

Serveur Apache

SISR
ALEXIS FRUCTUS

Table des matières

TP Serveur LAMP	2
I. Introduction	2
1) Installation :	2
▪ Apache	2
▪ PHP	2
▪ Mysql	2
▪ PHPMyAdmin	2
2) Démarrage	2
II. Test	3
▪ Test apache :	3
▪ Test php :	3
▪ Test Mysql :	4
▪ Test PHPMyAdmin :	5
III. Utilisation	5
1) Installation de WordPress	6
2) Test de wordpress	6
IV. Apache	6
1) Historique	6
2) Le protocole HTTP	7
3) Requêtes HTTP	8
4) Configuration de base	10
5) Script CGI simple	16
6) Protection des accès	18
7) Serveurs virtuels	21
8) HTTPS : le http sécurisé	22
V. Utilisation de MySql	23
1) Introduction	23
2) Installation	24
3) Fonctionnement	24



4) L'architecture	24
5) Utilisation de PHP	26
VI. Cluster serveur apache	28
1) HAProxy	29
2) Installation	29
3) Configuration	30
4) Test	31

TP Serveur LAMP

I. Introduction

LAMP : acronyme de Linux-Apache-Mysql-PHP

LAMP regroupe les 4 ensembles de paquets nécessaires pour la mise en place d'un serveur web Dynamique.

1) Installation :

- *Apache*

```
sudo apt install apache2
```

Au passage, nous allons en profiter pour donner des droits au dossier d'apache qui vous permettra de facilement administrer les sites. Pour cela, lancez les commandes suivantes :

```
sudo chown -R alex:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/
```

- *PHP*

```
sudo apt install php php-mbstring
```

- *Mysql*

```
sudo apt install mariadb-server php-mysql
```

- *PHPMyAdmin*

```
sudo apt install phpmyadmin
```

Comme nous avons déjà configuré la base de données, choisissez non à la question concernant l'utilisation de dbconfig-common. Choisissez d'utiliser PHPMyAdmin pour un serveur Apache. Pour le mot de passe root, il s'agit de celui que vous avez utilisé pour MySQL. Vous devez aussi activer l'extension mysqli si cela n'est pas encore fait. Pour cela, utilisez les commandes ci-dessous.

```
sudo phpenmod mysqli
sudo /etc/init.d/apache2 restart
```



2) Démarrage

```
Service mysql start  
Service apache2 start
```

II. Test

▪ *Test apache :*

Une fois l'installation terminée, nous pouvons tester qu'Apache fonctionne correctement en nous rendant sur l'adresse de la Raspberry.

Pour cela, il faut tenter d'accéder à la Raspberry depuis le port 80 (ce port n'étant pas encore ouvert depuis l'extérieur, il faudra le faire depuis la Raspberry elle-même). Ne vous inquiétez pas, c'est très facile. Il vous suffit d'ouvrir le navigateur web de la Raspberry (par défaut Midori sous Raspbian), et d'aller à l'adresse « <http://127.0.0.1> ». Vous devriez alors obtenir une page avec un message du genre « *It works!* » et plein d'autre texte.

Si jamais vous ne possédez pas d'interface graphique sur votre Raspbian, ou que vous utilisez le SSH pour vous connecter à votre Raspberry, vous pouvez utiliser la commande suivante :

```
wget -O verif_apache.html http://127.0.0.1
```

Cette commande va enregistrer le code HTML de la page dans le fichier « *verif_apache.html* » dans le répertoire courant.

Vous n'avez donc plus qu'à lire le fichier avec la commande

```
cat ./verif_apache.html
```

Si vous voyez marqué à un endroit dans le code « *It works!* », c'est qu'Apache fonctionne.

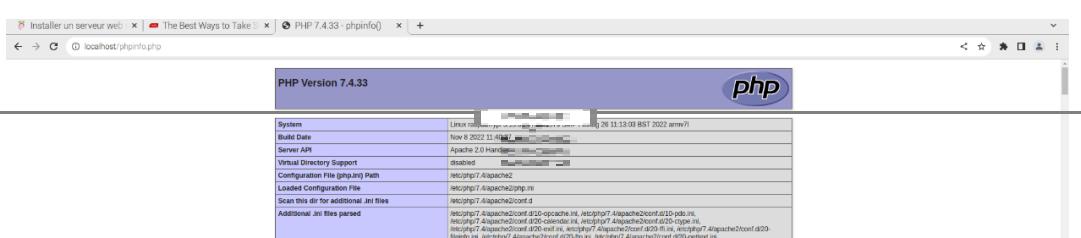
```
alex@raspberrypi:~ $ wget -O verif_apache.html http://127.0.0.1  
-- 2022-11-16 08:48:19 -- http://127.0.0.1/  
Connexion à 127.0.0.1:80... connecté.  
requête HTTP transmise, en attente de la réponse... 200 OK  
Taille : 10701 (10K) [text/html]  
Sauvegarde en : « verif_apache.html »  
  
verif_apache.html 100%[=====] 10,45K ---.KB/s ds 0s  
  
2022-11-16 08:48:19 (25,2 MB/s) - « verif_apache.html » sauvegardé [10701/10701]  
  
alex@raspberrypi:~ $ cat ./verif_apache.html  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
    <title>Apache2 Debian Default Page: It works</title>  
    <style type="text/css" media="screen">
```

▪ *Test php :*

Dans `/var/www/html`, créer le fichier `phpinfo.php` et écrivez la ligne suivante dans le fichier

```
echo "<?php phpinfo(); ?>"
```

Navigateur : <http://localhost/phpinfo.php>



- *Test Mysql :*

```
sudo mysql --user=root
```

Nous allons maintenant supprimer l'utilisateur root et créer un nouvel utilisateur root, car celui par défaut n'est utilisable que par le compte administrateur du système, et n'est donc pas accessible aux script PHP du serveur.

Pour cela, une fois connecté à MySQL, lancez les commandes suivantes (remplacez password par le mot de passe de votre choix) :

```
DROP USER 'root'@'localhost';
CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
```

```
alex@raspberrypi:~ $ sudo mysql --user=root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.15-MariaDB-0+deb11u1 Raspbian 11

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

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

MariaDB [(none)]> DROP USER 'root'@'localhost';
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> CREATE USER 'root'@'localhost' IDENTIFIED BY 'qzds';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> exit
Bye
alex@raspberrypi:~ $ sudo mysql --user=root --password=qzds
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.5.15-MariaDB-0+deb11u1 Raspbian 11

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

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

MariaDB [(none)]> exit
Bye
```

- *Test PHPMyAdmin :*

PHPMyAdmin vous posera plusieurs questions concernant ses réglages.

Comme nous avons déjà configuré la base de données, choisissez **no** à la question concernant l'utilisation de **dbconfig-common**. Choisissez d'utiliser PHPMyAdmin pour un serveur Apache. Pour le mot de passe root, il s'agit de celui que vous avez utilisé pour MySQL. Vous devez aussi activer l'extension mysqli si cela n'est pas encore fait. Pour cela, utilisez les commandes ci-dessous.

```
sudo phpenmod mysqli
sudo /etc/init.d/apache2 restart
```

Pour vérifier le bon fonctionnement de PHPMyAdmin, vous allez simplement tenter d'y accéder, en utilisant l'adresse de votre Raspberry suivie de /phpmyadmin. Par exemple, en local ce sera « <http://127.0.0.1/phpmyadmin> ».

Si jamais vous avez une erreur, cela peut venir du fait que PHPMyAdmin se soit installé dans un autre dossier. Dans ce cas, essayez la commande :

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```



Il faut peut-être donner les droites au fichier pour que ça marche.

```
alex@raspberrypi:~ $ sudo chown -R alex:www-data /var/www/html/
alex@raspberrypi:~ $ sudo chmod -R 770 /var/www/html/
```

The screenshot shows the phpMyAdmin interface running on a Raspberry Pi. The top bar shows the command line: 'alex@raspberrypi:~ \$ sudo chown -R alex:www-data /var/www/html/'. Below the terminal, the browser window displays the phpMyAdmin dashboard. The left sidebar lists databases: 'Nouvelle base de données', 'information_schema', 'mysql', and 'performance_schema'. The main content area has several sections: 'Paramètres généraux' (General parameters), 'Paramètres d'affichage' (Display parameters), 'Serveur de base de données' (Database server), 'Serveur Web' (Web server), and 'phpMyAdmin' (phpMyAdmin version). A red warning box at the bottom states: 'Le stockage de configurations phpMyAdmin n'est pas complètement configuré, certaines fonctionnalités ont été désactivées. Voir l'analyse du problème. Ou encore aller sur l'onglet <Opérations> de l'importe quelle base de données pour le définir à cet endroit.'

Trouver la version de php (7.4)et mysql (7.4.33)

III. Utilisation

- Installer le système de gestion de contenu (CMS) de votre choix : WordPress, Joowla, Drupal...
Tester

1) Installation de WordPress

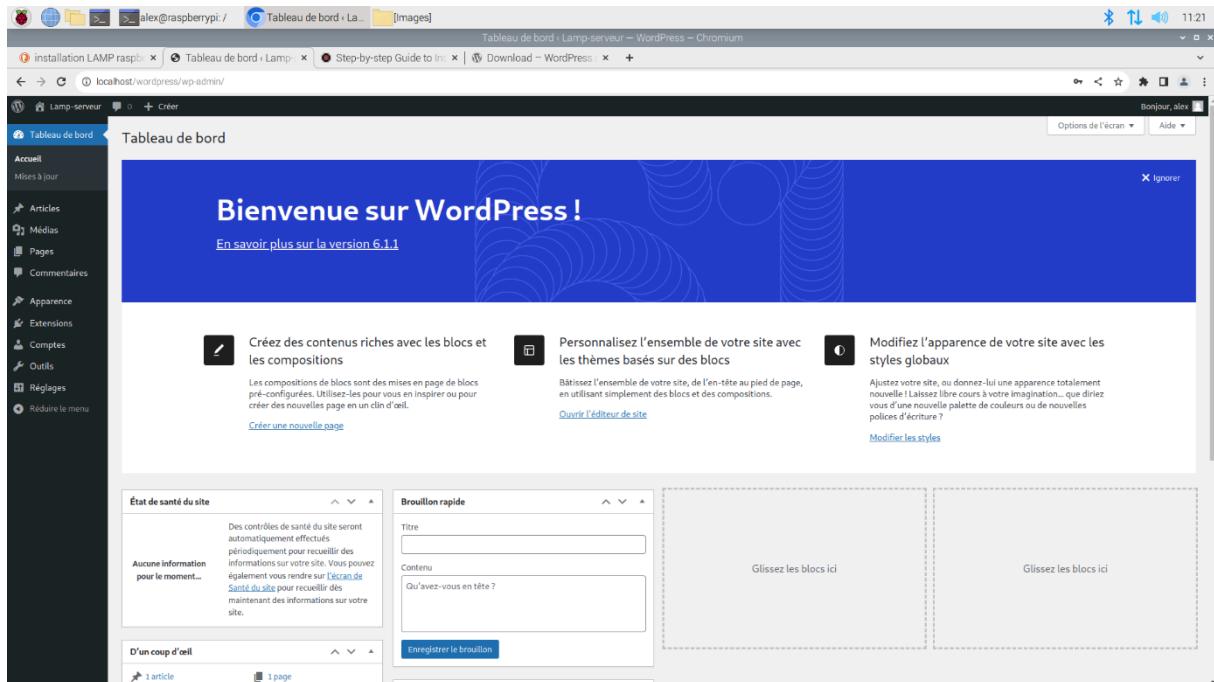
On crée une nouvelle base de données pour wordpress avec les commandes suivantes :

```
Sudo mysql -u root -p qzds
```

Une fois connecté à MariaDB on continue avec les commandes suivantes

```
CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'qzds';
CREATE DATABASE wordpress;
quit
```

2) Test de wordpress



On arrive sur la page d'accueil de wordpress, on peut à présent créer notre site web ou blog

IV. Apache

1) Historique

Internet désigne l'interconnexion de réseaux au niveau mondial. (Année 60, ministère de la défense américain, utilisation du mot à partir de 1972).

World Wide Web désigne le système hypertexte constitué par l'ensemble des pages qui utilisent le protocole http. Http est créé en 1989 dans les laboratoires de CERN à Genève. Premier navigateur convivial (Mozaic) en 1993, Netscape Navigator (1994), Mozilla Firefox (2004). La fondation apache

La apache software foundation est une organisation à but non lucratif dont l'objet est de développer des logiciels libres. Elle est née en 1995. Elle n'a pas de salariés, mais des certains bénévoles dans le monde répartis sur une centaine de projets dont http Apache, LDAP apache directory, spamassassin...

La licence apache est une licence libre compatible avec la GNU-GPL, mais il n'est pas obligatoire de republier sous licence apache. Tout développeur peut rester propriétaire de son code mais donne à la fondation l'autorisation de l'utiliser et de le protéger.

Http apache

La version 2.4 d'apache est sortie en 2012, utilisée sur plus de 50% des serveurs. Conception modulaire. Les hôtes virtuels sont compris dans le module de base. Le module mod_ssl permet le protocole sécurisé HTTPS. Plusieurs modules concernent la journalisation : mod_auth_config, mod_log_debug.... Pour les alias : mod_alias. Pour les redirections : mod_securite. Pour utiliser apache comme proxy : mod_proxy, mod_proxy_ftp

Où trouver les modules installés ?

On les trouve modules installés sur la page phpinfo

2) Le protocole HTTP

HTTP2 sortie en 2015.

Déroulement de la connexion :

- Connexion du client au serveur
- Envoi d'une requête GET par le client
- Réponse du serveur
- Envoi d'une 2e requête GET avec demande de fermeture
- 2e réponse du serveur
- Fermeture de la connexion

La première ligne d'une requête HTTP Précise la méthode d'interrogation :

- GET** : récupération d'un document
- HEAD** : récupération des entêtes 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 contiennent de nombreuses informations :

- Coté client (requête) :
 - o **Host** : indique au serveur quel hôte virtuel il veut interroger
 - o **Referer** : le site précédent
 - o **User-agent** : le navigateur que l'on utilise
 - o **Accept** : les formats de fichiers acceptés
 - o **Accept-language** : les langages acceptés en réponse
 - o **Authorization** : informations d'authentification
 - o **Connection** : pour fermer la connexion
- Coté serveur (réponse) :
 - o **Date**
 - o **Server** : logiciel de serveur
 - o **Content-type** : format des données renvoyées
 - o **Content-length** : taille totale du fichier envoyé
 - o **Expires** : date d'expiration pour suppression du cache
 - o **Last-modified** : date de dernière modification

Les codes de réponses du serveur sont classés en 5 catégories :

- 100 à 199 : message informatif
- 200 à 299 : succès de la requête
- 300 à 399 : redirections
- 400 à 499 : problème !
 - o 401 : requête non autorisée
 - o 404 : le fichier n'existe pas
 - o 405 : méthode GET ou POST incorrecte
- 500 à 599 : erreurs internes au serveur

3) Requêtes HTTP

Analyser quelques trames pour repérer les éléments ci-dessus.



Pour commencer j'essaye de me connecter au serveur pour voir s'il répond correctement

The screenshot shows the phpMyAdmin configuration page. On the left, there's a sidebar with 'Récentes' and 'Préférences'. The main area has tabs for 'Bases de données', 'SQL', 'État', 'Comptes utilisateurs', 'Exporter', 'Importer', 'Paramètres', 'Réplication', 'Variables', 'Jeux de caractères', 'Moteurs', and 'Extensions'. The 'Paramètres généraux' section includes fields for 'Modifier le mot de passe' and 'Interclassement pour la connexion au serveur' set to 'utf8mb4_unicode_ci'. The 'Serveur de base de données' section lists 'Srv' as 'localhost via UNIX socket', 'Type de serveur' as 'MariaDB', and 'Version du serveur' as '10.5.15-MariaDB-0+deb11u1 - Raspbian 11'. The 'Serveur Web' section shows 'Apache/2.4.54 (Raspbian)' and 'Version de PHP' as '7.4.33'. The 'phpMyAdmin' section provides links to version information, documentation, and other resources. A red banner at the bottom states: 'Le stockage de configurations phpMyAdmin n'est pas complètement configuré, certaines fonctionnalités ont été désactivées. Voir l'analyse du problème. Ou encore aller sur l'onglet <code></code> d'importe quelle base de données pour le définir à cet endroit.'

C'est un succès, je vais essayer d'atteindre un fichier qui n'existe pas pour voir comment ça réagit

On voit une erreur 404 dans l'onglet de firefox ce qui veut dire que la ressource n'existe pas dans le serveur. Maintenant analysons quelque trame réseau avec Wire Shark pour regarder de plus près les requêtes HTTP.

The screenshot shows a Firefox browser window with the title '404 Not Found'. The address bar shows '192.168.0.231/php.html'. The page content displays the Apache error message: 'Not Found'. Below it, it says 'The requested URL was not found on this server.' and 'Apache/2.4.54 (Raspbian) Server at 192.168.0.231 Port 80'. The status bar at the bottom right shows the date and time as '17/11/2022 08:31'.

On voit avec les trois premières lignes 3 types de requêtes :

Post : lors de l'envoie du formulaire d'authentification

http : pour nous dire que ça a marché et que l'on nous redirige

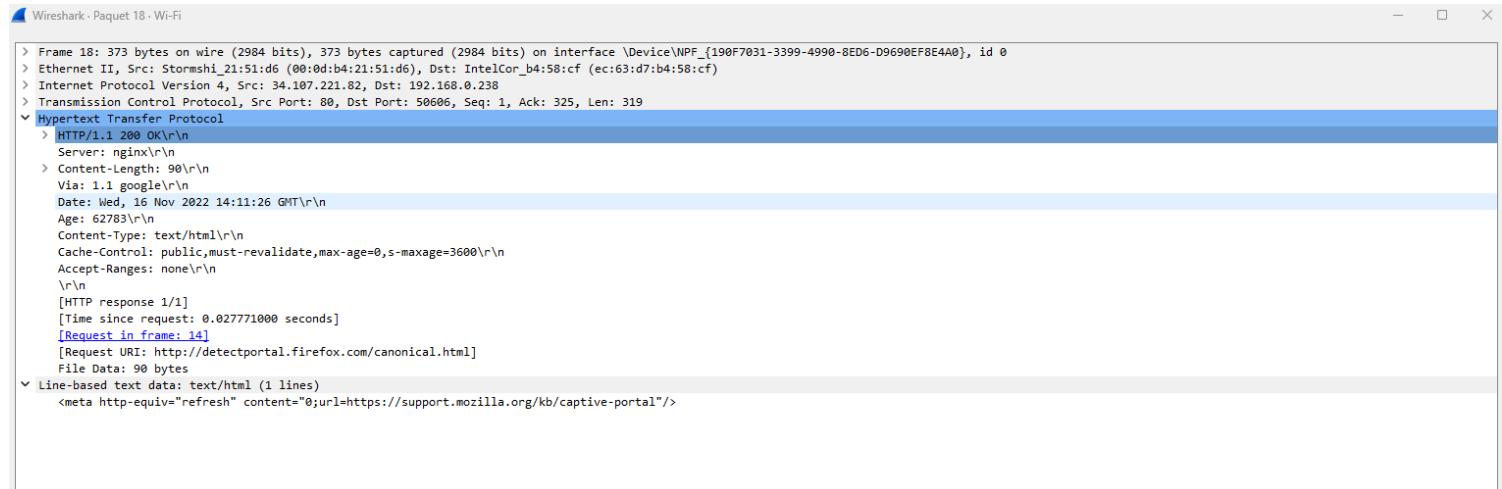
GET : pour demander la suite des pages

18 2.169271	192.168.0.238	192.168.0.231	HTTP	903 POST /phpmyadmin/index.php HTTP/1.1 (application/x-www-form-urlencoded)
21 2.301312	192.168.0.231	192.168.0.238	HTTP	1364 HTTP/1.1 302 Found
22 2.305188	192.168.0.238	34.107.221.82	HTTP	378 GET /canonical.html HTTP/1.1
23 2.305989	192.168.0.238	192.168.0.231	HTTP	876 GET /phpmyadmin/index.php HTTP/1.1
25 2.332176	34.107.221.82	192.168.0.238	HTTP	373 HTTP/1.1 200 OK (text/html)
26 2.335953	192.168.0.238	34.107.221.82	HTTP	380 GET /success.txt?ipv4 HTTP/1.1
27 2.364069	34.107.221.82	192.168.0.238	HTTP	270 HTTP/1.1 200 OK (text/plain)
43 2.492372	192.168.0.231	192.168.0.238	HTTP	1233 HTTP/1.1 200 OK (text/html)
47 2.554338	192.168.0.238	192.168.0.231	HTTP	860 GET /phpmyadmin/themes/pmahomme/css/theme.css?v=5.0.4deb2%2Bdeb11u1&nocache=-2060033308ltr&server-
63 2.566992	192.168.0.231	192.168.0.238	HTTP	279 HTTP/1.1 200 OK (text/css)
65 2.572218	192.168.0.238	192.168.0.231	HTTP	817 GET /phpmyadmin/themes/pmahomme/img/logo_left.png HTTP/1.1
68 2.575009	192.168.0.231	192.168.0.238	HTTP	134 HTTP/1.1 200 OK (PNG)
73 2.676287	192.168.0.238	192.168.0.231	HTTP	945 GET /phpmyadmin/themes/pmahomme/img/left_nav_bg.png HTTP/1.1
82 2.679618	192.168.0.231	192.168.0.238	HTTP	485 HTTP/1.1 200 OK (PNG)
90 2.679888	192.168.0.238	192.168.0.231	HTTP	940 GET /phpmyadmin/themes/pmahomme/img/b_home.png HTTP/1.1
91 2.680001	192.168.0.238	192.168.0.231	HTTP	943 GET /phpmyadmin/themes/pmahomme/img/s_logoff.png HTTP/1.1
92 2.680076	192.168.0.238	192.168.0.231	HTTP	940 GET /phpmyadmin/themes/pmahomme/img/b_docs.png HTTP/1.1
93 2.680136	192.168.0.238	192.168.0.231	HTTP	943 GET /phpmyadmin/themes/pmahomme/img/b_sqlhelp.png HTTP/1.1
94 2.680189	192.168.0.238	192.168.0.231	HTTP	939 GET /phpmyadmin/themes/pmahomme/img/s_cog.png HTTP/1.1
95 2.680261	192.168.0.238	192.168.0.231	HTTP	942 GET /phpmyadmin/themes/pmahomme/img/s_reload.png HTTP/1.1
101 2.682981	192.168.0.231	192.168.0.238	HTTP	1075 HTTP/1.1 200 OK (PNG)
102 2.682981	192.168.0.231	192.168.0.238	HTTP	1046 HTTP/1.1 200 OK (PNG)
103 2.682981	192.168.0.231	192.168.0.238	HTTP	853 HTTP/1.1 200 OK (PNG)
104 2.682981	192.168.0.231	192.168.0.238	HTTP	868 HTTP/1.1 200 OK (PNG)
105 2.682981	192.168.0.231	192.168.0.238	HTTP	966 HTTP/1.1 200 OK (PNG)
106 2.682981	192.168.0.231	192.168.0.238	HTTP	783 HTTP/1.1 200 OK (PNG)
107 2.683204	192.168.0.238	192.168.0.231	HTTP	940 GET /phpmyadmin/themes/pmahomme/img/s_link.png HTTP/1.1

On voit dans la capture au-dessus sur les lignes en bleu que la méthode d'interrogation utilisé est le POST pour envoyer les informations au serveur, ici c'est l'or de la connexion à PhpMyAdmin, puisqu'on utilise http et non HTTPS on peut voir l'identifiant et le mot de passe en clair dans les dernières lignes « Form Item ».

```
Wireshark - Paquet 13 - Wi-Fi
Frame 13: 1369 bytes on wire (10952 bits), 1369 bytes captured (10952 bits) on interface \Device\NPF_{190F7031-3399-4990-8ED6-D9690EF8E4A0}, id 0
Ethernet II, Src: Raspberri_34:b1:80 (e4:5f:01:34:b1:80), Dst: IntelCor_04:58:cf (ec:63:d7:b4:58:cf)
Internet Protocol Version 4, Src: 192.168.0.231, Dst: 192.168.0.238
Transmission Control Protocol, Src Port: 80, Dst Port: 50656, Seq: 1, Ack: 860, Len: 1315
HyperText Transfer Protocol
> HTTP/1.1 302 Found\r\n
Date: Thu, 17 Nov 2022 07:37:49 GMT\r\n
Server: Apache/2.4.54 (Raspbian)\r\n
Set-Cookie: phpMyAdmin=f0f72q5g3cdmu7b3rl16cj1fr9; path=/phpmyadmin/; HttpOnly\r\n
Expires: Thu, 19 Nov 1981 08:52:00 GMT\r\n
Cache-Control: private, max-age=10800\r\n
Last-Modified: Thu, 15 Oct 2020 18:08:52 GMT\r\n
Set-Cookie: goto=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/phpmyadmin/\r\n
Set-Cookie: back=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/phpmyadmin/\r\n
Set-Cookie: phpMyAdmin=permI34pln3jufdd4990pos59; path=/phpmyadmin/; HttpOnly\r\n
[truncated]Set-Cookie: pmaUser-1=e778%22iv%22%3A%22qG0000XIVJ4Hi%28MVYcVioA%3D%3D%22%2C%22mac%22%3A%2281621bc5fbfd81dad007128c7c2fd0f7e4789aa3%22%2C%22payload%22%3A%22QrwI680IyKqwJopFQkjfqg%3D%3D%22%7D; expires=[truncated]Set-Cookie: pmaAuth-1=e778%22iv%22%3A%2254vr6uP%5C%2FrC1wgwF2Jf1tg%3D%3D%22%2C%22mac%22%3A%22dca171d8bae1240608452333abc9130811011a79%22%2C%22payload%22%3A%22017F7HSSc40fCSCBZytFtpjMSPU21031V7n0iBcX-ob_mode: 1\r\n
Location: /phpmyadmin/index.php\r\n
Content-Encoding: gzip\r\n
Vary: Accept-Encoding\r\n
Content-Length: 20\r\n
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=utf-8\r\n
\r\n
[HTTP response 1/6]
[Time since request: 0.138405000 seconds]
[Request in frame: 10]
[Next request in frame: 15]
[Next response in frame: 36]
[Request URI: http://192.168.0.231/phpmyadmin/index.php]
Content-encoded entity body (gzip): 20 bytes -> 0 bytes
File Data: 0 bytes
```

Dans cette capture on peut voir en bleu le nombre 302 qui veut dire qu'on a été redirigé après l'authentification à PHPMyAdmin vers la pages d'accueil de la base de données.



Frame 18: 373 bytes on wire (2984 bits), 373 bytes captured (2984 bits) on interface \Device\NPF_{190F7031-3399-4990-8ED6-D9690EF8E4A0}, id 0
> Ethernet II, Src: Stormshi_21:51:d6 (00:0d:b4:21:51:d6), Dst: IntelCor_b4:58:cf (ec:63:d7:b4:58:cf)
> Internet Protocol Version 4, Src: 34.107.221.82, Dst: 192.168.0.238
> Transmission Control Protocol, Src Port: 80, Dst Port: 50606, Seq: 1, Ack: 325, Len: 319
HTTP/1.1 200 OK\r\nServer: nginx\r\nContent-Length: 90\r\nVia: 1.1 google\r\nDate: Wed, 16 Nov 2022 14:11:26 GMT\r\nAge: 62783\r\nContent-Type: text/html\r\nCache-Control: public,must-revalidate,max-age=0,s-maxage=3600\r\nAccept-Ranges: none\r\n\r\n[HTTP response 1/1]
[Time since request: 0.027771000 seconds]
[Request in frame: 14]
[Request URI: http://detectportal.firefox.comcanonical.html]
File Data: 90 bytes
Line-based text data: text/html (1 lines)
<meta http-equiv="refresh" content="0;url=https://support.mozilla.org/kb/captive-portal"/>

On voit ici le code 200 en bleu ce qui prouve que la requête a été un succès.

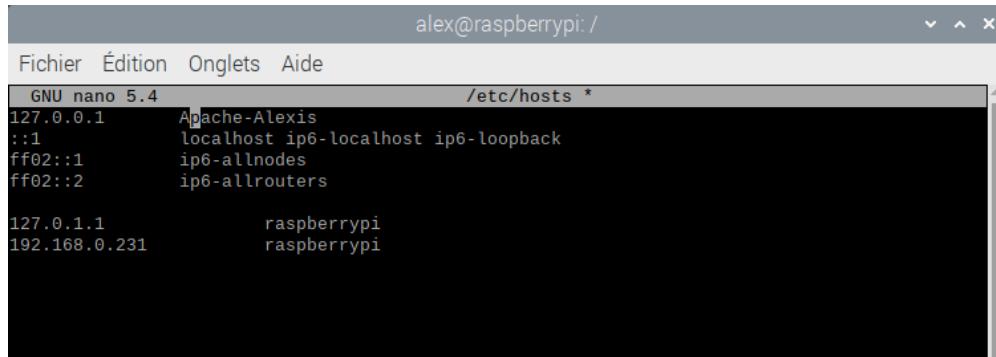
4) Configuration de base

- Démarrage

Demander le statut du démon apache2. Noter les erreurs ou warnings du fichier de logs. PB du nom d'hôtes.

La commande **hostname** vous donne le nom de votre machine.

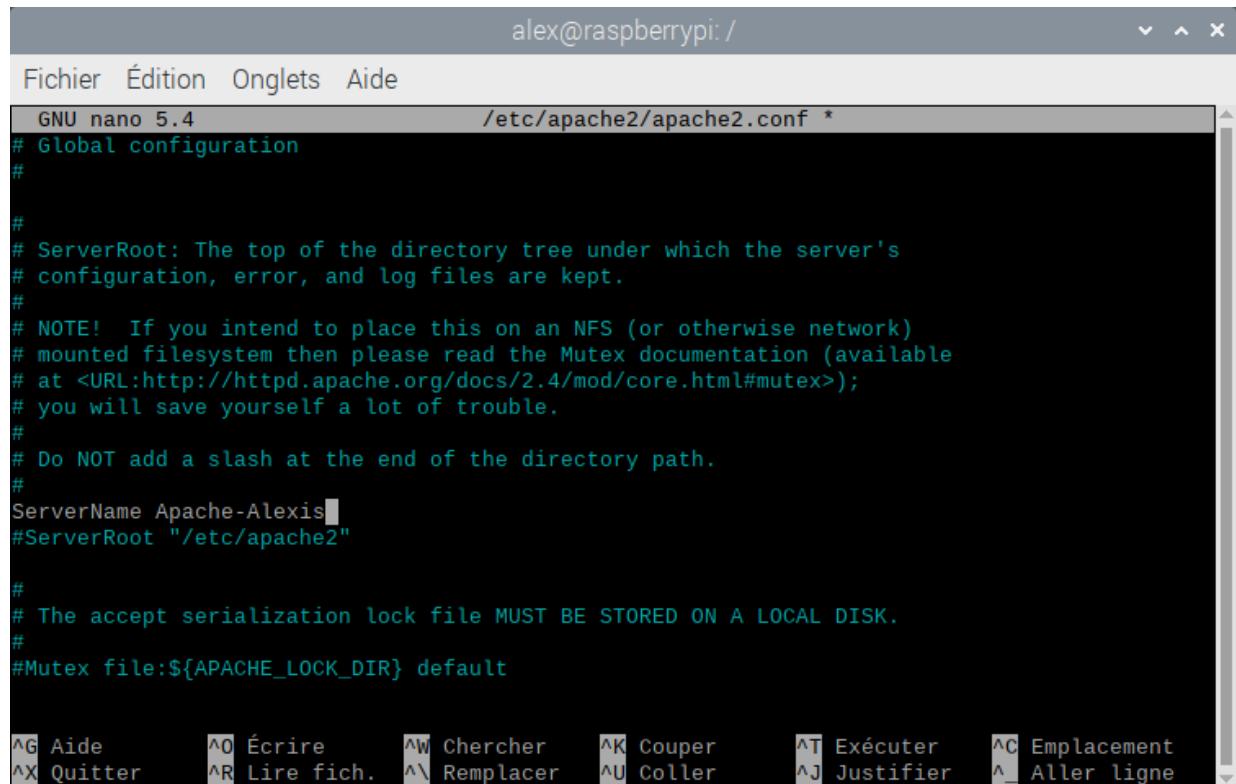
Pour enlever le warning, ajouter dans le fichier **/etc/hosts** le couple **address_ip nom**.



alex@raspberrypi: /

Fichier	Édition	Onglets	Aide
GNU nano 5.4			
/etc/hosts *			
127.0.0.1	Apache-Alexis		
::1	localhost ip6-localhost ip6-loopback		
ff02::1	ip6-allnodes		
ff02::2	ip6-allrouters		
127.0.1.1	raspberrypi		
192.168.0.231	raspberrypi		

Ajoutez dans `/etc/apache2/apache2.conf` la ligne `ServerName Nom`, cette commande sert à donner un nom au serveur et non à changer le nom local pour consulter le serveur.



```
alex@raspberrypi: /
```

Fichier Édition Onglets Aide

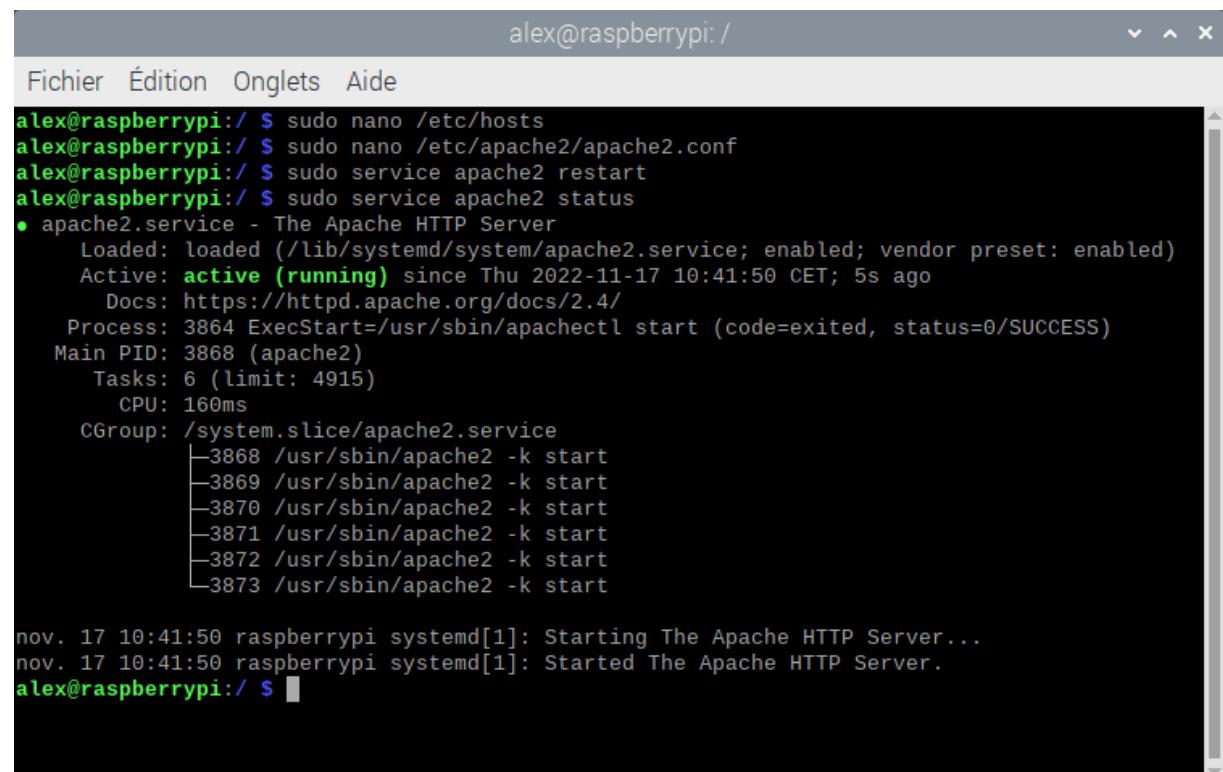
```
GNU nano 5.4          /etc/apache2/apache2.conf *
```

```
# Global configuration
#
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the Mutex documentation (available
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerName Apache-Alexis
#ServerRoot "/etc/apache2"

#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#Mutex file:${APACHE_LOCK_DIR} default
```

^G Aide **^O** Écrire **^W** Chercher **^K** Couper **^T** Exécuter **^C** Emplacement
^X Quitter **^R** Lire fich. **^V** Remplacer **^U** Coller **^J** Justifier **^L** Aller ligne

Relancer le serveur et tester.



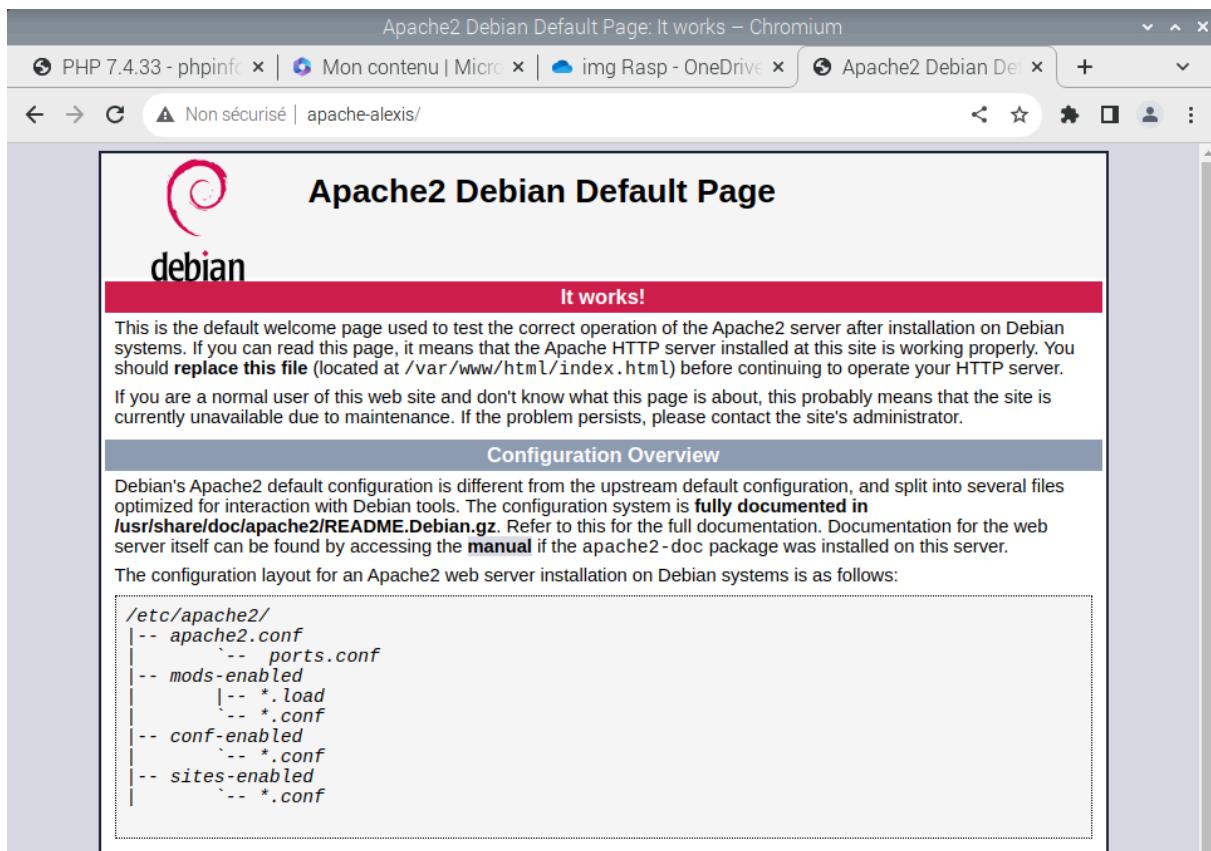
```
alex@raspberrypi: /
```

Fichier Édition Onglets Aide

```
alex@raspberrypi:/ $ sudo nano /etc/hosts
alex@raspberrypi:/ $ sudo nano /etc/apache2/apache2.conf
alex@raspberrypi:/ $ sudo service apache2 restart
alex@raspberrypi:/ $ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-11-17 10:41:50 CET; 5s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 3864 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 3868 (apache2)
    Tasks: 6 (limit: 4915)
      CPU: 160ms
     CGroup: /system.slice/apache2.service
             └─3868 /usr/sbin/apache2 -k start
                 ├─3869 /usr/sbin/apache2 -k start
                 ├─3870 /usr/sbin/apache2 -k start
                 ├─3871 /usr/sbin/apache2 -k start
                 ├─3872 /usr/sbin/apache2 -k start
                 └─3873 /usr/sbin/apache2 -k start

nov. 17 10:41:50 raspberrypi systemd[1]: Starting The Apache HTTP Server...
nov. 17 10:41:50 raspberrypi systemd[1]: Started The Apache HTTP Server.
alex@raspberrypi:/ $
```

L'erreur de nom a été résolue



On voit que maintenant on utilise plus localhost mais Apache-Alexis

- Le fichier de configuration principal

Répondre aux questions :

1. Quelle directive spécifie le port TCP d'écoute ? Quel est le port par défaut ?

On trouve les port par défaut dans `/etc/apache2/ports.conf` et par défaut se sont les port 80 pour HTTP et 443 pour HTTPS quand le SSL est activé. La directive s'appelle « Listen » suivie du port.

```
alex@raspberrypi: /etc/apache2
Fichier Édition Onglets Aide
GNU nano 5.4                               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>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

2. Y a-t-il d'autres fichiers de configuration chargés depuis apache2.conf ? lesquels ? ou sont-ils ?

Oui il y a le fichiers :

ports.conf dans **/etc/apache2**

tous les fichiers en .conf situé dans le dossier **/etc/apache2/mods-enabled**

Ainsi que les fichiers en .htaccess pour trouver des fichiers de configuration supplémentaire

3. Quelle est la directive chargeant un module ?

La directive est **IncludeOptional** et les modules se trouvent dans **/etc/apache2/mods-enabled**.

4. Sous quelle identité Unix exécute le serveur ? puis donner les UID et GID correspondantes.

Il l'exécute sous l'identité 3868(apache2)

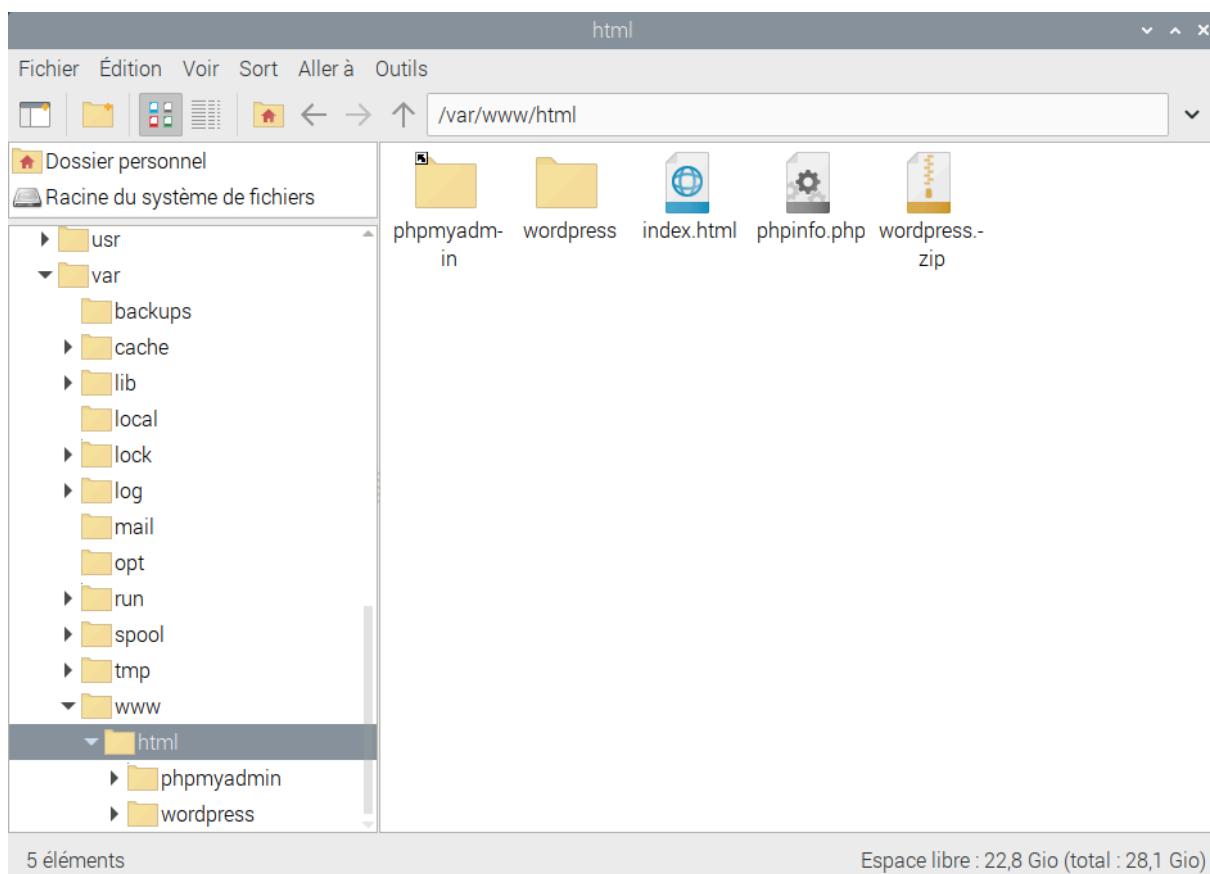
UID est APACHE_RUN_USER et GID est APACHE_RUN_GROUP

5. Quel est le répertoire racine pour les pages services ?

/var/www/html

6. Quelles pages par défaut est renvoyée lorsque l'URL demandée est un dossier ?

Index.html / phpinfo.php



7. Comment sont traitées les requêtes sous formes : <http://serveur/cgi-bin/toto>

L'erreur 404 est renvoyé

8. Quels sont les fichiers de logs générés ? Où sont-ils ? Format ?

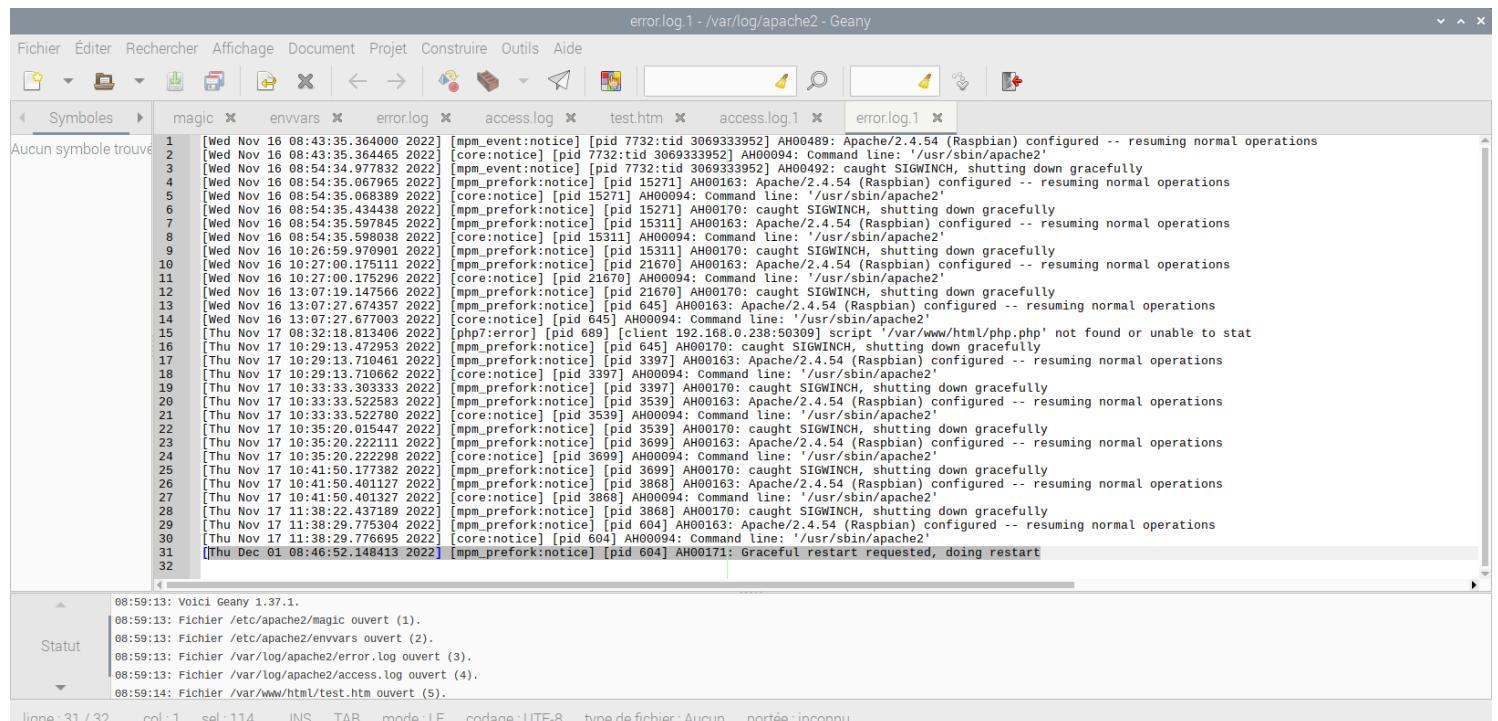
Les fichiers Log sont : **access.log / error.log / other_vhosts_access.log** et ils sont situés sous **/var/log/apache2**, ils sont tous en .log

Fichiers de log

Créer une page HTML minimale test.htm dans le dossier racine.

Redémarrer le serveur. Dans quel fichier de log peut-on constater le redémarrage ?

On peut voir le redémarrage dans le fichier **error.log.1**



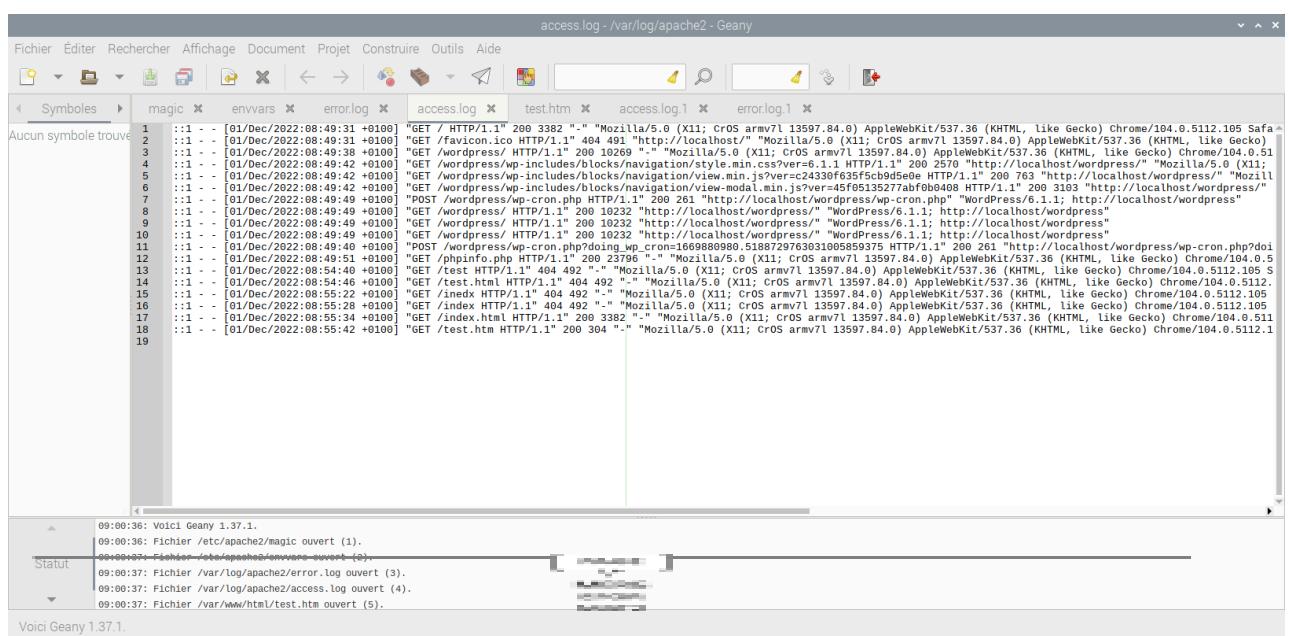
```
error.log.1 - /var/log/apache2 - Geany
Fichier Éditer Rechercher Affichage Document Projet Construire Outils Aide
magic envvars error.log access.log test.htm access.log.1 error.log.1
Aucun symbole trouvé
1 [Wed Nov 16 08:43:35.364000 2022] [mpm_event:notice] [pid 7732:tid 3069333952] AH00489: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
2 [Wed Nov 16 08:43:35.364465 2022] [core:notice] [pid 7732:tid 3069333952] AH00094: Command line: '/usr/sbin/apache2'
3 [Wed Nov 16 08:54:34.356789 2022] [mpm_event:notice] [pid 7732:tid 3069333952] AH00492: caught SIGWINCH, shutting down gracefully
4 [Wed Nov 16 08:54:35.067965 2022] [core:notice] [pid 15271] AH00094: Command line: '/usr/sbin/apache2'
5 [Wed Nov 16 08:54:35.068389 2022] [core:notice] [pid 15271] AH00094: Command line: '/usr/sbin/apache2'
6 [Wed Nov 16 08:54:35.434438 2022] [mpm_prefork:notice] [pid 15271] AH00170: caught SIGWINCH, shutting down gracefully
7 [Wed Nov 16 08:54:35.597845 2022] [mpm_prefork:notice] [pid 15311] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
8 [Wed Nov 16 08:54:35.598032 2022] [core:notice] [pid 15311] AH00094: Command line: '/usr/sbin/apache2'
9 [Wed Nov 16 10:26:59.970901 2022] [mpm_prefork:notice] [pid 15311] AH00170: caught SIGWINCH, shutting down gracefully
10 [Wed Nov 16 10:27:00.175111 2022] [mpm_prefork:notice] [pid 21670] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
11 [Wed Nov 16 10:27:00.175299 2022] [core:notice] [pid 21670] AH00094: Command line: '/usr/sbin/apache2'
12 [Wed Nov 16 13:07:19.147795 2022] [mpm_prefork:notice] [pid 21670] AH00170: caught SIGWINCH, shutting down gracefully
13 [Wed Nov 16 13:07:27.674357 2022] [mpm_prefork:notice] [pid 645] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
14 [Wed Nov 16 13:07:27.677003 2022] [core:notice] [pid 645] AH00094: Command line: '/usr/sbin/apache2'
15 [Thu Nov 17 08:32:18.813409 2022] [php7:error] [pid 689] [client 192.168.0.238:50309] script '/var/www/html/php.php' not found or unable to stat
16 [Thu Nov 17 10:29:13.472953 2022] [mpm_prefork:notice] [pid 645] AH00170: caught SIGWINCH, shutting down gracefully
17 [Thu Nov 17 10:29:13.710461 2022] [mpm_prefork:notice] [pid 3397] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
18 [Thu Nov 17 10:29:13.710662 2022] [core:notice] [pid 3397] AH00094: Command line: '/usr/sbin/apache2'
19 [Thu Nov 17 10:33:33.303333 2022] [mpm_prefork:notice] [pid 3397] AH00170: caught SIGWINCH, shutting down gracefully
20 [Thu Nov 17 10:33:33.522588 2022] [mpm_prefork:notice] [pid 3539] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
21 [Thu Nov 17 10:33:33.522789 2022] [core:notice] [pid 3539] AH00094: Command line: '/usr/sbin/apache2'
22 [Thu Nov 17 10:35:20.015447 2022] [mpm_prefork:notice] [pid 3539] AH00170: caught SIGWINCH, shutting down gracefully
23 [Thu Nov 17 10:35:20.222111 2022] [mpm_prefork:notice] [pid 3539] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
24 [Thu Nov 17 10:35:20.222111 2022] [core:notice] [pid 3539] AH00094: Command line: '/usr/sbin/apache2'
25 [Thu Nov 17 10:41:50.177382 2022] [mpm_prefork:notice] [pid 3689] AH00170: caught SIGWINCH, shutting down gracefully
26 [Thu Nov 17 10:41:50.401327 2022] [mpm_prefork:notice] [pid 3868] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
27 [Thu Nov 17 10:41:50.401327 2022] [core:notice] [pid 3868] AH00094: Command line: '/usr/sbin/apache2'
28 [Thu Nov 17 11:30:23.437180 2022] [mpm_prefork:notice] [pid 3868] AH00170: caught SIGWINCH, shutting down gracefully
29 [Thu Nov 17 11:38:29.775304 2022] [mpm_prefork:notice] [pid 604] AH00163: Apache/2.4.54 (Raspbian) configured -- resuming normal operations
30 [Thu Nov 17 11:38:29.776695 2022] [core:notice] [pid 604] AH00094: Command line: '/usr/sbin/apache2'
31 [Thu Dec 01 08:46:52.148413 2022] [mpm_prefork:notice] [pid 604] AH00171: Graceful restart requested, doing restart
32

08:59:13: Voici Geany 1.37.1.
08:59:13: Fichier /etc/apache2/magic ouvert (1).
08:59:13: Fichier /etc/apache2/envvars ouvert (2).
08:59:13: Fichier /var/log/apache2/error.log ouvert (3).
08:59:13: Fichier /var/log/apache2/access.log ouvert (4).
08:59:14: Fichier /var/www/html/test.htm ouvert (5).

ligne: 31 / 32  col.: 1  sel.: 114  INS TAB mode:LF codage: UTF-8 type de fichier: Aucun portée: inconnu
```

Accéder à votre page HTML. Observe-t-on une trace dans les logs ?

Oui on voit les requêtes envoyées au serveur



```
access.log - /var/log/apache2 - Geany
Fichier Éditer Rechercher Affichage Document Projet Construire Outils Aide
magic envvars error.log access.log test.htm access.log.1 error.log.1
Aucun symbole trouvé
1 ::1 - [01/Dec/2022:08:49:40 +0100] "GET / HTTP/1.1" 200 13597.84.0 "Mozilla/5.0 (X11; Linux armv7l) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
2 ::1 - [01/Dec/2022:08:49:41 +0100] "GET /favicon.ico HTTP/1.1" 404 401 "Mozilla/5.0 (X11; Linux armv7l) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
3 ::1 - [01/Dec/2022:08:49:43 +0100] "GET /wordpress HTTP/1.1" 200 10269 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
4 ::1 - [01/Dec/2022:08:49:42 +0100] "GET /wordpress/wp-includes/blocks/navigation/style.min.css?ver=6.1.1 HTTP/1.1" 200 2570 "http://localhost/wordpress/" "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
5 ::1 - [01/Dec/2022:08:49:42 +0100] "GET /wordpress/wp-includes/blocks/navigation/view.min.js?ver=c24330f635f5cb9d5e0e HTTP/1.1" 200 763 "http://localhost/wordpress/" "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
6 ::1 - [01/Dec/2022:08:49:42 +0100] "GET /wordpress/wp-includes/blocks/navigation/view-modal.min.js?ver=a5f05135277abf0b94088 HTTP/1.1" 200 3103 "http://localhost/wordpress/" "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
7 ::1 - [01/Dec/2022:08:49:42 +0100] "POST /wordpress/wp-cron.php HTTP/1.1" 200 26 "http://localhost/wordpress/wp-cron.php" "Wordpress/6.1.1; http://localhost/wordpress"
8 ::1 - [01/Dec/2022:08:49:43 +0100] "GET /wp-cron.php?doing_wp_cron=1669889880.5188729763031005859375 HTTP/1.1" 200 1032 "http://localhost/wordpress"
9 ::1 - [01/Dec/2022:08:49:42 +0100] "GET /index.php HTTP/1.1" 200 10232 "http://localhost/Wordpress/" "Wordpress/6.1.1; http://localhost/wordpress"
10 ::1 - [01/Dec/2022:08:49:43 +0100] "GET /wordpress HTTP/1.1" 200 10232 "http://localhost/wordpress/" "Wordpress/6.1.1; http://localhost/wordpress"
11 ::1 - [01/Dec/2022:08:49:40 +0100] "POST /wordpress/wp-cron.php?doing_wp_cron=1669889880.5188729763031005859375 HTTP/1.1" 200 261 "http://localhost/wordpress/wp-cron.php?doi
12 ::1 - [01/Dec/2022:08:49:51 +0100] "GET /phpinfo.php HTTP/1.1" 200 23796 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
13 ::1 - [01/Dec/2022:08:54:40 +0100] "GET /test HTTP/1.1" 404 492 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
14 ::1 - [01/Dec/2022:08:54:40 +0100] "GET /test2 HTTP/1.1" 404 492 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
15 ::1 - [01/Dec/2022:08:55:22 +0100] "GET /index.html HTTP/1.1" 404 492 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
16 ::1 - [01/Dec/2022:08:55:22 +0100] "GET /index.html HTTP/1.1" 200 3382 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
17 ::1 - [01/Dec/2022:08:55:34 +0100] "GET /index.html HTTP/1.1" 200 304 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"
18 ::1 - [01/Dec/2022:08:55:42 +0100] "GET /test.htm HTTP/1.1" 200 304 "Mozilla/5.0 (X11; CrOS armv7l 13597.84.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.512.18 Safari/537.36"

09:00:36: Voici Geany 1.37.1.
09:00:36: Fichier /etc/apache2/magic ouvert (1).
09:00:37: Fichier /etc/apache2/envvars ouvert (2).
09:00:37: Fichier /var/log/apache2/error.log ouvert (3).
09:00:37: Fichier /var/log/apache2/access.log ouvert (4).
09:00:37: Fichier /var/www/html/test.htm ouvert (5).

Voici Geany 1.37.1.
```

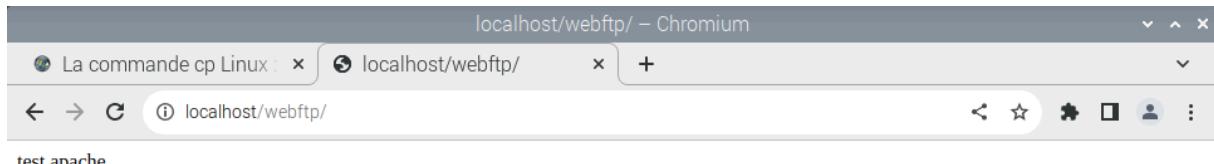
- Accès aux dossiers

Créer un dossier webftp dans **/var/www/html** et mettre une copie de **test.htm**

Test avec le navigateur

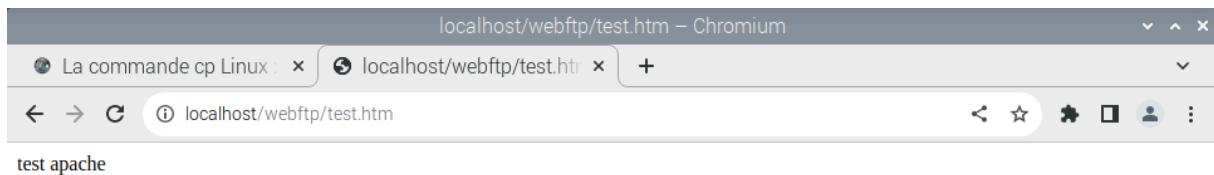
Enlever le fichier de l'URL (seulement le dossier). Qu'obtenez-vous ?

On obtient la liste des fichiers et dossiers du serveur



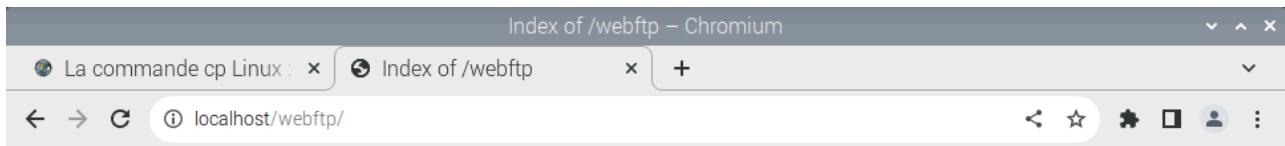
Renommer le fichier en **index.htm** et actualiser. Qu'obtenez-vous ?

La page est affichée directement



Si on veut lister (ou pas) le dossier webftp, il faut modifier le fichier de configuration :

```
<Directory /var/www/html/webftp>
    Options -Indexes
</Directory>
```



Index of /webftp

Name	Last modified	Size	Description
Parent Directory		-	
test.htm	2022-11-17 11:38	19	

Apache/2.4.54 (Raspbian) Server at localhost Port 80

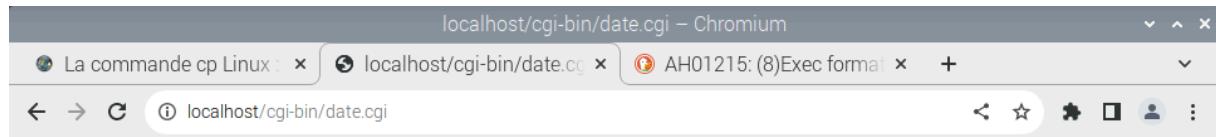
5) Script CGI simple

- 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"
```

Où placez-vous le script? Testez

On met le script dans /usr/lib/cgi-bin



Faut-il configurer quelque chose ?

Oui il faut utiliser la commande

```
sudo a2enmod cgi
```

et donner les droits au fichier date.cgi avec la commande :

```
sudo chmod 705 /.../date.cgi
```

- Modifier la configuration pour que toutes les pages suffixées par .sh soient considérées comme des requêtes CGI et exécutées.

On ajoute les règles suivantes pour autoriser l'exécution des scripts en .sh .py .cgi et .pl



J'ai aussi créeé un dossier scripts dans le dossier html pour éviter que les serveur aille chercher les fichiers dans `usr/....` Et donc tout centraliser

```
<Directory "/var/www/html/scripts">
    AddHandler cgi-script .cgi .py .sh .pl
    Options FollowSymLinks ExecCGI
    Options +ExecCGI
    AllowOverride None
</Directory>
```

- Ecrire un CGI qui permette d'afficher la liste des processus appartenant à un utilisateur donné par `http://localhost ou @IP/listeproc.sh ? user='utilisateur'`

```
#!/bin/bash
echo "Content-type : text/html"
echo
user='echo "$QUERY_STRING" | cut -f 2 -d "=" '
ps -u $user
```

The screenshot shows two windows. On the left is a terminal window titled 'alex@raspberrypi: /var/www/html/cgi-enabled' containing the source code for 'listeproc.sh'. The code is a simple CGI script that reads the QUERY_STRING environment variable, extracts the user name, and lists all processes belonging to that user using the 'ps' command. On the right is a Chromium browser window titled 'Liste des processus de alex – Chromium' showing the resulting HTML page with a list of processes for the user 'alex'.

```
alex@raspberrypi: /var/www/html/cgi-enabled
GNU nano 5.4
listeproc.sh

#!/bin/bash
# Récupération du nom d'utilisateur à partir de la variable d'environnement 'QUERY_STRING'
USER=$(echo "$QUERY_STRING" | grep -oP '(?<=user=)\w+')

# Vérification que le nom d'utilisateur a été fourni
if [ -z "$USER" ]; then
    # Si non, affichage d'un message d'erreur
    echo "Erreur : nom d'utilisateur non fourni"
    exit 1
fi

# Affichage de l'en-tête HTTP
echo "Content-type: text/html"
echo ""

# Affichage de la page HTML
echo "<html><head><title>Liste des processus de $USER</title></head><body>"
echo "<h1>Liste des processus de $USER</h1>"
echo "</pre>"

# Affichage de la liste des processus de l'utilisateur
echo "ps -u \"$USER\""
echo "</pre>"

# Fin de la page HTML
echo "</body></html>"
```

PID	TTY	TIME	CMD
733	?	00:00:00	systemd
734	?	00:00:00	(sd-pam)
748	?	00:00:00	pipewire
749	?	00:00:01	pulseaudio
750	?	00:00:00	lxsession
764	?	00:00:00	dbus-daemon
778	?	00:00:00	pipewire-media-session
835	tty1	00:00:00	gnome-terminal
836	?	00:00:00	ssh-agent
998	?	00:00:00	gvfsd
913	?	00:00:00	gvfsd-fuse
920	?	00:01:28	mutter
924	?	00:00:00	lxpolkit
927	?	00:00:15	lxpanel
929	?	00:00:26	pcmanfm
941	?	00:00:00	ssh-agent
943	?	00:00:00	apport.py
997	?	00:00:00	gvfs-smb2-vo
998	?	00:00:00	menu-cached
997	?	00:00:00	gvfs-afc-volume
1002	?	00:00:00	gvfs-ghostot2-vo
1006	?	00:00:00	gvfs-mtp-volume
1010	?	00:00:00	gvfs-goa-volume
1019	?	00:00:00	gvfsd-trash
1138	?	00:00:36	lxterminal
1140	pts/0	00:00:00	gnome-terminal
1410	?	00:03:15	chromium-browse
1432	?	00:00:00	chrome_crashpad
1436	?	00:00:00	chrome_crashpad
1439	?	00:00:00	chromium-browse
1440	?	00:00:00	chromium-browse
1442	?	00:00:00	chromium-browse
1462	?	00:01:45	chromium-browse
1466	?	00:00:45	chromium-browse
1474	?	00:00:00	chromium-browse
1491	?	00:00:00	chromium-browse
1564	?	00:00:47	chromium-browse
1790	?	00:00:12	chromium-browse
1884	?	00:00:01	gvfsd-metadata
1946	pts/1	00:00:00	bash
2394	?	00:00:08	geany
2404	pts/2	00:00:00	bash
2591	?	00:00:10	chromium-browse
2757	?	00:00:00	chromium-browse
2806	?	00:06:51	chromium-browse
2978	?	00:00:00	chromium-browse
3035	?	00:00:00	chromium-browse

- Ecrivez dans une page HTML avec un formulaire simple permettant de saisir l'utilisateur :

HTML

```
<form method="get" action="/cgi-bin/user.cgi">
```

```

<input type="text" name="user">

```

```
</form>
```

The terminal window shows the following code:

```

GNU nano 5.4          /var/www/html/listeproc.html
<html><head><title>Lister les Processus</title></head><body>
<h1> Entrer le nom d'un utilisateur</h1>
<form method = "get" action = "/cgi-enabled/listeproc.sh">
    <input type = "text" name = "user">
</form>

```

The browser window shows the title "Entrer le nom d'un utilisateur" and an empty input field.

6) Protection des accès

1. Créer un dossier « secret » dans `/var/www/html` et un fichier index.htm basique ; Configurer Apache pour que le dossier ne soit pas accessible du serveur localhost et pas d'un client.

[Quel est le code renvoyé par Apache à partir d'un client ?](#)

[Qu'observe-t-on dans les logs ?](#)

The browser title bar says "403 Forbidden – Chromium". The address bar shows "localhost/cgi-enable" and "localhost/secret". The main content area displays the word "Forbidden".

Forbidden

You don't have permission to access this resource.

Apache/2.4.54 (Raspbian) Server at localhost Port 80

[Wed Dec 14 10:58:53.245288 2022] [access_compat:error] [pid 4129] [client ::1:54914] AH01797: client denied by server configuration: /var/www/html/secret

2. On souhaite protéger l'accès à un dossier `/var/www/html/prive`

Il devra être accessible qu'à un ensemble limité de comptes apache (en non linux) à créer. Une requête sur ce dossier provoquera l'affichage d'une fenêtre d'authentification. La configuration doit contenir la directive : AccessFileName.htaccess

```

<Directory /var/www/html/prive>
    AllowOverride AuthConfig
    Options -Indexes
</Directory>

```

Exemples de directives d'un fichier .htaccess :

```

AuthType basic      #mot de passe en clair
authName alex       # texte d'accueil
AuthUserfile chemin/fichier #fichier contenant les mots de passes
Require valid-name

```

Ou Require user toto

Ou Require group prof

J'ai préféré tout mettre dans le fichier de configuration car il y moins de chance d'avoir des erreurs

Pour créer un compte Apache :

```
Htpasswd /etc/apache2/password toto
```

```
<Directory "/var/www/html/privee">
    AuthType Basic
    AuthName "Authentication Requise"
    AuthUserFile /etc/apache2/password
    Require valid-user
</Directory>
```

Ici ces lignes servent à indiquer que le dossier privée sera protégé par un mot de passe qu'il faut retrouver dans le fichier password, ce dernier peut être modifier sans que l'on est besoin de redémarrer les serveur à chaque fois.

Vérifier le fichier user

Créer un compte toto

Tester l'accès au dossier privé

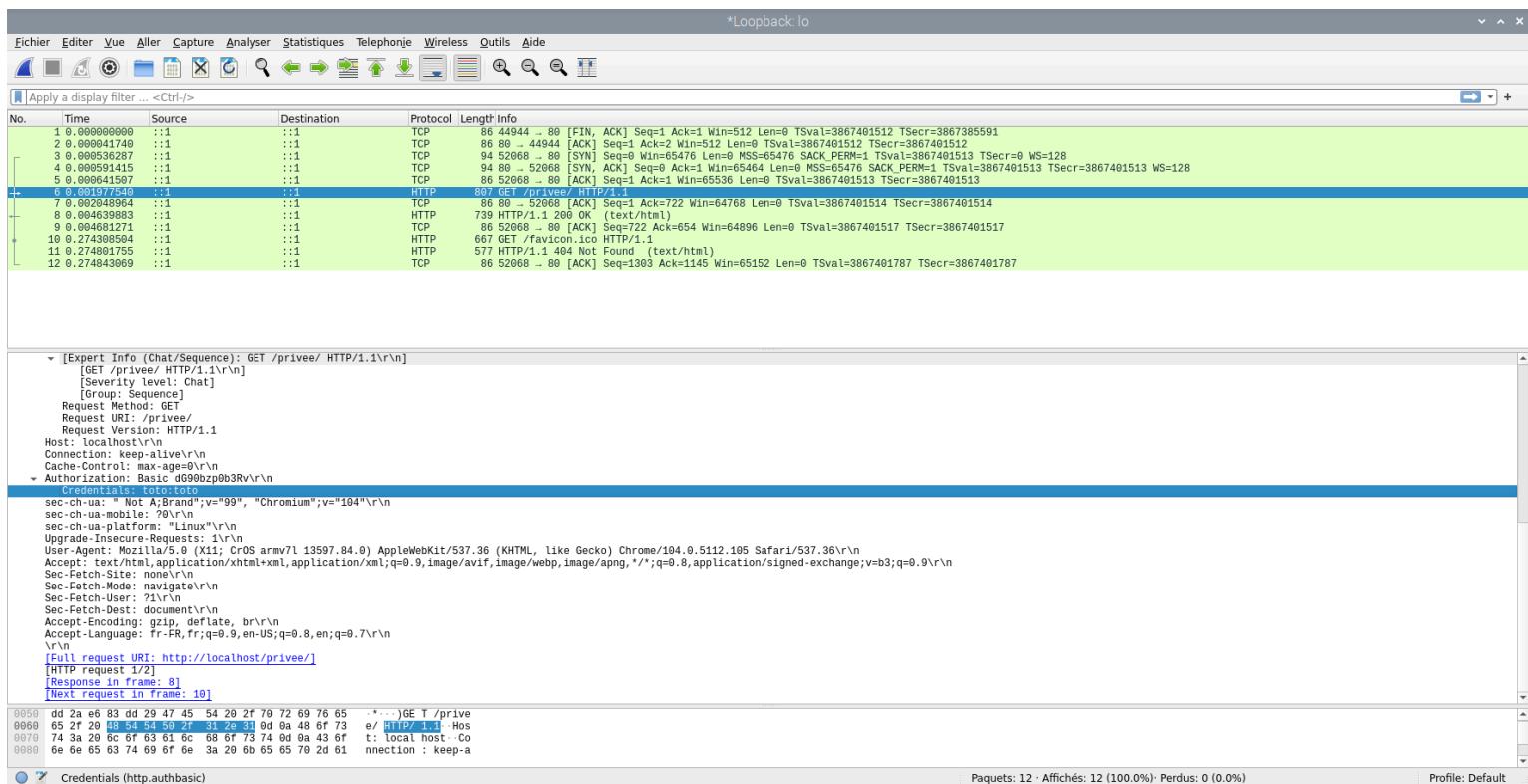
J'ai créé et utiliser l'utilisateur toto avec le mdp toto



Sur Wireshark :

- capturer un mot de passe

Pour lancer Wireshark il faut faire clic droit sur l'icône puis aller dans « Entrée de bureau » et mettre sudo devant « wireshark %f » car pour analyser le réseau il faut avoir les droits administrateur sur le raspberry. Et voici ce que ça donne :



Service httpd reload #recharge les directives

7) Serveurs virtuels

Lorsqu'une machine doit héberger plusieurs sites différents et même si elle ne dispose que d'une adresse IP, on utilise la technique des serveurs virtuels.

- Donner plusieurs noms à votre machine dans `/etc/hosts`

```

alex@raspberrypi: ~
Fichier Édition Onglets Aide
GNU nano 5.4          /etc/hosts *
127.0.0.1      Apache-Alexis
192.168.0.212    sites1.local

::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

127.0.1.1      raspberrypi
192.168.0.253   raspberrypi

```

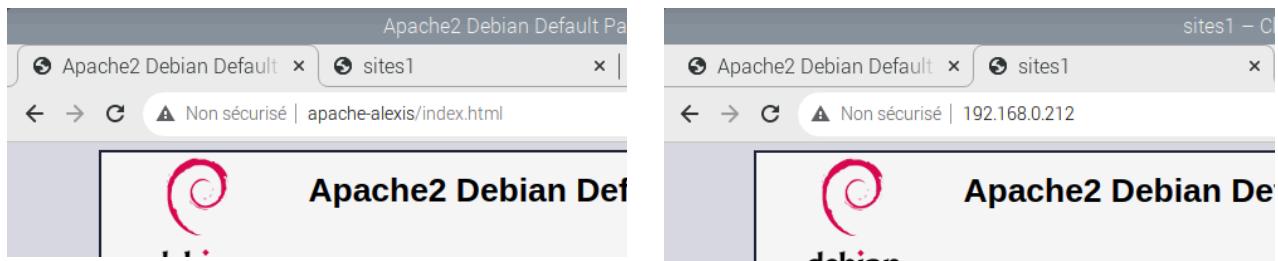
- Créer des hôtes virtuels dans des dossiers différents dans le fichiers .conf :

```

<VirtualHost toto>
    DocumentRoot /var/www/html/toto
    ServerName toto.org
</VirtualHost>

```

- Tester



Comme on peut le voir avec les deux capture d'écran j'ai nommé chaque site avec un nom différent apache-alexis pour celui par default et sites1 pour le serveur virtuel. Et voici les fichiers de configurations :

```
alex@raspberrypi: /etc/apache2/sites-available
Fichier Édition Onglets Aide
GNU nano 5.4          000-default.conf
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

```
alex@raspberrypi: /etc/apache2/sites-available
Fichier Édition Onglets Aide
GNU nano 5.4          sites1.conf
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    Servername sites1.local
    DocumentRoot /var/www/html/sites1

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- Comment séparer les logs de chaque serveur virtuel ?

Il suffit de changer les lignes errorlog et customlog avec celle suivante dans le fichier de configuration du serveur virtuel :

```
ErrorLog /var/log/apache2/sites-error.log
CustomLog /var/log/apache2/sites1-access.log combined
```

```
alex@raspberrypi:/var/log/apache2 $ ls
access.log      access.log.4.gz  error.log.3.gz      sites1-access.log
access.log.1    error.log       error.log.4.gz      sites1-error.log
access.log.2.gz  error.log.1    other_vhosts_access.log
access.log.3.gz  error.log.2.gz  other_vhosts_access.log.1
alex@raspberrypi:/var/log/apache2 $
```

8) HTTPS : le http sécurisé

par défaut, le protocole http port 80 non sécurisé.

HTTPS port 443 sécurisé SSH

Activer le module SSL et le site default-ssl

```
a2enmod ssl
```

```
a2ensite default-ssl  
service apache2 restart
```

- Vérifier que le site fonctionne en HTTPS et contrôler le chiffrement avec wireshark

Après avoir lancer wireshark on ne voit plus aucune requête http passer donc on ne pourra plus voir le mot de passe en clair .

- On supprime l'accès au http

Pour cela il faut se rendre dans le fichier `ports.conf` puis mettre un `#` devant la ligne `Listen 80`

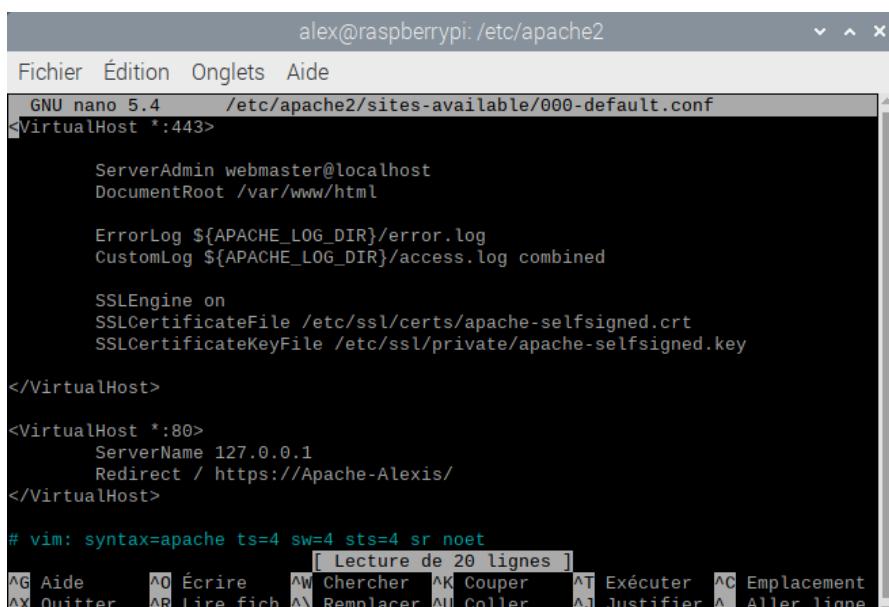
En essayant de se connecter avec le port 80 le navigateur affiche une erreur.

- Trouver le certificat SSL par défaut ([sites-available](#))

Le certificat par défaut se trouve dans `/etc/ssl/certs/apache-selfsigned.crt`

Et la clé dans `/etc/ssl/private/apache-selfsigned.key`

- Rediriger automatiquement toutes les requêtes HTTP dans le fichier `sites-available/default` :

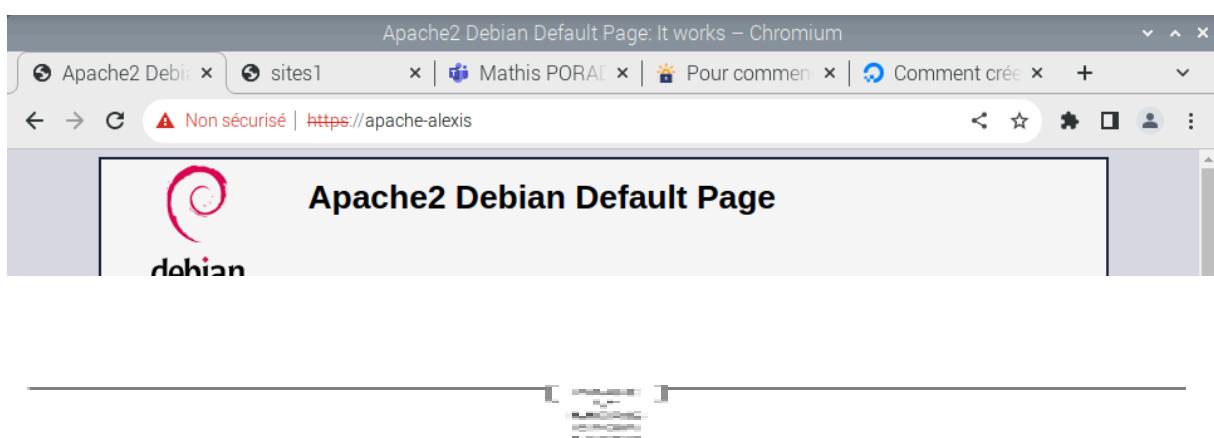


```
alex@raspberrypi: /etc/apache2  
Fichier Édition Onglets Aide  
GNU nano 5.4          /etc/apache2/sites-available/000-default.conf  
<VirtualHost *:443>  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/html  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
    SSLEngine on  
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt  
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key  
  
</VirtualHost>  
  
<VirtualHost *:80>  
    ServerName 127.0.0.1  
    Redirect / https://Apache-Alexis/  
</VirtualHost>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet [ Lecture de 20 lignes ]  
AG Aide      AÉ Écrire   AW Chercher  ^K Couper   ^T Exécuter  AÉ Emplacement  
AX Quitter   AR Lire fich.  ^R Remplacer  ^U Coller   ^J Justifier  A Aller ligne
```

La ligne `redirect` permet de rediriger toutes les requêtes http vers HTTPS

- Vérifier que toutes les requêtes http sont redirigées en HTTPS

On peut voir que pour accéder au site on utilise HTTPS alors que j'ai tapé `apache-alexis :80` dans le navigateur.



En revanche le certificat par défaut n'est reconnu que par la machine local donc pour demander un certificat on peut s'adresser à Lets Encrypt qui peut nous en fournir un gratuitement et qui nous sera utile si on veut ensuite mettre notre site en ligne.

V. Utilisation de MySql

1) Introduction

MySql est un SGBD (Système de Gestion de Bases de Données) libre, complémentaire de PHP pour créer des sites web dynamiques. Les informations du site seront donc toujours à jour des modifications dans la base de données. Autres SGBD :

- Postgresql, libre (recommandé pour de grosse bases)
- Oracle et Microsoftsql sont propriétaires

Tous exploitent le language SQL (Strutured Query Language).

Suite au rachat de MySql par Sun MicroSystem puis Oracle en 2009, le projet libre MySql s'est transformé en MariaDB, soutenue par une fondation indépendante.

2) Installation

Déjà faite précédemment et testée avec la commande PHP :`phpinfo()`

3) Fonctionnement

MySql gère les fichiers de la BdD, prend en charge la protection et la sécurité, et fournit de interfaces de programmations (dont une en PHP) facilitant l'accès aux données.

Architecture client/serveur. Le processus mysqld est le serveur de MySql qui est le seul à accéder aux données.

4) L'architecture

Commande déjà vue : `mysql -u root -p`

Commande utile : `help`

Exemple :

```
CREATE DATABASE Films ;
GRANT ALL PRIVILEGES ON Films.* TO adminFilms@localhost
IDENTIFIED BY 'motdepasse';
```

Pour se déconnecter : `exit`



On se reconnecte comme adminFilms : `mysql -u adminFilms -p Films`

Création d'une table :

```
CREATE TABLE FilmSimple(  
    titre VARCHAR (30),  
    annee INTEGER  
)
```

Affichage de la structure d'une table : `DESC FilmSimple ;`

Suppression d'une table : `DROP TABLE FilmSimple ;`

Pour remplir une table :

```
INSERT INTO FilmSimple  
(titre, annee)  
VALUES  
(‘Pulp fiction’, 1994),  
(‘Alien’,1979),  
(‘Titanic’,1997),  
(‘Iron man’,2008) ;
```

Interrogation par de requêtes :

```
SELECT titre FROM FilmSimple WHERE annee=’1997’ ;
```

Comment sélectionner tous les films après 1980 ?

```
SELECT titre FROM FilmSimple WHERE annee>=’1980’ ;
```

```

alex@raspberrypi: / 
Fichier Édition Onglets Aide

MariaDB [Films]> INSERT INTO FilmSimple (titre,annee) VALUES ('Pulp fiction',1994) ('Alien',1979) ('Titanic',1997) ('Iron man',2008);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '('Alien',1979) ('Titanic',1997) ('Iron man',2008)' at line 1
MariaDB [Films]> INSERT INTO FilmSimple (titre,annee) VALUES ('Pulp fiction',1994), ('Alien',1979), ('Titanic',1997), ('Iron man',2008);
Query OK, 4 rows affected (0.020 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [Films]> DESC FilmSimple
-> ;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| titre | varchar(30) | YES |     | NULL    |       |
| annee | int(11)    | YES |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.004 sec)

MariaDB [Films]> SELECT titre FROM FilmSimple WHERE annee='1997'
-> ;
+-----+
| titre  |
+-----+
| Titanic |
+-----+
1 row in set (0.001 sec)

MariaDB [Films]> SELECT titre FROM FilmSimple WHERE annee>='1997'
-> ;
+-----+
| titre  |
+-----+
| Titanic |
| Iron man |
+-----+
2 rows in set (0.001 sec)

MariaDB [Films]> █

```

[Compléter la table avec PHPMyAdmin ...](#)

Avec la commande suivante je supprime les films Transformers 2 :

DELETE FROM 'SimpleFilm' WHERE titre='Transformers 2';

5) Utilisation de PHP

Avec tout ce que l'on a pu voir depuis le début de ce TP nous sommes capable d'utiliser le serveur Apache et sa base de données, on va donc maintenant utiliser PHP pour connecter la BDD à une page web. Pour commencer je me connecte sur PHPMyAdmin pour ajouter une colonne « id » dans ma table film simple car j'en aurais besoin pour plus tard, je le fais avec la commande suivante :

ALTER TABLE FilmSimple ADD id INT NOT NULL AUTO_INCREMENT PRIMARY KEY;

Cette ligne permet d'ajouter la colonne id en lui disant de s'auto incrémenter et not null permet que le compte commence à partir de 1 et non 0.

Il ne reste plus qu'à coder une pages web pour utiliser cette base de données.



```
index.php > html > body > main > em
1  <?php
2
3  require_once('request.php');
4
5  ?>
6
7  <!DOCTYPE html>
8  <html lang="fr">
9
10 <head>
11   <meta charset="UTF-8">
12   <meta http-equiv="X-UA-Compatible" content="IE=edge">
13   <meta name="viewport" content="width=device-width, initial-scale=1.0">
14   <title>My First APP PHP</title>
15 </head>
16
17 <body>
18   <main>
19     <?php foreach(findAll() as $FilmSimple) : ?>
20       <h2><?= $FilmSimple['titre'] ?></h2>
21       <em><?= $FilmSimple['annee'] ?></em>
22
23
24     <?php endforeach; ?>
25   </main>
26 </body>
27 </html>
28
29
```

Ce fichier index.php sert de page d'affichage pour ma base de données, le premier paragraphe PHP sert à connecter ce fichier au fichier request.php, la deuxième partie en php sert à afficher les éléments titre et année qui sont dans un tableau stocké dans la variable FilmSimple.

```

request.php > ...
1  <?php
2
3  require_once 'mysql.php';
4
5  function findAll(): array
6  {
7
8      global $db;
9
10     $query = "SELECT * FROM FilmSimple";
11     $sqlStatement = $db->prepare($query); # prepare la requete
12     $sqlStatement->execute(); # execute la requete
13
14     return $sqlStatement->fetchAll(); # recuperे les résultats
15 }
16 /*cette fonction permet de récupérer tous les films de la base de données et de les stocker
17 dans un tableau qui sera renvoyer ensuite.
18 */
19 function findById($id): array
20 {
21
22     global $db;
23
24     $query = "SELECT * FROM FilmSimple WHERE id = :id";
25     $sqlStatement = $db->prepare($query);
26     $sqlStatement->execute([
27         'id' => $id,
28     ]);
29
30     return $sqlStatement->fetchAll();
31 }
32 /*cette fonction permet de recuperer un film spécifique en utilisant l'id passé en parametre
33 et de stocker les données dans un tableau qui sera renvoyer*/
34

```

Ici on va chercher les informations dans la base de données toutes les requêtes sont expliquer dans ce code, au début de celui-ci on fait appel au fichier mysql.php .

```

mysql.php > ...
1  <?php
2
3  try {
4      $db = new PDO('mysql:host=apache-alexis;dbname=Films;charset=utf8mb4', 'adminFilms', 'qzds');
5      $db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
6  } catch (PDOException $e) [
7      die($e->getMessage());
8 ]

```

Ce fichier sert à se connecter à la base de données, c'est un fichier qu'il faut bien garder car il y a le mot de passe et l'identifiant pour la BBD, le code sert à se connecter et à retourner une erreur si la connexion échoue pour qu'on puis découvrir la cause de l'échec.

Voici le résultat de ces codes.



Pulp fiction

1994

Alien

1979

Titanic

1997

Iron man

2008

Transformers

2007

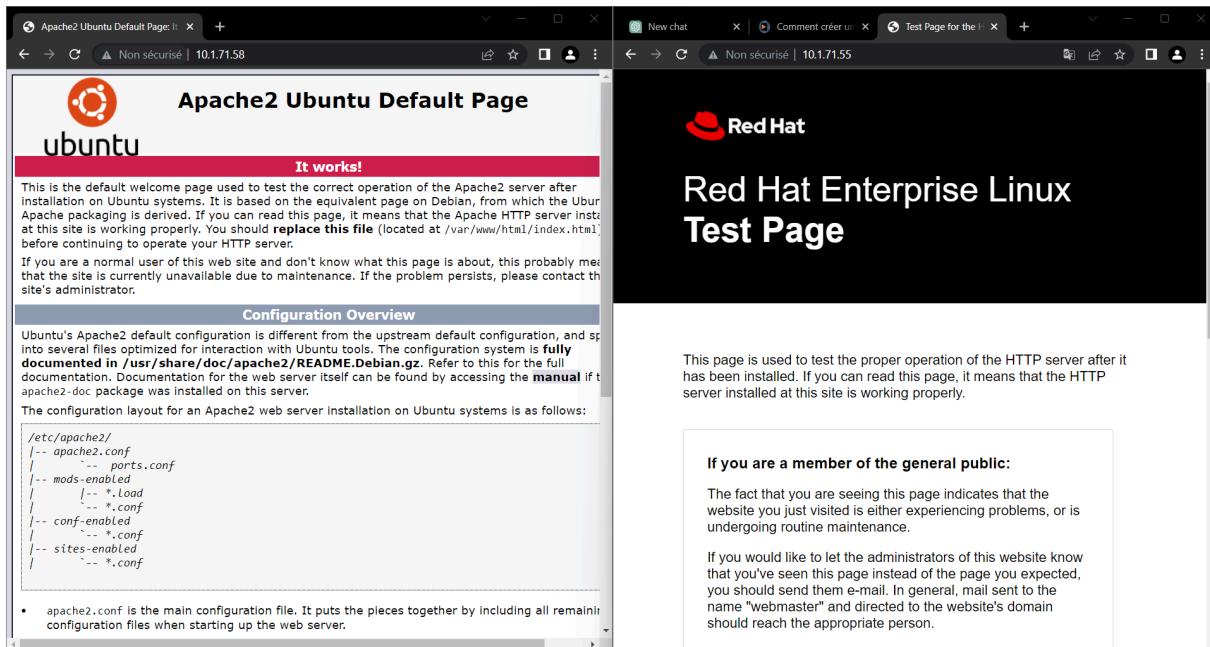
Transformers

2007

VI. Cluster serveur apache

Pour aller un peu plus loin je vais créer un cluster de serveur apache, pour cela je recrée deux VM ou j'installe apache, une VM Red Hat qui sera le serveur principale et une Zorin OS qui sera le secondaire.

Je passe sur l'installation et la configuration d'Apache car on l'a déjà fait.



Mes deux serveur sont maintenant monté et fonctionnels. Il ne reste plus qu'à créer le cluster.

1) HAProxy

J' ai choisi d'utiliser HAProxy pour créer le cluster.

HAProxy (High Availability Proxy) est un logiciel de balance de charge open-source qui permet de répartir les requêtes entrantes entre plusieurs serveurs. Il est souvent utilisé pour gérer les clusters de serveurs web et pour assurer la haute disponibilité d'un service en répartissant les requêtes entre plusieurs serveurs.

Il a les fonctionnalités suivantes:

- Il permet de répartir les requêtes entrantes entre plusieurs serveurs en fonction de critères tels que le poids, la disponibilité ou les capacités des serveurs.
- Il permet de surveiller les performances et la disponibilité des serveurs pour détecter les éventuels problèmes et basculer automatiquement les requêtes vers des serveurs de secours en cas de panne.
- Il prend en charge différents protocoles, tels que HTTP, HTTPS, TCP et SMTP, ce qui en fait un outil polyvalent pour gérer les clusters de serveurs.
- Il permet de compresser les données transmises pour optimiser les performances.
- Il permet de mettre en place des règles de sécurité pour protéger les serveurs.

En résumé, HAProxy est un outil utile pour gérer les clusters de serveurs et assurer la haute disponibilité d'un service en répartissant les requêtes entre plusieurs serveurs et en surveillant les performances et la disponibilité des serveurs.

2) Installation

Pour l'installer il faut avoir une autre machine qui sert de serveur, je crée donc une nouvelle VM avec Linux Mint OS basé sur Debian, je l'utilise car cela me permet de découvrir un nouvel OS.

Pour installer HAProxy on utilise la commande :



```
sudo apt install haproxy
```

3) Configuration

Il faut ensuite modifier le fichier de configuration qui se trouve sous `/etc/haproxy/haproxy.cfg`, conseil de faire une copie du fichier.

```
listen http-in
bind *:80
mode http
balance roundrobin
option forwardfor
option http-server-close
server server1 192.168.1.100:80 check
server server2 192.168.1.101:80 check
```

La section "`listen http-in`" de la configuration d'HAProxy définit un service d'écoute pour les requêtes HTTP entrantes sur le port 80. Les paramètres de cette section incluent :

- **bind** : Cette ligne spécifie l'adresse IP et le port sur lequel le service d'écoute doit écouter les requêtes entrantes. Dans ce cas, "*" signifie que le service d'écoute écoutera toutes les adresses IP disponibles sur le port 80.
- **mode** : Cette ligne spécifie le mode de fonctionnement du service d'écoute. Dans ce cas, "http" indique que le service d'écoute traitera les requêtes HTTP.
- **balance** : Cette ligne spécifie la méthode utilisée pour répartir les requêtes entre les serveurs. Dans ce cas, "roundrobin" indique que les requêtes seront réparties entre les serveurs de manière équilibrée en utilisant la méthode de rotation.
- **option forwardfor** : Cette ligne permet à HAProxy d'ajouter un en-tête "X-Forwarded-For" aux requêtes envoyées aux serveurs, indiquant l'adresse IP de l'origine de la requête.
- **option http-server-close** : Cette ligne permet à HAProxy de fermer la connexion après chaque requête, ce qui peut améliorer les performances pour certaines configurations.
- **server** : Ces lignes définissent les serveurs qui font partie du cluster et les paramètres de configuration de chaque serveur. Dans ce cas, "server1" et "server2" sont les noms de deux serveurs avec des adresses IP respectives "192.168.1.100" et "192.168.1.101" sur le port 80. La commande "check" permet à HAProxy de vérifier si le serveur est disponible ou non.

On peut remplacer le paramètre `listen` par **frontend** ou **backend** :

- **frontend** : Cette section définit les paramètres de configuration pour les différents services d'entrée d'HAProxy, comme les ports d'écoute et les règles de répartition de charge.
- **backend** : Cette section définit les paramètres de configuration pour les différents services de sortie d'HAProxy, comme les serveurs et les options de répartition de charge



J'ajoute également une surveillance des performances dans la configuration pour être sûr qu'il fonctionne toujours et gérer d'éventuel problème. Il faut la mettre dans la partie **defaults**

```
stats enable
stats auth admin:password
stats uri /haproxy?stats
monitor-uri /health
log global
log 127.0.0.1 local0
```

J'ai ensuite réécrit le fichier de configuration pour utiliser les paramètres frontend et backend car c'est plus propre et professionnel .

```
frontend http-in
  bind *:80
  mode http
  default_backend servers

backend servers
  balance roundrobin
  option forwardfor
  option http-server-close
  server server1 10.1.71.58:80 check weight 10
  server server2 10.1.71.55:80 check weight 5
```

ATTENTION : le moindre espace peu engendré une erreur par exemple j'avais un espace a la fin de la ligne `default_backend servers` et pas a la fin de `backend servers` ce qui entraînait une erreur.

J'ai seulement ajouté les arguments **weight** pour signifier que le premier serveur est plus important que le deuxième et donc a la priorité.

Pour voir la cause de l'erreur on utilise la commande :

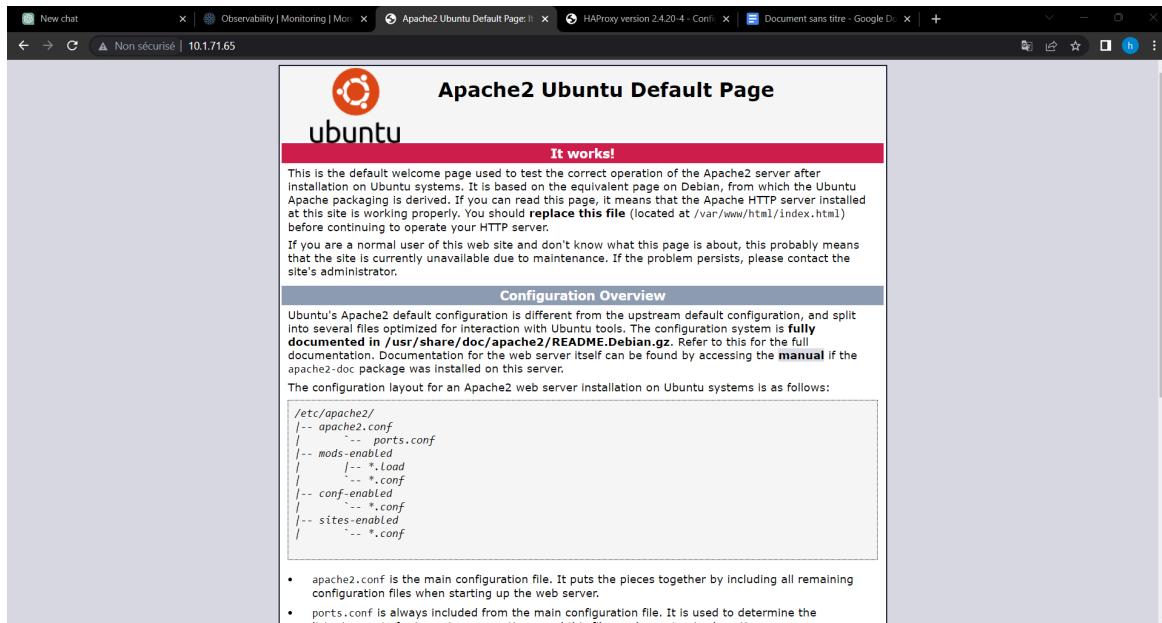
```
sudo tail -f /var/log/haproxy.log
```

```
alex@alex-VirtualBox:~$ sudo tail -f /var/log/haproxy.log
Jan 20 14:14:37 alex-VirtualBox haproxy[4313]: [ALERT]      (4313) : Proxy 'http-in': unable to find required default_backend: 'serveur '
Jan 20 14:14:37 alex-VirtualBox haproxy[4313]: [ALERT]      (4313) : Fatal errors found in configuration.
Jan 20 14:14:37 alex-VirtualBox haproxy[4315]: [ALERT]      (4315) : Proxy 'http-in': unable to find required default_backend: 'serveur '
Jan 20 14:14:37 alex-VirtualBox haproxy[4315]: [ALERT]      (4315) : Fatal errors found in configuration.
Jan 20 14:14:38 alex-VirtualBox haproxy[4317]: [ALERT]      (4317) : Proxy 'http-in': unable to find required default_backend: 'serveur '
Jan 20 14:14:38 alex-VirtualBox haproxy[4317]: [ALERT]      (4317) : Fatal errors found in configuration.
Jan 20 14:14:38 alex-VirtualBox haproxy[4319]: [ALERT]      (4319) : Proxy 'http-in': unable to find required default_backend: 'serveur '
Jan 20 14:14:38 alex-VirtualBox haproxy[4319]: [ALERT]      (4319) : Fatal errors found in configuration.
Jan 20 14:14:38 alex-VirtualBox haproxy[4321]: [ALERT]      (4321) : Proxy 'http-in': unable to find required default_backend: 'serveur '
Jan 20 14:14:38 alex-VirtualBox haproxy[4321]: [ALERT]      (4321) : Fatal errors found in configuration.
```

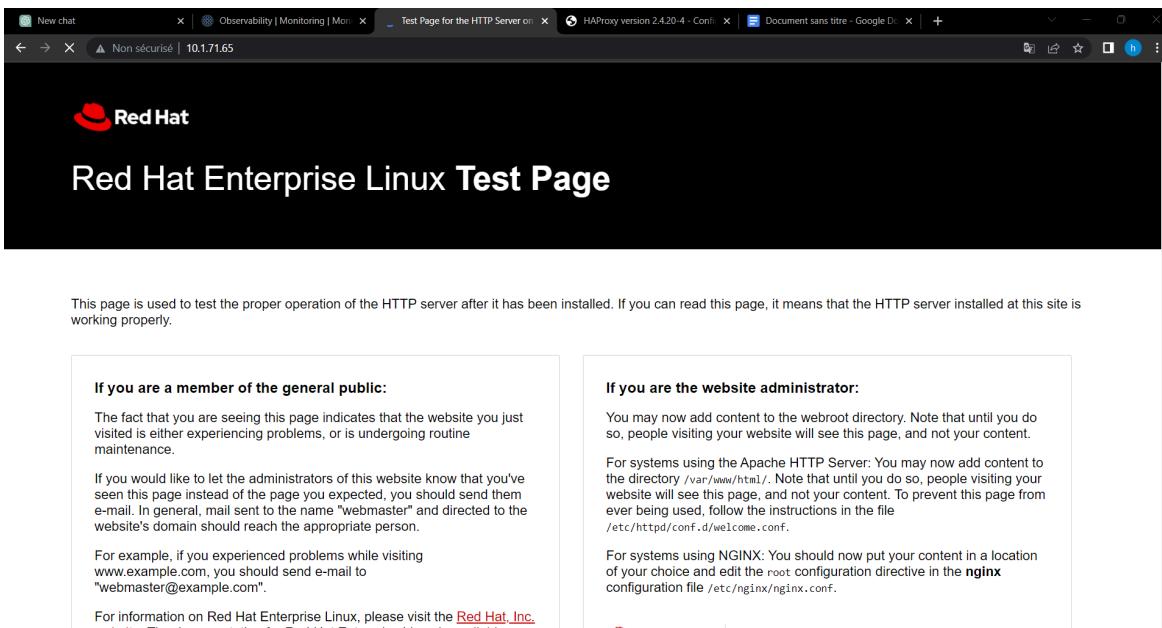
On voit dans l'encadrer orange il y a un espace entre `serveur` et ' ce qui a créé une erreur.

4) Test

IL est maintenant temps de tester ma configuration, pour accéder au serveur auparavant j'utilisais les @IP **10.1.71.58** et **.55** maintenant je me connecte sur la **.65** qui est mon serveur HAProxy.



J'arrive sur cette page qui est celle de mon serveur principale, puis lorsque j'éteint la VM de ce serveur, HAProxy vous me rediriger vers le deuxième alors que j'utilise la même ip pour l'atteindre.

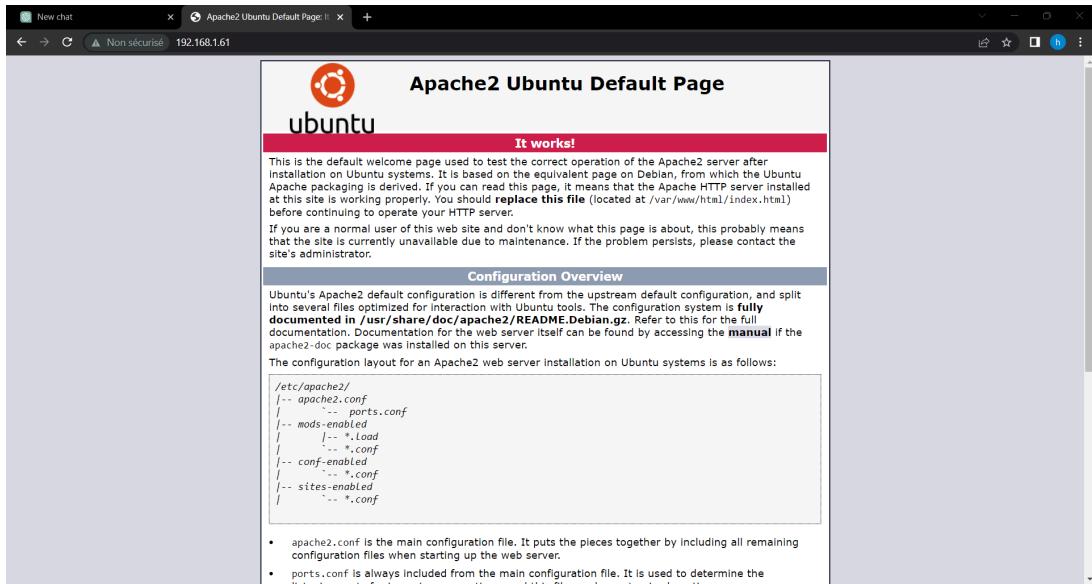


On remarque que le serveur Haproxy a bien basculer sur le deuxième apache, si je rallume ma VM et que je recharge la page alors puisqu'il a la priorité dans la conf le serveur devrait m'afficher la page de mon premier apache.

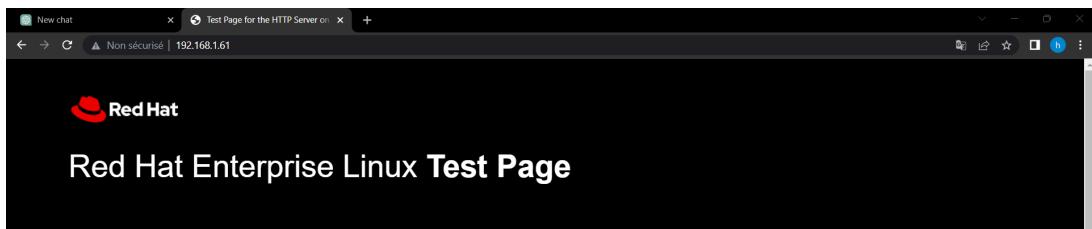
Pendant le test les IP ont changé car j'ai suis chez moi et non au charmilles.

Serveur 1 : **192.168.1.26** / serveur 2 **192.168.1.97** / HAProxy **192.168.1.61**

Je met l'IP de HAProxy dans le navigateur.



Puis j'éteint la VM serveur 1

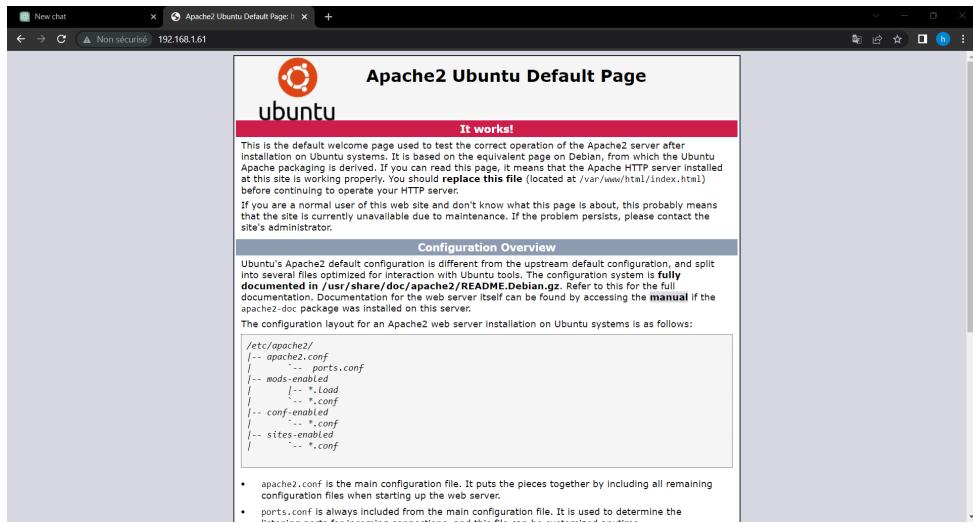


This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.

If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
For example, if you experienced problems while visiting www.example.com, you should send e-mail to webmaster@example.com.
For information on Red Hat Enterprise Linux, please visit the [Red Hat, Inc.](#)

If you are the website administrator:
You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.
For systems using the Apache HTTP Server: You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.
For systems using NGINX: You should now put your content in a location of your choice and edit the `root` configuration directive in the `nginx` configuration file `/etc/nginx/nginx.conf`.

La page a changé donc le deuxième serveur a pris la relève maintenant je vais relancer la VM serveur 1 et le serveur devrait reprendre le main.



Le HAProxy a bien choisi le serveur 1 qui est prioritaire, la configuration marche correctement.

Je vais regarder les pages de diagnostique que j'ai crée grâce à certain paramètres. La première sert à m'avertir si tout les services marche correctement. Il suffit de rajouter **/health** dans l'URL



Ensuite en ajoutant **/haproxy?stats** dans l'URL on obtient un page qui affiche bien plus d'information.

The screenshot shows the HAProxy Statistics Report for pid 1693. The report displays various metrics and configurations for the HAProxy instance. At the top, it shows the HAProxy version (2.4.18-0ubuntu1, released 2022/08/25) and a general process information table. Below this, there are two main sections: "http-in" and "serveur".

General process information:

http-in															serveur														
Queue			Session rate			Sessions			Bytes			Denied			Errors			Warnings			Server								
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtm	Thrde
Frontend			0	2	-	2	4	-	262 124	12		4 908	34 716	0	0	0	3				OPEN								

serveur

serveur																													
Queue			Session rate			Sessions			Bytes			Denied			Errors			Warnings			Server								
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtm	Thrde
server1	0	0	-	0	2	0	1	-	7	4	5m21s	1 739	10 571	0	1	0	3	0			6m7s UP	L4OK in 1ms	10/10	Y	-	3	1	2m24s	-
server2	0	0	-	0	4	0	3	-	6	6	5m22s	2 320	23 732	0	0	0	0	0			13m24s UP	L4OK in 1ms	5/5	Y	-	0	0	0s	-
Backend	0	0	-	0	4	0	4	-	26 213	10	5m21s	4 059	34 303	0	0	1	0	3	0		13m24s UP		15/15	2	0			0	0s

Je vais utiliser un outil pour générer une charge de trafics qui s'appelle siege. Voici la commande pour installer siege :

```
sudo apt install siege
```

Puis pour simuler un charge de trafics avec 10 personnes qui effectue des requêtes pendant 30 seconde.

```
sudo siege -c10 -t30S http://example.com
```

HAProxy version 2.4.18-0ubuntu1, released 2022/08/25

Statistics Report for pid 1693

> General process information

General process information																		Display option:	External resources:										
<p>pid = 1693 (process #1, nbproc = 1, nbthread = 1) uptime = 0d 0h26m07s system limits: memmax = unlimited, ulimit-n = 524287 maxsock = 524287, maxconn = 262142, maxpipes = 0 current connis = 2, current pipes = 0/0, conn rate = 0/sec, bit rate = 161.767 kbps Running tasks: 0/14, idle = 99 %</p> <p>Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.</p>																		Scope: <input type="text"/>	Hide DOWN servers	Refresh now	CSV export	JSON export (schema)							
http-in	Queue	Cur	Max	Limit	Session	rate	Cur	Max	Limit	Sessions	Total	LbTot	Last	In	Bytes	Denied	Errors	Warnings	Status	Server									
Frontend		0	571	-	2	10	262	124	13 795		2 065 419	49 472 517	0	0	6	Req	Conn	Retr	Redis	OPEN	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrile	
serveur	Queue	Cur	Max	Limit	Session	rate	Cur	Max	Limit	Sessions	Total	LbTot	Last	In	Bytes	Denied	Errors	Warnings	Status	Server									
server1	0	0	-	0	380		0	10	-	9 193	9 190	23s	1 373 846	31 810 426	0	1	0	3	0	10m44s UP	L40K in 2ms	10/10	Y	-	3	1	2m24s	-	
server2	0	0	-	0	190		0	4	-	4 598	4 598	23s	688 768	17 590 788	0	0	0	0	0	26m1s UP	L40K in 2ms	5/5	Y	-	0	0	0s	-	
Backend	0	0	-	0	571		0	10	26 213	13 788	13 788	23s	2 062 614	49 401 214	0	0	1	0	3	26m1s UP		15/15	2	0			0	0s	

HAProxy version 2.4.18-0ubuntu1, released 2022/08/25

Statistics Report for pid 1693

> General process information

General process information																		Display option:	External resources:										
<p>pid = 1693 (process #1, nbproc = 1, nbthread = 1) uptime = 0d 0h27m37s system limits: memmax = unlimited, ulimit-n = 524287 maxsock = 524287, maxconn = 262142, maxpipes = 0 current connis = 2, current pipes = 0/0, conn rate = 2/sec, bit rate = 0.000 kbps Running tasks: 0/14, idle = 100 %</p> <p>Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.</p>																		Scope: <input type="text"/>	Hide DOWN servers	Refresh now	CSV export	JSON export (schema)							
http-in	Queue	Cur	Max	Limit	Session	rate	Cur	Max	Limit	Sessions	Total	LbTot	Last	In	Bytes	Denied	Errors	Warnings	Status	Server									
Frontend		2	571	-	2	20	262	124	28 138		4 208 211	100 814 524	0	0	10	Req	Conn	Retr	Redis	OPEN	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrile	
serveur	Queue	Cur	Max	Limit	Session	rate	Cur	Max	Limit	Sessions	Total	LbTot	Last	In	Bytes	Denied	Errors	Warnings	Status	Server									
server1	0	0	-	0	382		0	20	-	18 752	18 749	7s	2 801 570	64 800 801	0	1	0	3	0	20m20s UP	L40K in 2ms	10/10	Y	-	3	1	2m24s	-	
server2	0	0	-	0	191		0	6	-	9 377	9 377	7s	1 403 347	35 824 461	0	0	0	0	0	27m37s UP	L40K in 4ms	5/5	Y	-	0	0	0s	-	
Backend	0	0	-	0	572		0	20	26 213	28 126	28 126	7s	4 204 917	100 725 262	0	0	1	0	3	27m37s UP		15/15	2	0			0	0s	

Ses deux tableau ont été générés après la commande pour simuler le trafics la première avec 10 personnes pendant 30s et la deuxième avec 20 personnes.

On peut également mettre HAProxy en cluster pour plus de sécurité, mais je vais m'arrêter la pour le moment.