TP LAMP

Table des matières

Introduction et installation

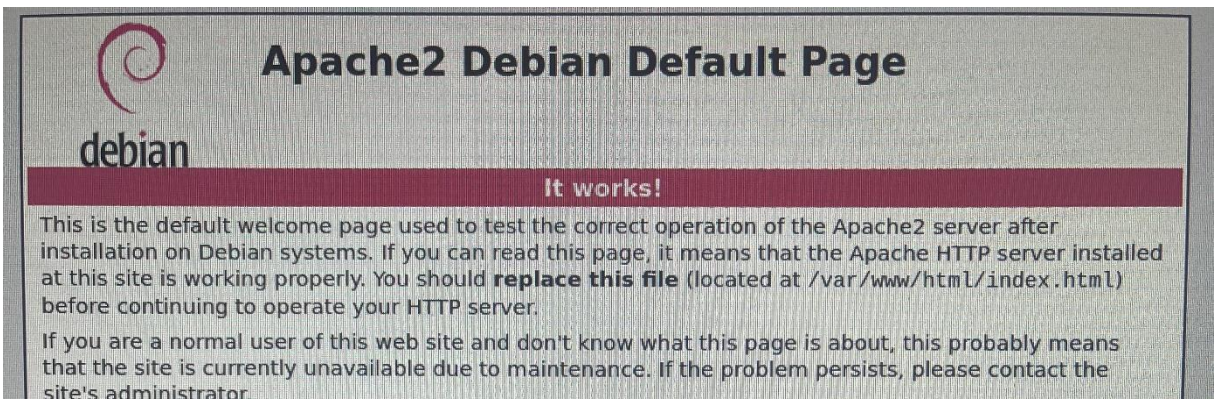
Acronyme LAMP :

- Linux : le système d'exploitation
- Apache : le serveur http
- MySQL : le serveur de base de données (Maria DB)
- PHP : le langage de script

Installation : apt install apache2 , apt install php, apt install mariadb-server, apt install libapache2-mod-php, apt install php-mysql

Lancement : service apache2 start

Test : http://localhost It Works!!!!



Test

Base de données : service mysqld start

Test console: mysql -u root -p

Test console: mdp

Test console: exit

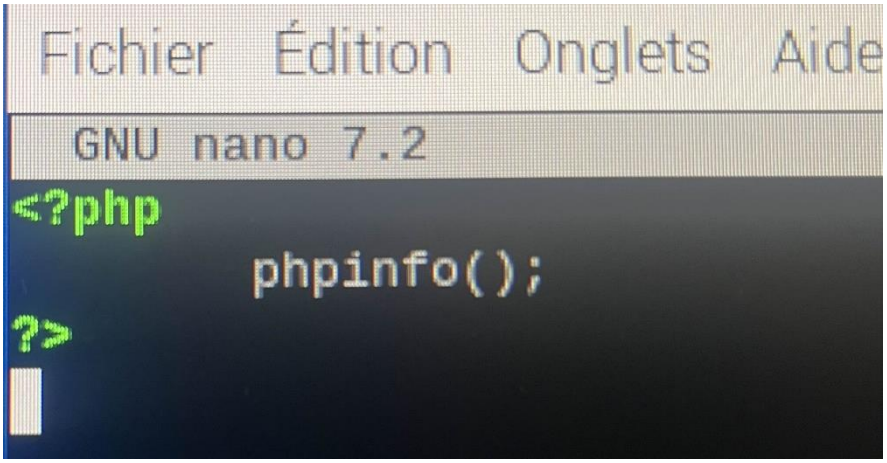
Test php: cd /var/www/html

Test php: nano phpinfo.php

Test php: <? php

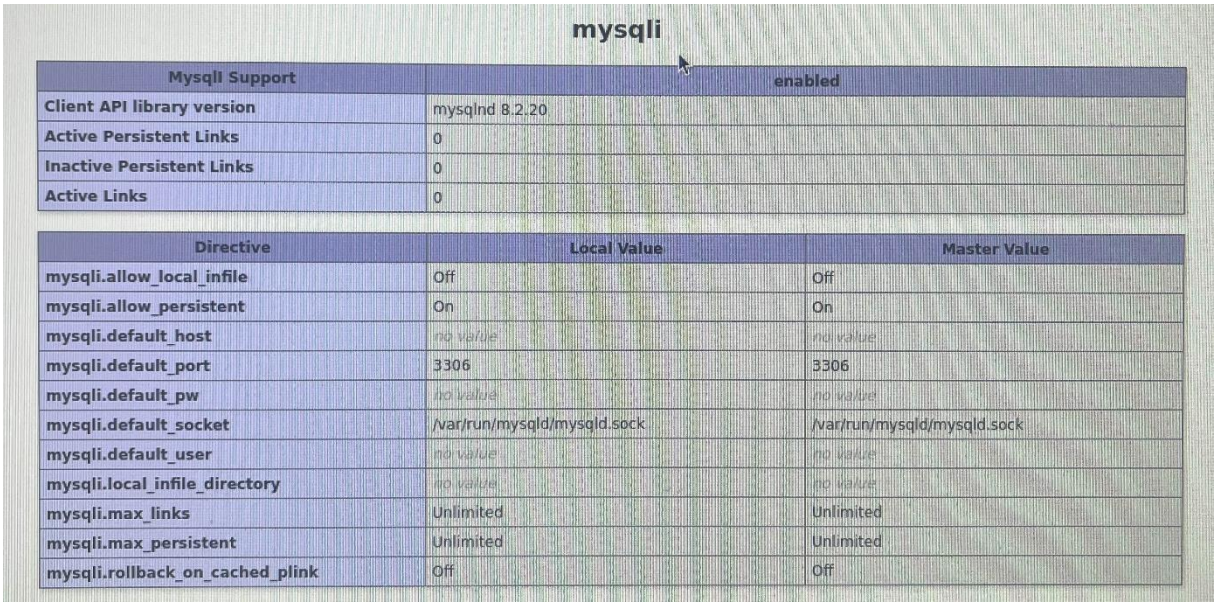
Phpinfo();

Test php: ?>



Test php: http://localhost/phpinfo.php

Le module Mysql est il installé ?



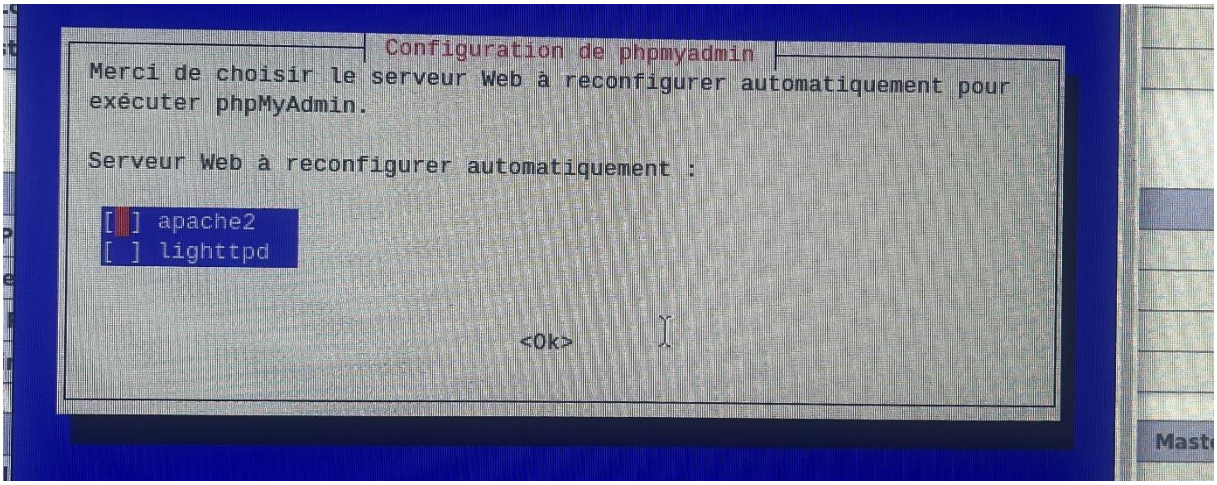
mysqli

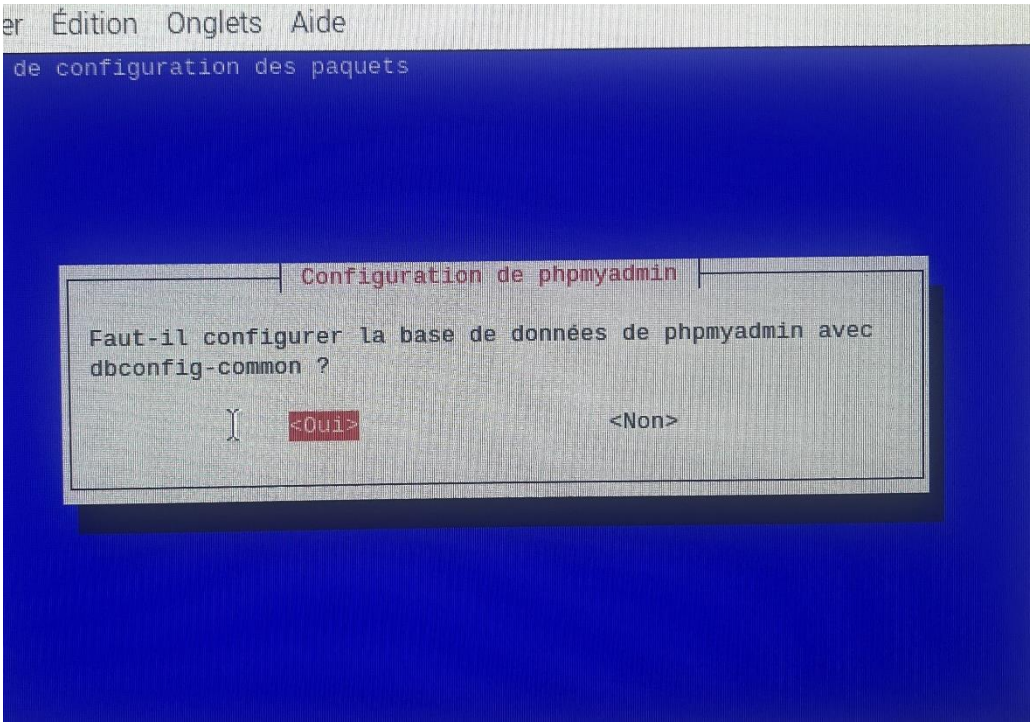
Mysqli Support	enabled
Client API library version	mysqlnd 8.2.20
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

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

Utilisation

1) On peut installer phpmyadmin pour administrer graphiquement les bases de données. apt install phpmyadmin

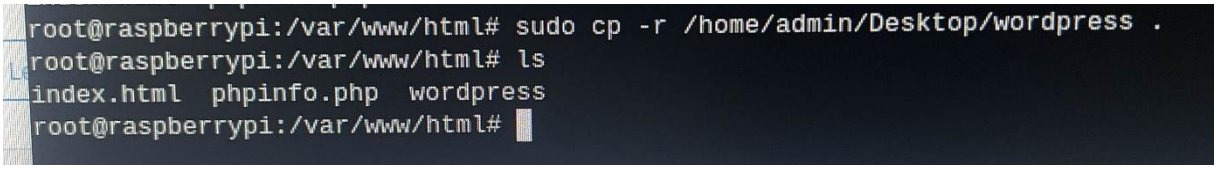




Test : http://localhost/phpmyadmin

2) On installe Wordpress :

- Télécharger l'archive
- La décompresser dans le dossier /var/www/html



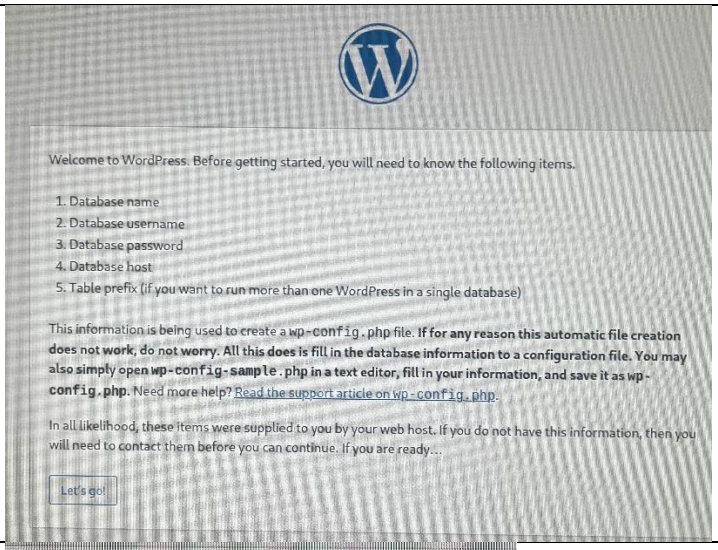
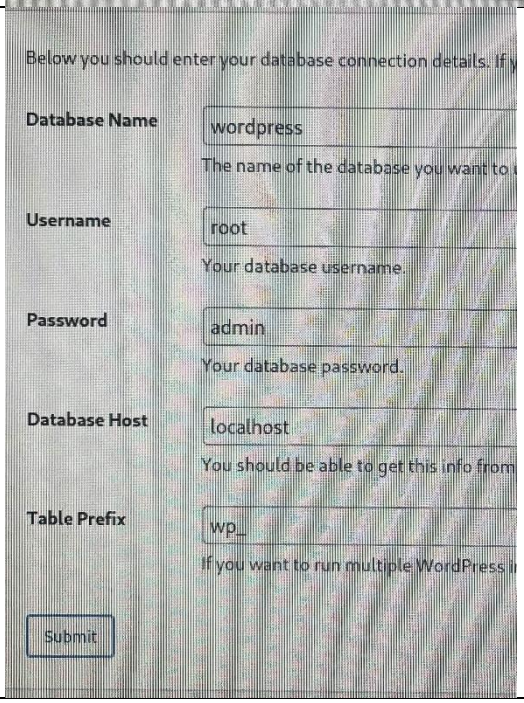
-

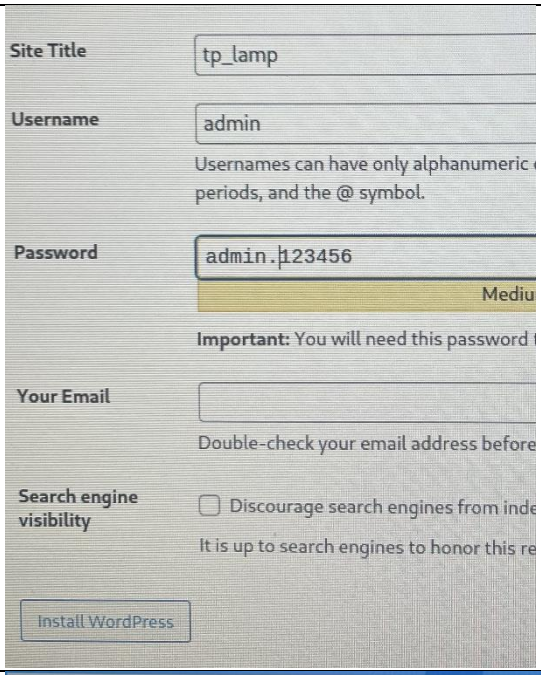
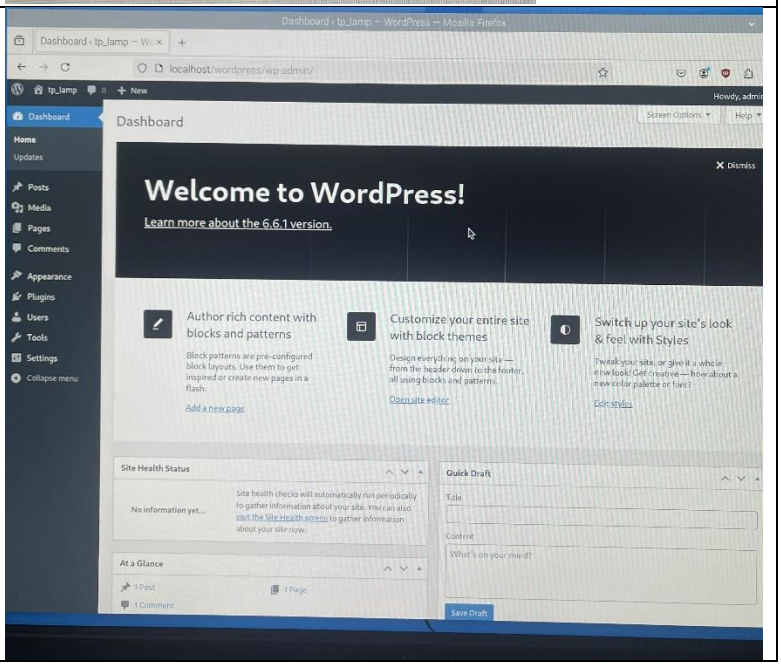
- http://localhost/wordpress

Avant de continue il faut bien penser à créer une base de données pour pouvoir continuer.

- mysql -u root -p
- mdp
- CREATE DATABASE wordpress; #penser a mettre le “;”
- USE wordpress;

Installation WorPress

1 – Page d’acceuil wordpresse http://localhost/wordpress	
2 – Renseigner le nom de la base de donnée que nous avons crée en amont, root , le mdp.	

3 – Renseigner le nom de votre site , un user et un mot de passe, il faut mettre une adresse mail sinon vous ne pourrez pas continuer.	
4 – Connectez vous, voici la page de gestion de votre site.	

Apache

1) Historique

Internet : Interconnexion mondiale des réseaux, créé dans les années 60 au ministère de la défense américain.

World Wide Web (WWW) : système hypertexte constitué par l'ensemble des pages reliées par le protocole http, créer en 1989 au CERN à Genève .
Mosaic : premier navigateur Web graphique en 1993, renommé Netscape Navigator en 1994, puis Mozilla Firefox en 2004. De son coté, Microsoft développe son Internet Explorer à partir de 1995.

2) Le protocole http

Deuxième et dernière version en 2015.

Déroulement d'une connexion :

- Connexion du client au serveur,
- Envoi par le client d'une requête GET,
- Réponse du serveur,
- Envoi par le client d'une requête de fermeture,
- Réponse du serveur,
- Fermeture de la connexion

Requêtes courantes :

- GET : récupération d'un document
- HEAD : récupération des en-têtes seulement
- POST : envoie des données au serveur
- PUT : envoi d'un fichier
- DELETE : suppression d'un fichier
- CONNECT : accès serveur sécurisé HTTPS

En-têtes client (requête) :

- host : nom du site recherché,
- référer : nom du site qui nous a renvoyé,
- user-agent : votre navigateur,
- accept : format de fichier accepté,
- accept-language : langue accepté.

En-têtes serveur (réponse) :

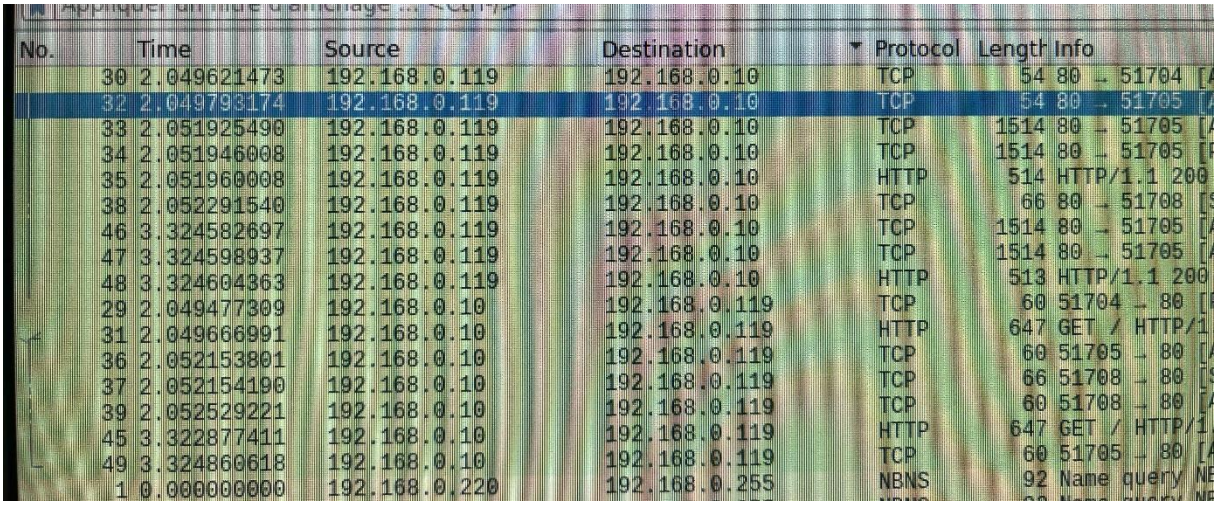
- date : du serveur
- server : logiciel du serveur (Apache, IIS, Nginx)
- content-type : format du fichier envoyé

- content-length : taille du fichier

Code réponses du serveur en 5 catégories :

- 100 à 199 : message informatif
- 200 à 299 : succès de la requête
- 300 à 399 : redirection
- 400 à 499 : erreur
- 500 à 599 : erreur interne serveur

TEST avec WireShark



No.	Time	Source	Destination	Protocol	Length	Info
30	2.049621473	192.168.0.119	192.168.0.10	TCP	54	80 → 51704 [A
32	2.049793174	192.168.0.119	192.168.0.10	TCP	54	80 → 51705 [A
33	2.051925490	192.168.0.119	192.168.0.10	TCP	1514	80 → 51705 [A
34	2.051946008	192.168.0.119	192.168.0.10	TCP	1514	80 → 51705 [P
35	2.051960008	192.168.0.119	192.168.0.10	HTTP	514	HTTP/1.1 200
38	2.052291540	192.168.0.119	192.168.0.10	TCP	66	80 → 51708 [S
46	3.324582697	192.168.0.119	192.168.0.10	TCP	1514	80 → 51705 [A
47	3.324598937	192.168.0.119	192.168.0.10	TCP	1514	80 → 51705 [A
48	3.324604363	192.168.0.119	192.168.0.10	HTTP	513	HTTP/1.1 200
29	2.049477309	192.168.0.10	192.168.0.119	TCP	60	51704 → 80 [F
31	2.049666991	192.168.0.10	192.168.0.119	HTTP	647	GET / HTTP/1
36	2.052153801	192.168.0.10	192.168.0.119	TCP	60	51705 → 80 [A
37	2.052154190	192.168.0.10	192.168.0.119	TCP	66	51708 → 80 [S
39	2.052529221	192.168.0.10	192.168.0.119	TCP	60	51708 → 80 [A
45	3.322877411	192.168.0.10	192.168.0.119	HTTP	647	GET / HTTP/1
49	3.324860618	192.168.0.10	192.168.0.119	TCP	60	51705 → 80 [A
1	0.000000000	192.168.0.220	192.168.0.255	NBNS	92	Name query NB

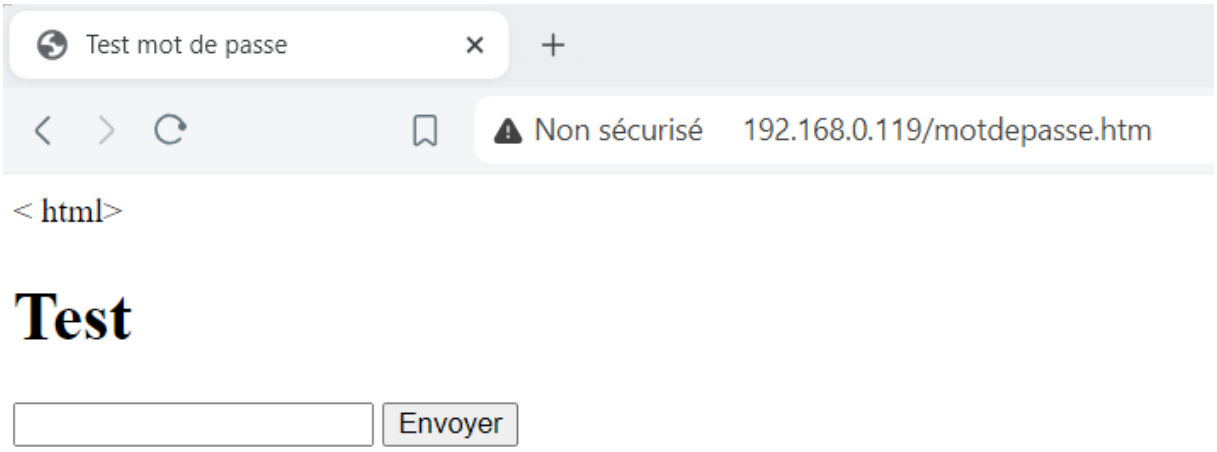
Nous pouvons voir les échanges entre la Raspberry Pi (192.168.0.119) et le Poste (192.168.0.10).

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

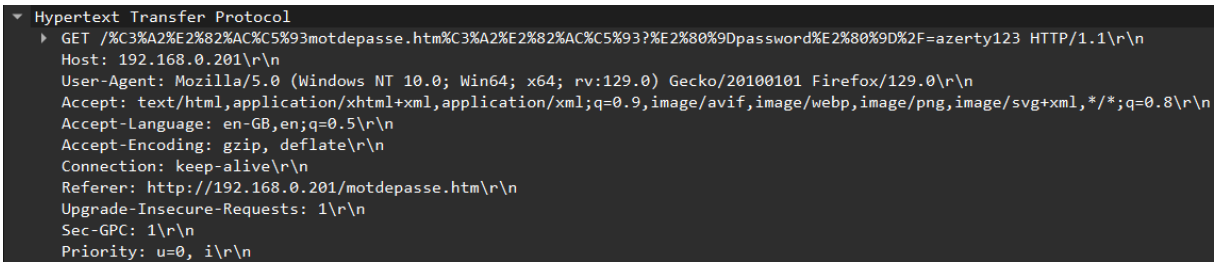
```
< html>
    <head>
        <title> Test mot de passe </title >
    <head>
    <body>
        <h1> Test </h1>
        <form action = ”motdepasse.htm” method =”post”>
            <input type = ”text” id=”passe” name=”password” />
            <button type=”submit”> Envoyer </button>
        </form>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

Capture de mot de passe avec Wireshark

Depuis le navigateur se rendre sur l'ip de la Rspberry pi /motdepasse.htm



Depuis Wireshark il faut intercepter la requête



Et vous verrez le mot de passe ! !!! ! !

La fondation Apache :

- ONG créée en 1995 pour développer des logiciels libres.
- Des centaines de bénévoles sur une centaine de projets.

- Licences Apache : à la différence de la GNU-GPL, on n'est pas obligé de republier sous licence Apache.

Le projet HTTP Apache :

- Version 2.4 en 2012
- Conception modulaire : mod_php, mod-ssl, etc…
  ➔ Phpinfo.php

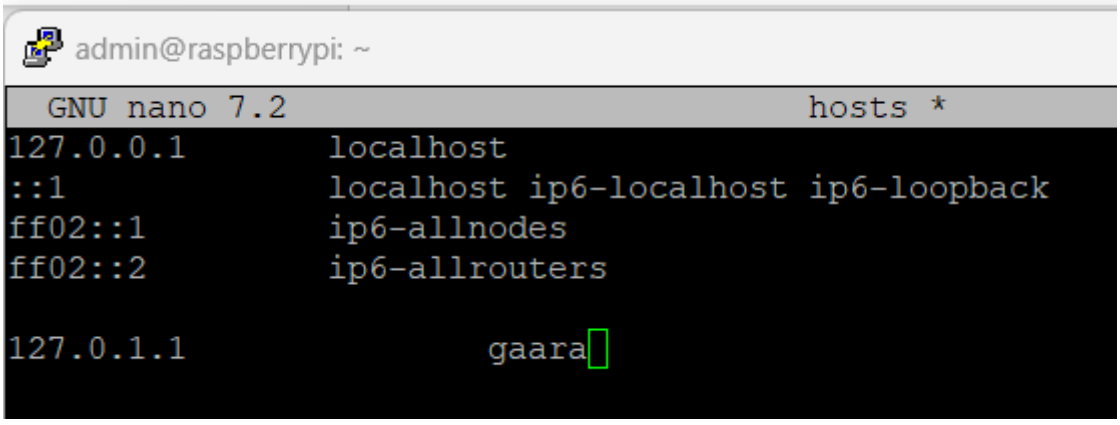
3) Configuration de base

- Démarrage
  - Démon apache2 ou httpd
  - Nom de la machine : localhost
  - Commande hostname

```
root@raspberrypi:/home/admin# hostname
raspberrypi
root@raspberrypi:/home/admin#
```

  - cd /etc/ nano hosts

  Déclarer un nom.




```
root@raspberrypi:/etc# nano hosts
root@raspberrypi:/etc# hostname
gaara
```

- Fichier de configuration : /etc/apache2/apache2.conf

  Questions :

  - Quel est le port par défaut ? Quelle directive ? 80 listen

- Y a-t-il d'autres fichiers de configuration ? Ou ? /etc/apache2
- Directive pour charger un module ? load mod_enable
- Sous quelle identité (user/group) fonctionne le serveur ? root apache
- Quel est le répertoire racine ? /var/www/html
- Quelle page est renvoyée si l'adresse est un dossier ? index.htm
- Comment sont traitées les adresses http://serveur/cgi-bin/toto? Fichier de commande shell
- Ou sont les fichiers de logs ? /var/log/apache2

Accéder d'une autre autre machine et vérifier access.log

- Accès au dossiers

  Créer un dossier webftp dans /var/www/html

  Copier dedans le fichier motdepasse.htm et accéder

  Supprimer motdepasse.htm de l'URL . Que se passe t'il ?

  Renommer motdepasse.htm en index.htm et actualiser. Pourquoi ?

  Renommer index.htm en test.htm

  Dans le fichier apache2.conf

  <Directory /var/www/html/webftp>

  Options -/+Indexes Redémarrer Apache

  </Directory>

  Tester avec http://localhost/webftp

4) Script CGI simple

   Dans votre dossier personnel, créer un fichier date.cgi :

   # !/bin/bash
   Echo "Content-type : text/html" "
   Echo ""
   Date + "%A %d %B %Y %T"

   Donner les droits d'exécution et tester

```
 GNU nano 7.2                                    date.cgi
#!/bin/bash
echo "content-type : text/html"
echo ""
date +"%A %d %B %Y %T"
```

Maintenant il faut donner les droits d'exécution

```
root@raspberrypi:/home/admin# chmod +x date.cgi
root@raspberrypi:/home/admin#
```

Il faut l'exécuter

```
root@raspberrypi:/home/admin# ./date.cgi
content-type : text/html

lundi 02 septembre 2024 15:21:53
```

- Déplacer le script dans /var/www/html/cgi-bin

```
root@raspberrypi:/var/www/html/cgi-bin# ls
date.cgi
root@raspberrypi:/var/www/html/cgi-bin#
```

- Modifier le fichier de conf :

< Directory "/usr/lib/cgi-bin">

AllowOverride None

Options +ExecCGI

AddHandler cgi-script .cgi

Require all granted

</Directory>

- Deuxième exemple : listeproc.cgi

#!/bin/bash
Echo “content-type : text/plain”
Echo
User =$(echo “$QUERY_STRING” | grep -oP ‘(?<= user =) \w+’)
Ps -fu $user

```
 GNU nano 7.2                                  listeproc.cgi *
#!/bin/bash
echo "content-type : text/plain"
echo
        user =$(echo "$QUERY_STRING" | grep -oP '(?<= user =) \w+')
        ps -fu $user
```

Il faut lui donner les droit d’exécution.

```
root@raspberrypi:/var/www/html/cgi-bin# chmod +x listeproc.cgi
root@raspberrypi:/var/www/html/cgi-bin# ./listeproc.cgi
content-type : text/plain

./listeproc.cgi: ligne 4: user : commande introuvable
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1850  0.0  0.0  10824  4224 pts/0    S+   16:38   0:00 sudo su
root        1851  0.0  0.0  10824  1700 pts/1    Ss   16:38   0:00  \_ sudo su
root        1852  0.0  0.0   9472  3456 pts/1    S    16:38   0:00      \_ su
root        1853  0.0  0.0   7064  3584 pts/1    S    16:38   0:00          \_ bash
root        1940  0.0  0.0   6800  3072 pts/1    S+   16:57   0:00              \_ /bin/bash
root        1945  250  0.0  11408  4352 pts/1    R+   16:57   0:00                  \_ ps -fu
root        1754  0.0  0.0   7496  3072 tty1     Ss   16:34   0:00 /bin/login -f
root@raspberrypi:/var/www/html/cgi-bin#
```

Depuis le navigateur si vous souhaiter accéder à listeproc.cgi il faut bien que le module soit actif, pour vérifier il faut entrer cette commande :

```
root@raspberrypi:/var/www/html/cgi-bin# sudo a2enmod cgi
```

5) Protection des accès

- Créer un dossier secret
  Configurer Apache pour que ce dossier soit accessible uniquement à partir du serveur, et pas d’un client. Test

  <Directory /var/www/html/secret>
  Deny from all
  Allow from xxx.xxx.xxx.xxx (adresse ip de la machine)
  </Directory>

```
<Directory /var/www/html/secret>
        deny from all
        allow from 192.168.0.119
</Directory>
```

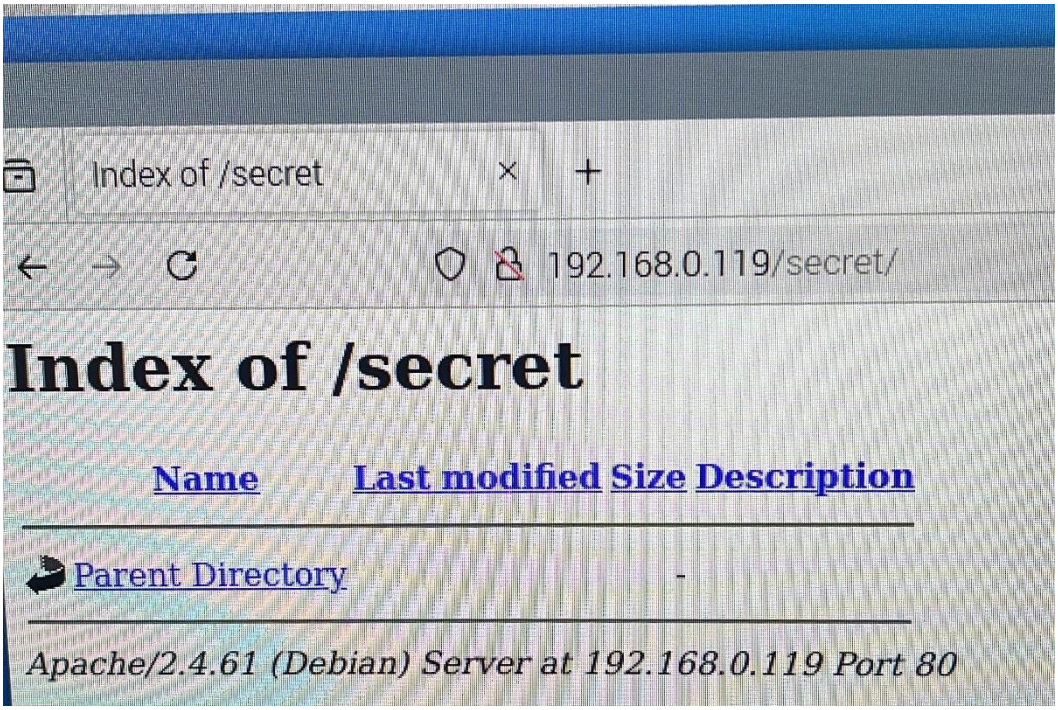
Screen depuis le client.

Forbidden

You don't have permission to access this resource.

Apache/2.4.61 (Debian) Server at 192.168.0.119 Port 80

Screen depuis le serveur



Et voici les erreur dans le fichier error.log →/var/log/apache2/ nano error.log

```
[client 192.168.0.214:59631] AH01797: client denied by server configuration: /var/www/html/secret/
[client 192.168.0.214:59631] AH01797: client denied by server configuration: /var/www/html/secret/
[client 127.0.0.1:36952] AH01797: client denied by server configuration: /var/www/html/secret
```

- On veut protéger l'accès à un dossier par mot de passe :
    - Créer un dossier privé

- Vérifier qu'il y a une directive AccessfileName

```
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives.  See also the AllowOverride
# directive.
#

AccessFileName .htaccess
```

- Dans le dossier prive, créer un fichier .htaccess :

  touch .htaccess

  ls sudo apt install apache2-utils

  ls -a

```
root@raspberrypi:/var/www/html# mkdir prive
root@raspberrypi:/var/www/html# ls
cgi-bin  index.html  motdepasse.htm  phpinfo.php  prive  secret  wordpress
root@raspberrypi:/var/www/html# cd prive/
root@raspberrypi:/var/www/html/prive# touch .htaccess
root@raspberrypi:/var/www/html/prive# ls
root@raspberrypi:/var/www/html/prive# ls -a
.  ..  .htaccess
root@raspberrypi:/var/www/html/prive#
```

- Exemple de fichier .htaccess :

  AuthUserfile /etc/apache2/users
  AuthName « Accès privé »
  AuthType Basic
  require valid-user

```
  GNU nano 7.2
AuthUserfile /etc/apache2/users
AuthName "Accès privé"
AuthType Basic
require valid-user
```

- Création d'un utilisateur Apache

  cd /etc/apache2
  htpasswd -c users toto

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

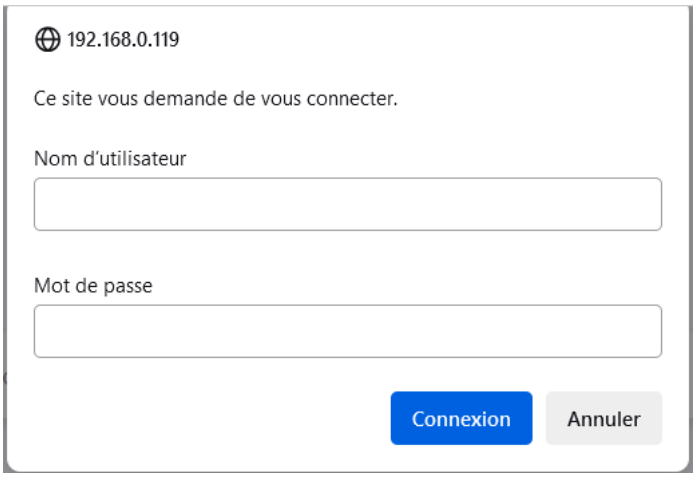
Nous pouvons voir que dans le fichier users notre utilisateur est bien présent.

```
admin@raspberrypi: ~
  GNU nano 7.2                                          users
toto:$apr1$q2uIt2B5$riuvCB0V1r3e1yVhsjSp21
```

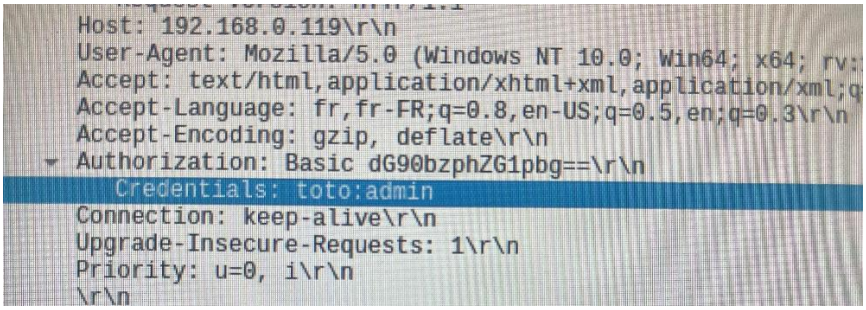
Dans le fichier de configuration il faut rajouter un paragraphe Directory

```
<Directory /var/www/html/prive>
        Options Indexes FollowSymLinks
        AllowOverride AuthConfig
        Require all granted
</Directory>
```

Actualiser votre navigateur, vous verrez une page de connexion.



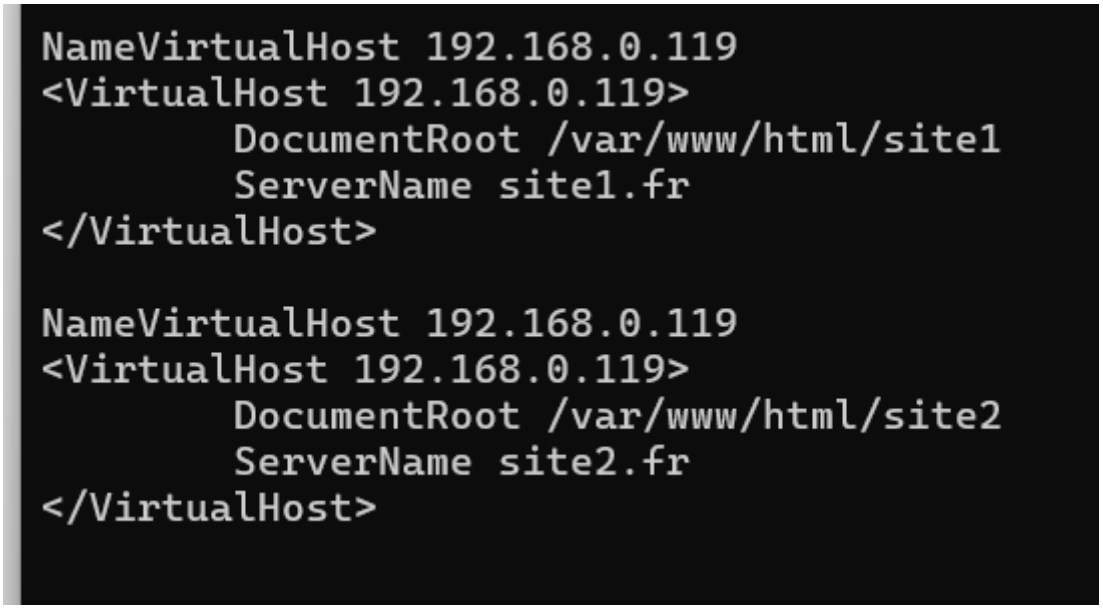
ANALYZE AVEC WIRESHARK



6) Hôtes virtuels

Une machine peut héberger plusieurs sites différents dans les dossiers différents.

NameVirtualHost @IP_serveur
<VirtualHost @IP_serveur>
    DocumentRoot /var/www/html/site1
    ServerName site1.fr
</VirtualHost>



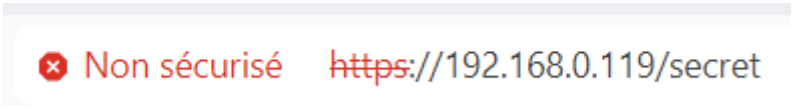
7) HTTPS : Le http sécurisé

Par défaut, Apache utilise le protocole http sur le port 80. Or nous avons vue que les mots de passe circulaient en clair ! Donc nous voulons passer le serveur en HTTPS sur le port par défaut 443.

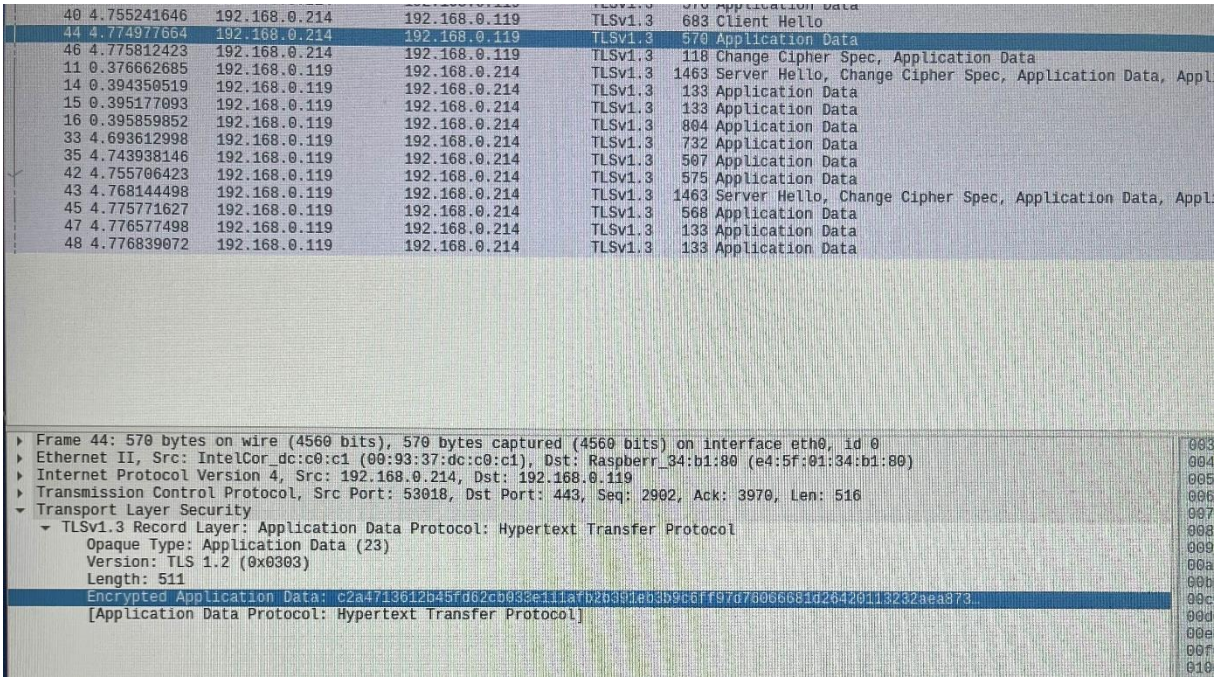
Apache contient deux sites préconfigurés qui pointent tous les deux vers le dossier racine /var/www/html. Le premier, « default », est actif par défaut et permet d'accéder à la page « It Works !». Le second, « default.ssl », est désactivé par défaut.

Commandes d'activation :
    a2emmode ssl
    a2ensite default-ssl
    service apache2 restart

```
root@gaara:/home/admin# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@gaara:/home/admin# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@gaara:/home/admin# service apache2 restart
root@gaara:/home/admin#
```



Depuis Wireshark nous voyons la trame avec le mot de passe crypter.



Le certificat par défaut dans /etc/ssl/private

```
root@gaara:/etc/ssl/private# cat ssl-cert-snakeoil.key
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQC2kN0jGsGt9rno
e8sBQEdlbBeYaBatxD7KEv9tleMy3lb/x4K+oANiB7uAfezd75CIp+2LuGn8YmnW
kENgwPnrTC9OET0dq3nfocrU5iewCp1/CP+DGyUiUjCqpPAZFnBFDIkdPlaQA3U3
Bj5AEmHYo1ELSljakVeJGvIJ3D42GkRDvYzRKlKdJQUjpVB0X06Aghbkn0/phu6R
1lW/ahEuWsEYR5nSCwCROBI4k9DyorgR97JeXCqx4Hal0/YbBJvxIjZs+nd3zH/2
P9VkkjKVPPzG/0cgoZRFQQHCXr41RXbjq+PCSgXAjqzW7RD5iJ6QkHgfLNxUm53G
BkEyTTftAgMBAAECggEAD6HyaZPcM0T98MgO3z3Nj0P09kYtcPYtqTh8ujm8MWoj
+XVwPtntSDrgxZUiz/M4RVdZyh3Bp4yWwvpp5pJHpCwLHexwdwasoidUuqu1iMez
6kGPPX74zEpM5APJEmDfPq5e3Evu9hhxtEM619orGjWL1kNovdh1riYKRSlgg1nR
3J2H5nHxJOAwYpz04rASy+Vi3DmbYc++EKQiErDpbe4Z5CpGQVqHjivJsq2LEjiC
nT2zcgYRoDS/gcETvPHV+LJqrBge8LdLWaV2nChnn01Yxy8WYEGWJqPfDUboXaX4
fBYRuqoAV1MnEeUsa0VXjK+7Jxp8+nQ4VNWzX2tveQKBgQDkBIHHreG+AZILuNyd
FiddFVgupZcKHQWwo3kGE8EZArPaqCaPae5i725yyrzDNABUCPxjbTz5QD9jDr+v
+dF2f9glAwqEdzVQpjudLSHp7PbXKDhghkKq6hbzx+Bh3bJJXeIts1+YLuMXsC6k
PEAGIrnGoUshbYbUblvXOrnpiQKBgQDM+G1MuozsqOzXpUYyiNzDflaFmjbDtbZz
QMyztWxQZIOcZfir/kOCfb6iecULubQtmcWkGfvpaTfkqPJJ2Ls4B9XwxfKvTOZU
e6XSta/0XhmBai9Ukt5CkHUIB6gJxRnlEPqoKw1HxUPfpO+/NXoCLtly/YzgNbre
5QwtLMOWRQKBgQDKsT701BuCHj9HcFOcyJ7BXkH/ZW/4q4ftV8QzpMfYhdB/pU5x
2kbxZ8oU/SgDlmxcR4ikkezL3VL7qZa4xihvYIIxlJx62MVqJfuEYyRsoBtlc/l5
xJ/q8yDoIFvUY5EjKvPukC+ARt3UpucvU+fODHx8l9EDrCtZF6lIkInHkQKBgQC1
KEBaF2skY+zob2/KjEIqGohYDsOmV/1IghYqIwf3Sc80tL+DK/f45jBOI7evNFN2
FjmjXz/RvSgkVdnGwOI/lFVXN2zV2O6d7qbRw7kfDugWBIaOhjwgstJs5W90KxtG
5KnVZi+Xplbl3D31TktmkIep5UTUiUh/ZgtMBgtbzQKBgG5EicMod/8WxqXtzyH6
fSHSBSbDBRfeITJ8tzQUef3nHQiL5hXV/DC6GG1+jPqwhOsacN9TKQaGDssvnMSC
U3biIskdlZ278OIutvXualp7xr4/sSCaJYnwP/TJ1z/FitVAFLU50yENvltTAQYc
dffpubht6sOV/q5mXLSNLe5D
-----END PRIVATE KEY-----
root@gaara:/etc/ssl/private#
```

Création d'un nouveau certificat avec la commande openssl et des options . Exemple :

> openssl req -x509 -nodes -days 365 -newkey rsa:2048 -sha256 -out /etc/apache2/server.crt -keyout /etc/apache2/server.key

Deux manière pour forcer le HTTPS :

- Soit désactiver le HTTP : a2dissite default
- Rediriger le HTTP vers HTTPS : il faut remplacer le fichier /etc/apache2/sites-available /default.conf: nano default.conf redirect permanent / https://localhost

Utilisation de MySql (MariaDB)

1) Introduction

MySql est un SGBG (systéme de Gestion de Bases de données) libre, complémentaire de PHP, pour créer des sites Web dynamiques.
Autres SGBD libre : Postgresql
Autres SGBD propriétaires : Oracle, Microsoft SQL Server

2) Installation

Déjà faite, vérification avec http://localhost/phpinfo.php

3) Fonctionnement

Architecture client/serveur.
Serveur mysql

4) Commande mysql

Entrée dans l'éditeur : mysql -u root -p
Création d'une base Films :

```
CREATE DATABASE Films ;
```

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

```
GRANT ALL PRIVILEGES ON Films.* TO adminFilms@localhost
INDENTIFIED BY 'admin;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON Films.*TO adminFilms@localhost IDENTIFIED BY 'admin';
Query OK, 0 rows affected (0,004 sec)
```

Sortie : exit
Reconnexion : mysql -u adminFilms -p Films

```
root@gaara:~# mysql -u adminFilms -p Films
Enter password:
```

Création d'une table :

```
CREATE TABLE FilmSimple (
        Titre VARCHAR(30),
        Année INTEGER
);
```

```
MariaDB [Films]> CREATE TABLE FilmsSimple(
    -> titre VARCHAR(30),
    -> annee INTEGER
    -> );
Query OK, 0 rows affected (0,026 sec)
```

Verification : DESC FilmSimple ;

```
MariaDB [Films]> DESK FilmsSimple;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual t
hat corresponds to your MariaDB server version for the right syntax to use n
ear 'DESK FilmsSimple' at line 1
```

Remplissage :

INSERT INTO FilmSimple (titre,année) VALUES (‘Pulp Fiction’, 1994),
(‘Alien’, 1979)
(‘Titanic’, 1997)

```
MariaDB [Films]> INSERT INTO FilmsSimple (titre,annee) VALUES ('Pulp Fiction', 1994), ('Joker', 2024), ('Titanic', 1997), ('Alien', 1979);
Query OK, 4 rows affected (0,004 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Interrogation : exemple

SELECT titre FROM FilmSimple WHERE annee=1997;

Ajouter, Supprimer, Modifier des films
Tester des requêtes

Ajouter :

```
MariaDB [Films]> INSERT INTO FilmsSimple (titre,annee) VALUES ('Batman',1989);
Query OK, 1 row affected (0,002 sec)
```

Modifier : Changer la date

```
MariaDB [Films]> UPDATE FilmsSimple
    -> SET annee = 1992
    -> WHERE titre = 'Batman';
Query OK, 1 row affected (0,002 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [Films]> SELECT * FROM FilmsSimple;
+--------------+-------+
| titre        | annee |
+--------------+-------+
| Pulp Fiction |  1994 |
| Joker        |  2024 |
| Titanic      |  1997 |
| Alien        |  1979 |
| Batman       |  1992 |
+--------------+-------+
5 rows in set (0,001 sec)
```

Supprimer :

```
MariaDB [Films]> DELETE FROM FilmsSimple
    -> WHERE titre = 'Batman';
Query OK, 1 row affected (0,003 sec)
```

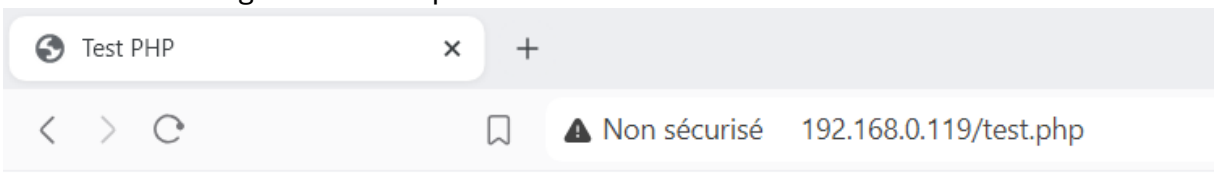
```
+--------------+-------+
| titre        | annee |
+--------------+-------+
| Pulp Fiction |  1994 |
| Joker        |  2024 |
| Titanic      |  1997 |
| Alien        |  1979 |
+--------------+-------+
4 rows in set (0,001 sec)
```

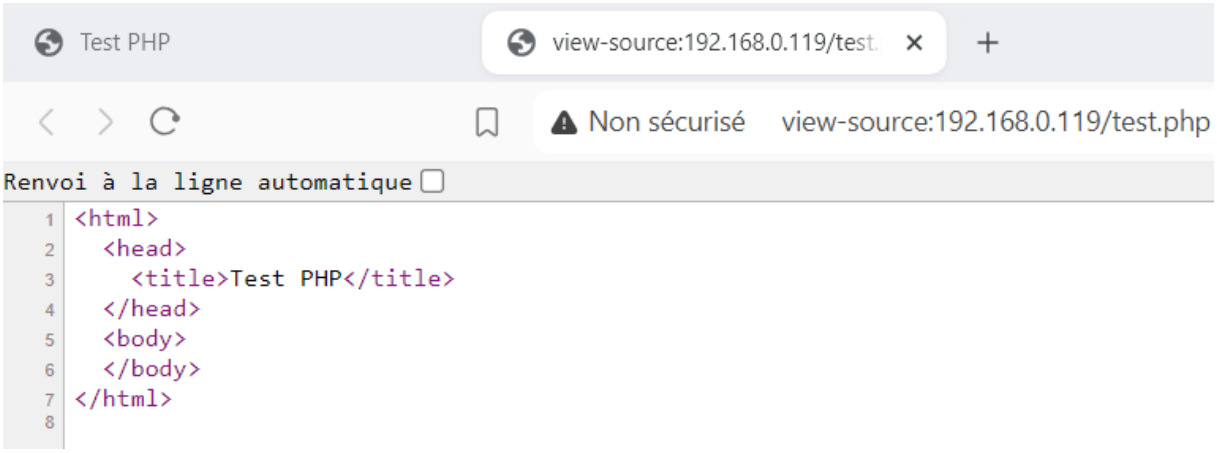
5) Lien avec PHP

On crée un fichier test.php dans /var/www/html :

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

Test dans un navigateur. Ctrl+u pour vérifier le code





6) Administration MySql

- Ajouter un nouvel utilisation sur la base Films avec tous les droits, puis enlevez lui les droits d'écriture. Testez !

```
MariaDB [(none)]> CREATE USER 'tot'@'localhost' IDENTIFIED BY 'toto';
Query OK, 0 rows affected (0,002 sec)
```

Accorder les privilège

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON Films.* TO 'tot'@'localhost';
Query OK, 0 rows affected (0,003 sec)
```

Retirer les droits d'écriture

```
MariaDB [(none)]> REVOKE INSERT, UPDATE, DELETE ON Films.* FROM 'tot'@'localhost';
Query OK, 0 rows affected (0,005 sec)
```

Test

```
MariaDB [Films]> INSERT INTO FilmsSimple (titre, annee) VALUES ('Blanch_Fesse_Et_Les_7_Mains', '2012');
ERROR 1142 (42000): INSERT command denied to user 'tot'@'localhost' for table `Films`.`FilmsSimple`
MariaDB [Films]>
```

Créer un nouvel utilisateur qui a uniquement les droits de sélection.

```
MariaDB [(none)]> GRANT SELECT ON Films.* TO 'tot'@'localhost';
Query OK, 0 rows affected (0,002 sec)
```

Test , ici nous voyons que l'utilisateur « TOT » peux sélectionner des élément mais il ne peut pas en créer ni les modifier.

```
MariaDB [Films]> INSERT INTO FilmsSimple (titre, annee) VALUES ('batman2', '1234');
ERROR 1142 (42000): INSERT command denied to user 'tot'@'localhost' for table `Films`.`FilmsSimple`
MariaDB [Films]> SELECT * FROM FilmsSimple
    -> ;
+--------------+-------+
| titre        | annee |
+--------------+-------+
| Pulp Fiction |  1994 |
| Joker        |  2024 |
| Titanic      |  1997 |
| Alien        |  1979 |
+--------------+-------+
4 rows in set (0,001 sec)

MariaDB [Films]>
```

Quelle sont les options de la commande mysqladmin ?

- **-u** ou **--user** : Spécifie le nom d'utilisateur pour se connecter au serveur MySQL.
- **-p** ou **--password** : Demande le mot de passe de l'utilisateur. Si utilisé sans argument, il demande d'entrer le mot de passe interactif.
- **-h** ou **--host** : Définit l'adresse du serveur MySQL (par exemple, localhost ou une adresse IP).
- **-P** ou **--port** : Spécifie le port à utiliser pour se connecter à MySQL.
- **-S** ou **--socket** : Indique le chemin du fichier de socket pour se connecter via un socket Unix.
- **-r** ou **--relative** : Affiche les résultats relatifs (utile pour les statistiques).
- **-i** ou **--sleep** : Réexécute la commande à un intervalle défini (en secondes).

```
admin@gaara:~ $ mysqladmin -u root -p status
Enter password:
Uptime: 26393  Threads: 1  Questions: 172  Slow queries: 0  Opens: 34  Open tables: 27  Queries per second avg: 0.006
admin@gaara:~ $ *
```

Utiliser la commande mysqldump pour sauvegarder votre base dans un fichier SQL.

```
admin@gaara:~ $
mysqldump -u root -p Films > sauvegarde_Films.sql
Enter password:
```

Vérifier le contenu du fichier.

```
Enter password:
admin@gaara:~ $ cat sauvegarde_Films.sql
-- MariaDB dump 10.19  Distrib 10.11.6-MariaDB, for debian-linux-gnu (aarch64)
--
-- Host: localhost    Database: Films
-- ------------------------------------------------------
-- Server version       10.11.6-MariaDB-0+deb12u1

/*!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' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `FilmsSimple`
--

DROP TABLE IF EXISTS `FilmsSimple`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `FilmsSimple` (
  `titre` varchar(30) DEFAULT NULL,
  `annee` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

Supprimer la base Films.

```
MariaDB [(none)]> DROP DATABASE Films
    -> ;
Query OK, 1 row affected (0,027 sec)

MariaDB [(none)]>
```

Recréer la base Films en importante le fichier SQL.

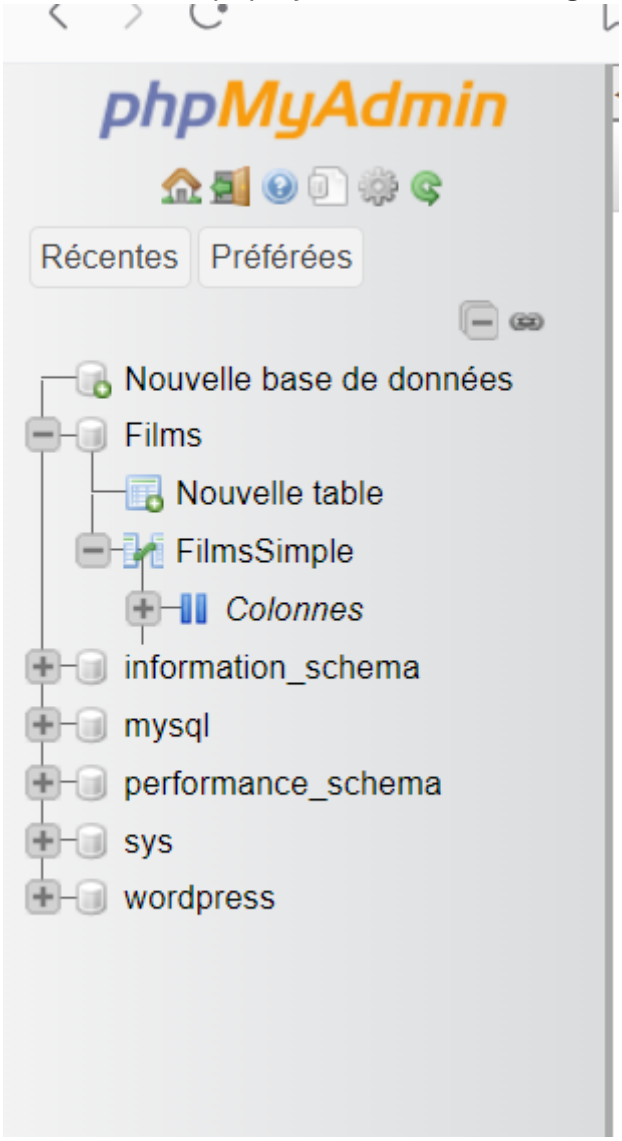
```
MariaDB [(none)]> CREATE DATABASE Films;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> USE Films;
Database changed
```

```
MariaDB [Films]> exit
Bye
```

```
admin@gaara:~ $ mysql -u root -p Films < sauvegarde.sql
-bash: sauvegarde.sql: Aucun fichier ou dossier de ce type
admin@gaara:~ $ ls
Bookshelf  Desktop  Documents  Images  Modèles  Musique  Public  sauvegarde_Films.sql  Téléchargements  Vidéos
admin@gaara:~ $ mysql -u root -p Films < sauvegarde_Films.sql
Enter password:
admin@gaara:~ $
```

Afficher phpmyadmin dans le navigateur



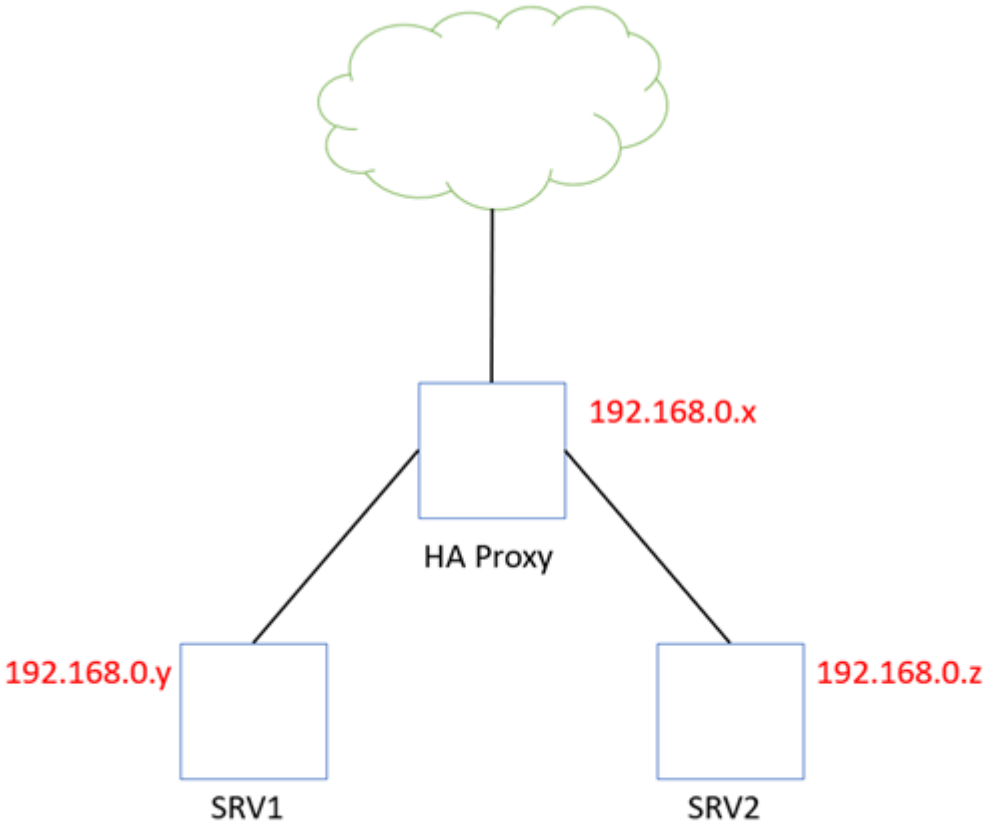
- Rajouter un film, le modifier , le supprimer

titre	annee
Pulp Fiction	1994
Joker	2024
Titanic	1997
Alien	1979
Le combat de l'année	2024

- Rajouter un utilisateur avec droits de sélection
- Se connecter avec cet utilisateur et faire une sélection
- Exporter la table FilmsSimple dans FilmsSimple.sql
- Supprimer la table et la ré-importer

Cluster Apache

HAProxy (High Availability Proxy) est un logiciel de balance de charge open-source qui permet de répartir les requêtes entre plusieurs serveurs pour assurer la haute disponibilité du service.



SRV1 = 192.168.0.119
SRV2 = 192.169.0.180
HA PROXY = 192.168.0.179

Configuration HAProxy :

```
Sudo apt install haproxy
```

Fichier /etc/haproxy/haproxy.cfg ->copie .old

```
Frontend http-in
        bind *:80
        Mode http
        default_backend serveurs
```

```
backend servers
        Balance roudrobin
        Option forwardfor
        Option http-server-close
        Server SRV1 192.168.0.119 check
        Server SRV2 192.168.0.180 check
```