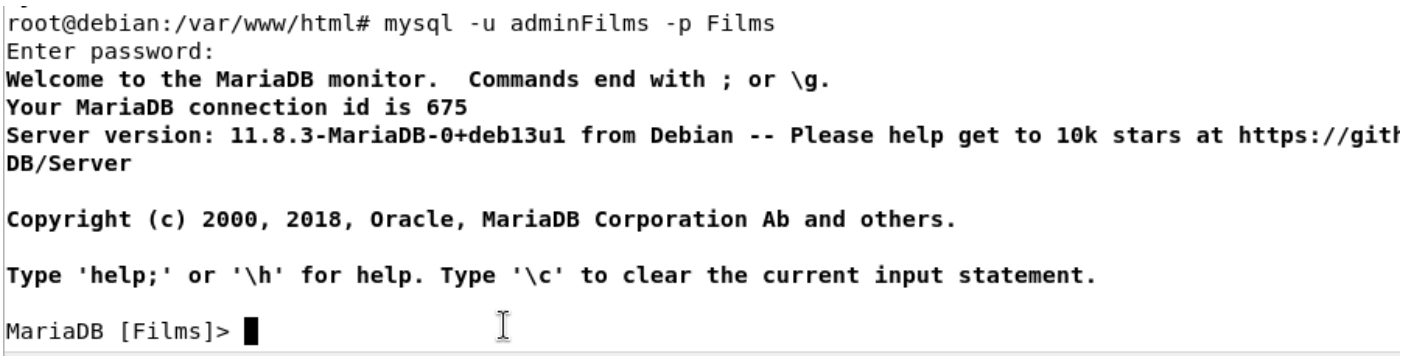
exit

Commandes executes avec succès sur notre serveur MySQL :



Maintenant, nous allons nous connecter à la base de données Films grâce à l'utilisateur adminFilms :

mysql -u adminFilms -p Films



Maintenant, nous allons créer la base FilmSimple avec comme valeur, le titre (variable avec 30 caractères aux maximum) et l'année du film (integer) :

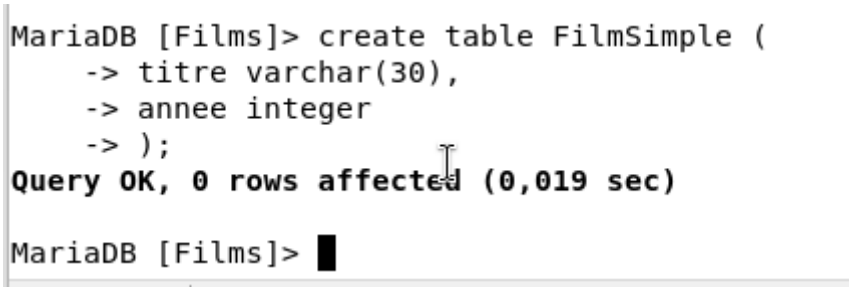
create table FilmSimple (

titre varchar(30),

annee integer

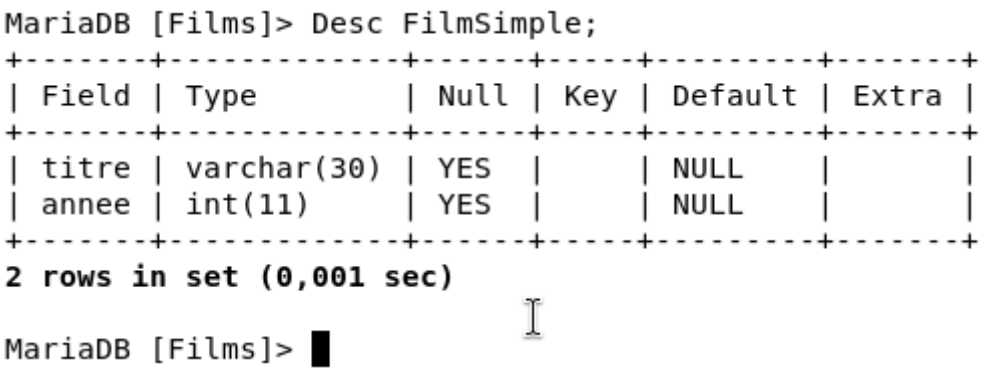
);

Résultat de la console MySQL :



Maintenant, nous allons vérifier la structure de notre table FilmSimple (nous y retrouvons les données précédemment entrées) :

Desc FilmSimple;



Maintenant, nous allons insérer différents films dans notre table FilmSimple, grâce à la commande insert :

insert into FilmSimple

(titre,annee)

Values

('Pulp Fiction', 1994),

('Alien', 1079),

('Titanic', 1997);

Résultat de la console :

```
MariaDB [Films]> insert into FilmSimple
    -> (titre,annee)
    -> Values
    -> ('Pulp Fiction', 1994),
    -> ('Alien', 1079),
    -> ('Titanic', 1997);
Query OK, 3 rows affected (0,004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [Films]>
```

Maintenant, nous allons interroger notre table, en demandant les films sortis en 1997, pour cela, nous allons utiliser la commande select, avec ce que l'on cherche, notre table, et ce que ou l'année est égale à 1997 :

select titre from FilmSimple where annee=1997;

Résultat de la console :

```
MariaDB [Films]> select titre from FilmSimple where annee = 1997
    -> ;
+---------+
| titre   |
+---------+
| Titanic |
+---------+
1 row in set (0,001 sec)

MariaDB [Films]>
```

Nous obtenons le film Titanic qui est bien sorti en 1997.

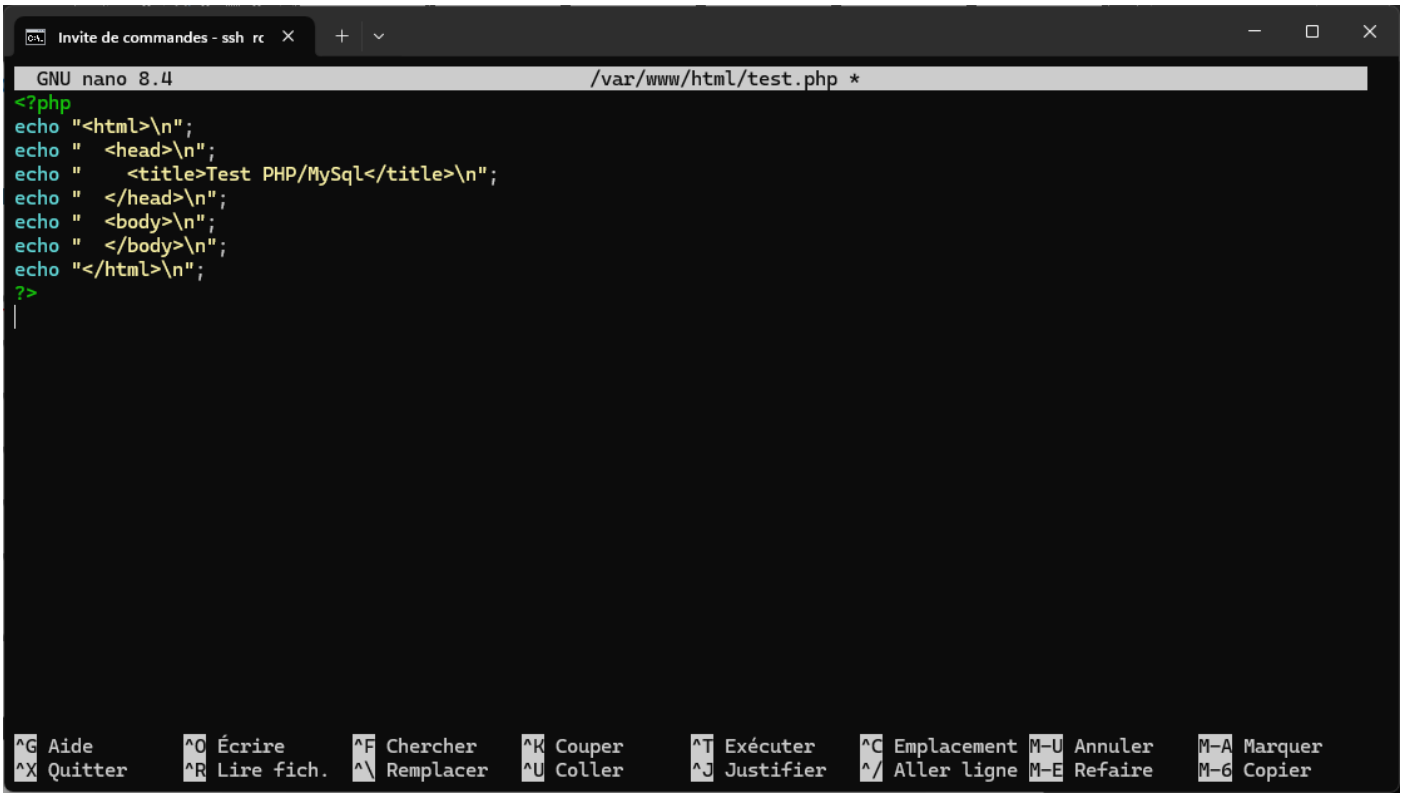
5) Lien avec PHP

Créer le fichier /var/www/html/test.php :

```
<?php
echo "<html>\n";
echo "  <head>\n";
echo "    <title>Test PHP/MySql</title>\n";
echo "  </head>\n";
echo "  <body>\n";
```

```
echo "  </body>\n";

echo "</html>\n";

?>
```

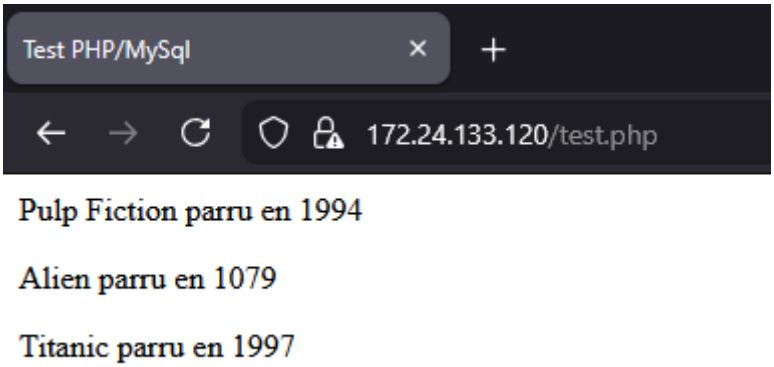
Création du fichier avec nano :



A présent, nous allons modifier le code pour que notre page affiche tous les films de la base de la table FilmSimple. Voici les modifications proposées au code pour que ce dernier ouvre une connexion avec la base de donnée Film, affiche les informations dans le navigateur et ferme la connexion avec la base :

```php
<?php
echo "<html>\n";
echo "  <head>\n";
echo "    <title>Test PHP/MySql</title>\n";
echo "  </head>\n";
echo "  <body>\n";
# Ouverture de la connexion avec le serveur MySQL
$connexion = mysqli_connect("localhost","adminFilms","1234","Films");

# Requête à envoyé au serveur MySQL (obtenir les films)
$requete = "select * from FilmSimple";
$resultat = mysqli_query($connexion, $requete);

# Pour toute les données dans la table FilmSimple, afficher ces dernières
# avec le titre du film, suivi de "parru e" avec l'année
while ($film = mysqli_fetch_assoc($resultat)) {
    echo "<p>" . $film['titre'] . " parru en " . $film['annee'] . "</p>\n";
}

# Fermer la connexion avec la base de donnée
mysqli_close($connexion);
echo "  </body>\n";
echo "</html>\n";
?>
```

Vérification dans le navigateur :



La connexion entre le PHP et la base de données a bien été effectuée !

6) Administration MySQL

- Gestion des droits dans la console Mysql

Ajoutez un nouvel utilisateur qui a tous les droits sur la base Films :

```
# Commande pour créer l'utilisateur

CREATE USER 'nouvel_utilisateur'@'%' IDENTIFIED BY
'motdepasse';

#Donner tous les droits au nouvel utilisateur

GRANT ALL PRIVILEGES ON *.* TO 'nouvel_utilisateur'@'%' WITH
GRANT OPTION;

# Recharger les permissions

FLUSH PRIVILEGES;
```

Résultat de la console (l'utilisateur a bien été créer) :

```
MariaDB [(none)]> CREATE USER 'nouvel_utilisateur'@'%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0,007 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'nouvel_utilisateur'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]>
```

Enlevez-lui les droits d'écriture :

Nous pouvons utiliser la commande REVOKE pour supprimer une permission, nous allons dans notre cas, supprimer la permission INSERT, UPDATE et DELETE pour plus que l'utilisateur puisse écrire :

```
REVOKE INSERT, UPDATE, DELETE ON *.* FROM
'nouvel_utilisateur'@'%';
```

Résultat de la console :

```
MariaDB [(none)]> REVOKE INSERT, UPDATE, DELETE ON *.* FROM 'nouvel_utilisateur'@'%';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]>
```

Maintenant, nous allons utiliser cet utilisateur pour nous connecter à la table Films :

```
root@ytest998:~# mysql -u nouvel_utilisateur -p Films
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 683
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k
r

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Une fois ceci fait, nous allons voir si nos permissions ont bien été appliquées (lecture possible mais écriture bloquée). Pour cela, nous allons premièrement faire un **select * from FilmSimple;** pour vérifier si nous pouvons lire les données de la table FilmSimple :

```
MariaDB [Films]> select * from FilmSimple
    -> ;
+-------------+-------+
| titre       | annee |
+-------------+-------+
| Pulp Fiction |  1994 |
| Alien       |  1079 |
| Titanic     |  1997 |
+-------------+-------+
3 rows in set (0,001 sec)
```

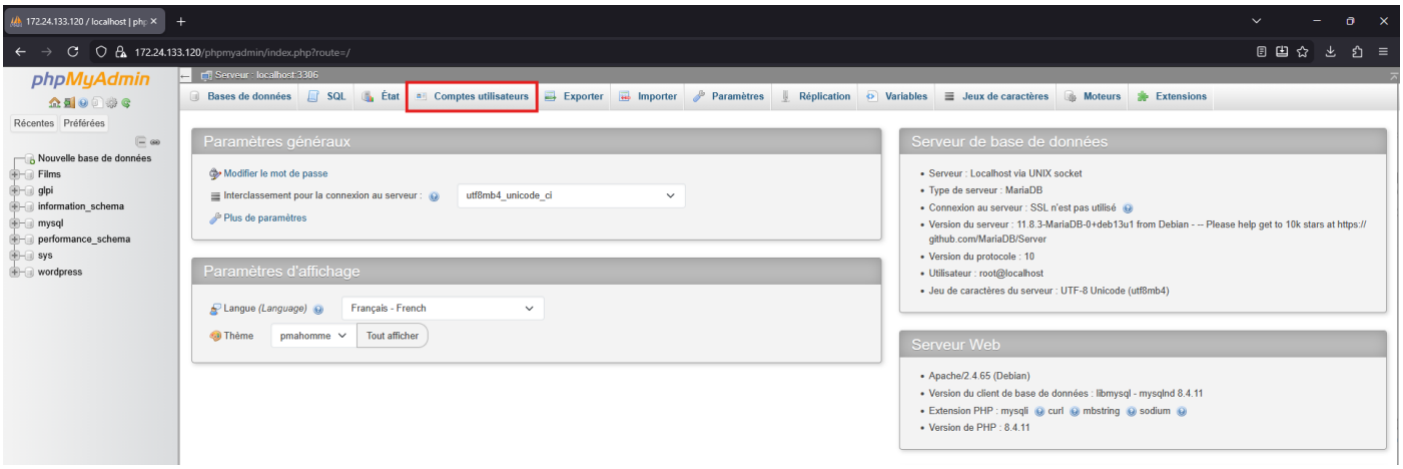
Comme nous l'observons, nous pouvons lire les données. Maintenant, nous allons essayer d'ajouter des données dans cette table :

```
MariaDB [Films]> insert into FilmSimple
    -> (titre,annee)
    -> Values
    -> ('Test', 1999);
ERROR 1142 (42000): INSERT command denied to user 'nouvel_utilisateur'@'localhost' for table `Films`.`FilmSimple`
MariaDB [Films]>
```

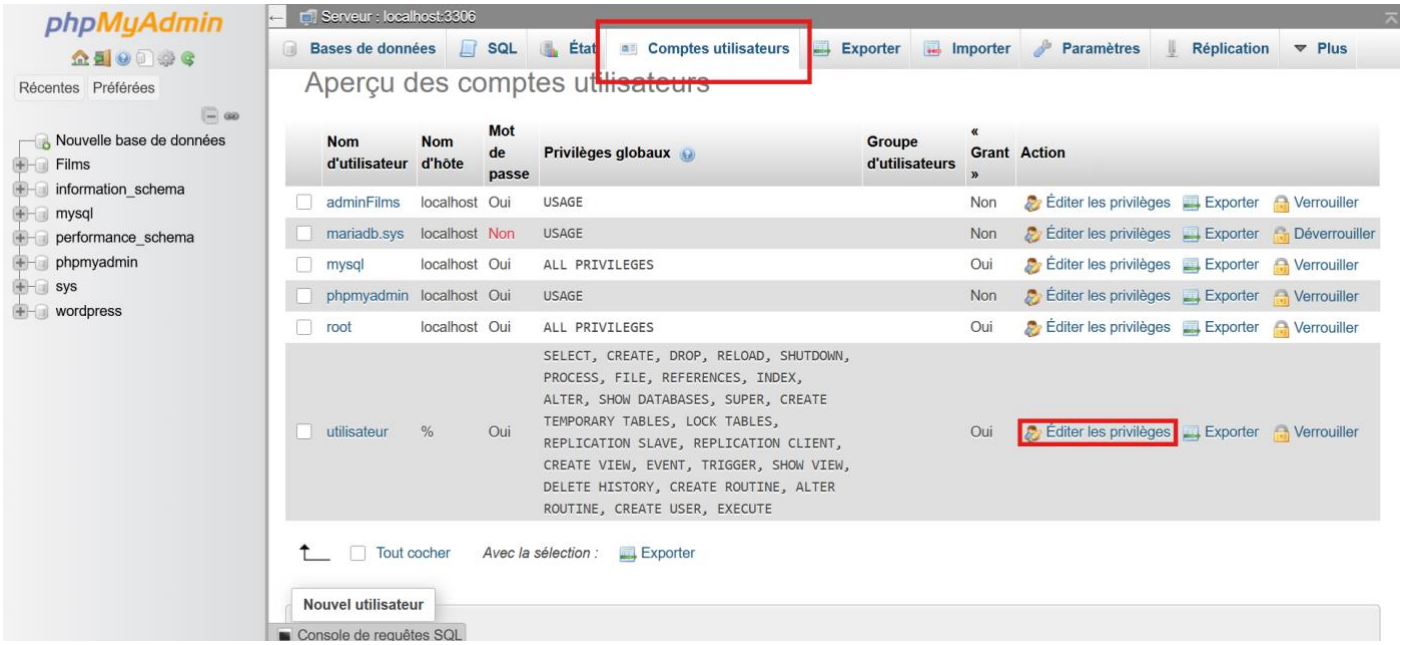
On remarque que nous avons bien une erreur « INSERT command denied » qui signifie que l'utilisateur n'as pas le droit d'écrire sur le serveur MySQL.

- Dans PhpMyAdmin, où se trouve la table des utilisateurs ?

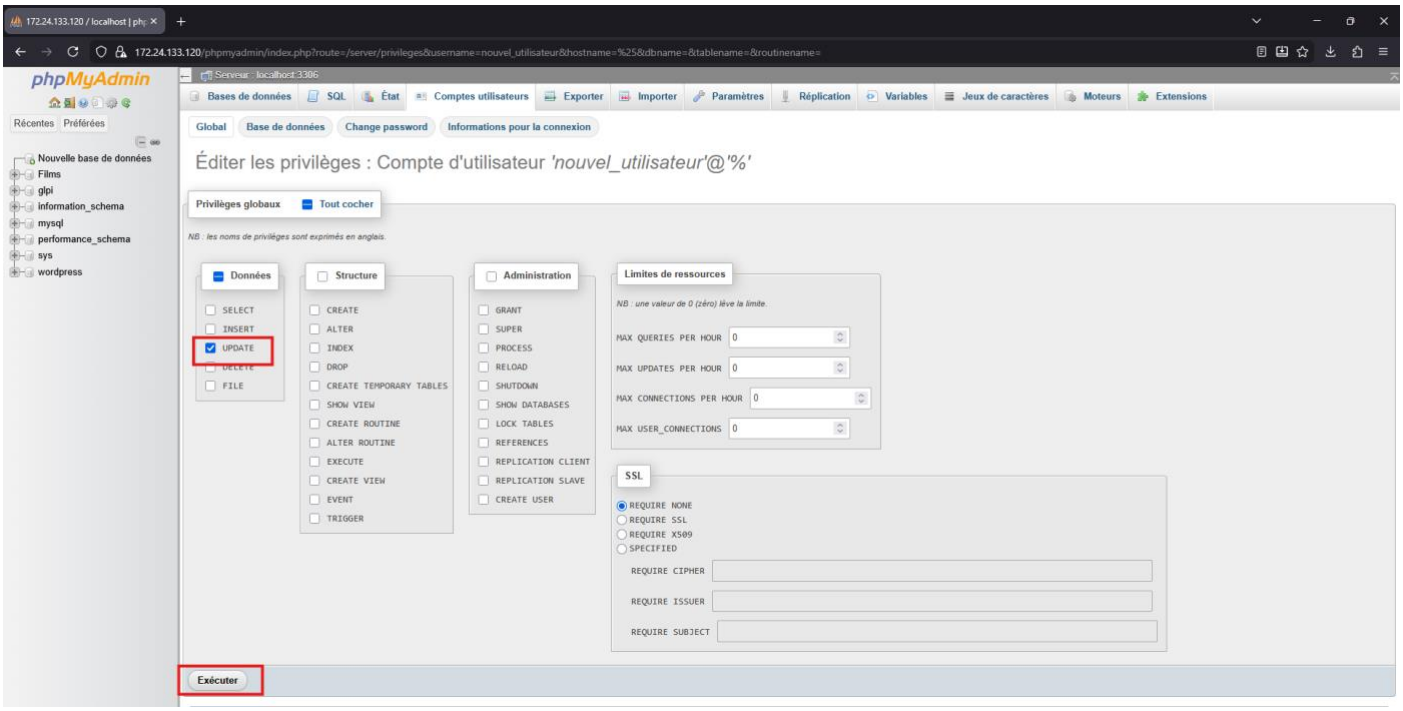
Nous allons nous connecter à PhpMyAdmin avec l'utilisateur root afin d'avoir accès à toute les base de données et tables. Les comptes utilisateurs sont affichées dans le volet compte utilisateurs sur la page d'accueil de PhpMyAdmin :



- Modifier les droits de votre utilisateur pour uniquement mise à jour



Nous avons une vue sur les permissions de l'utilisateur, dans notre cas, nous allons uniquement cocher « Update » (signifiant mise à jour) et exécuter la requête :



Une fois ceci fait, nous avons un message nous indiquant que les modifications ont bien été effectuées (avec la commande exécutée) :



- Créer un nouvel utilisateur avec uniquement les droits de sélection

Vidéo de démonstration de la création d'un utilisateur via PhpMyAdmin avec seulement les droits de sélection :
https://drive.proton.me/urls/2WJ3JGCGJ4#5M6QxSIjnxfK

- Utiliser la commande mysqladmin pour voir l'état du serveur (status)

Utilisation de la commande suivante (pour faire un status en root) : **mysqladmin -u root -p status**

```
root@ytest998:~# mysqladmin -u root -p status
Enter password:
Uptime: 92882  Threads: 1  Questions: 13485  Slow queries: 0  Opens: 87  Open tables: 80  Queries per second avg: 0.145
root@ytest998:~#
```

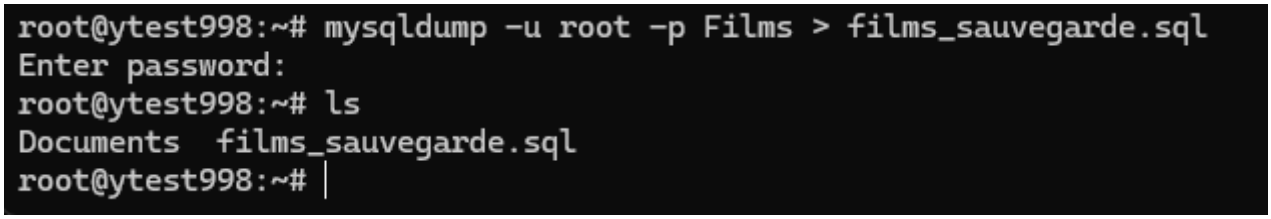
La commande nous affiche diverses informations tel que le temps de démarrage du serveur, le nombre de questions postées au serveur, le nombre de tables, la moyenne de temps de réaction du serveur par requêtes.

- Utiliser la commande mysqldump pour sauvegarder votre base Films dans un fichier Sql.

Pour sauvegarder notre base Films, nous allons utiliser la commande suivante, qui comme toujours, va se connecter à l'utilisateur root et exporter la table dans un fichier .sql :

```
mysqldump -u root -p Films > films_sauvegarde.sql
```

Résultat de la console :



Que comprenez-vous de ce fichier ?

Aperçu du fichier SQL :

```sql
/*M!999999\- enable the sandbox mode */
-- MariaDB dump 10.19-11.8.3-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: Films
-- ------------------------------------------------------
-- Server version       11.8.3-MariaDB-0+deb13u1 from Debian

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*M!100616 SET @OLD_NOTE_VERBOSITY=@@NOTE_VERBOSITY, NOTE_VERBOSITY=0 */;

--
-- Table structure for table `FilmSimple`
--

DROP TABLE IF EXISTS `FilmSimple`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8mb4 */;
CREATE TABLE `FilmSimple` (
  `titre` varchar(30) DEFAULT NULL,
  `annee` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_uca1400_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `FilmSimple`
--

LOCK TABLES `FilmSimple` WRITE;
/*!40000 ALTER TABLE `FilmSimple` DISABLE KEYS */;
set autocommit=0;
INSERT INTO `FilmSimple` VALUES
('Pulp Fiction',1994),
('Alien',1079),
('Titanic',1997);
/*!40000 ALTER TABLE `FilmSimple` ENABLE KEYS */;
UNLOCK TABLES;
commit;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*M!100616 SET NOTE_VERBOSITY=@OLD_NOTE_VERBOSITY */;

-- Dump completed on 2025-12-14 14:48:33
```

Dans ce fichier, on observe des commandes que nous avons précédemment indiqué dans notre console MySQL, comme les valeurs insérées dans la table FilmSimple, ou bien même tout simplement la table avec les valeurs qu'elle doit contenir (comme défini précédemment).

Ce fichier n'est simplement que les commandes que nous avons indiquées précédemment, suivi de quelques commentaires auto généré par MySQL.

Supprimer votre base Films. (Preuve)

Nous allons nous reconnecter à la console MySQL (**mysql -u root -p**)

Puis, nous allons supprimer la base Films (**DROP DATABASE Films;)**

```
root@ytest998:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 788
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

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)]> DROP DATABASE Films;
Query OK, 1 row affected (0,016 sec)

MariaDB [(none)]>
```

Maintenant, nous allons vérifier les tables (**SHOW DATABASES;**)

```
MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| glpi               |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
6 rows in set (0,001 sec)

MariaDB [(none)]>
```

La table Film a bien été supprimée.

Importer le fichier de sauvegarde

Nous allons recréer la table « Films » grâce à la commande suivante :

```
mysql -u root -p -e "CREATE DATABASE Films;"
```

```
root@ytest998:~# mysql -u root -p -e "CREATE DATABASE Films;"
Enter password:
root@ytest998:~#
```

Maintenant, nous allons importer le fichier dans cette table grâce à la commande suivante :

```
mysql -u root -p Films < films_sauvegarde.sql
```

```
root@ytest998:~# mysql -u root -p Films < films_sauvegarde.sql
Enter password:
root@ytest998:~#
```

Maintenant, les données ont été restaurées sur la base Films, on peut tenter de s'y connecter et de faire un select pour voir si les données ont bien été restaurées :

```
root@ytest998:~# mysql -u root -p Films
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 800
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

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 [Films]> select * from FilmSimple;
+--------------+-------+
| titre        | annee |
+--------------+-------+
| Pulp Fiction |  1994 |
| Alien        |  1079 |
| Titanic      |  1997 |
+--------------+-------+
3 rows in set (0,001 sec)

MariaDB [Films]>
```

Les données ont bien été restaurées !