

## Documentation Technique :

### Mise en place d'un serveur LAMP

<b>Savoirs</b>	<ul style="list-style-type: none"><li>• Principes d'architecture d'un service (architecture du service Web) ;</li><li>• Protocoles associés au service WEB : http, https ;</li><li>• Chiffrement et certificat ;</li><li>• Le protocole HTTPS et service Web sécurisé ;</li></ul>
<b>Savoir-faire</b>	<ul style="list-style-type: none"><li>• Caractériser un service et le serveur associé</li><li>• Exploiter les fonctions de base d'un langage de commandes</li><li>• Installer, configurer et administrer un service</li><li>• Mettre en œuvre un protocole sécurisé associé à un service</li></ul>
<b>Outils</b>	<p>Serveur Linux Debian.</p> <p><b>Serveurs/service</b> : apache2</p> <p><b>Clients</b> : navigateur web sur STA Linux, Windows ou autre système.</p> <p>Outils de capture et d'analyse de trames (Wireshark)</p>
<b>Mots-clés</b>	Service, dns, url, web, http, https, chiffrement, certificat

## Etape 1 : Mise en place d'un serveur WEB

### Installation du serveur Web apache2 :

1. On installe les paquets `apache2` et `libapache2-mod-php` avec la commande :

**Apt update + apt upgrade**

**Apt install apache2 libapache2-mod-php**

2. Avant de commencer à modifier la configuration, nous faisons une sauvegarde du répertoire contenant la configuration d'apache : `/etc/apache2` :

`cp -r /etc/apach2 /etc/apache2-copie`

La **configuration d'Apache** (comme celle de la plupart des services sur Linux) est **modulaire** c'est à dire que le fichier principal de configuration inclut des fichiers de configuration « externalisés » ayant chacun un rôle bien précis ; de plus, lors d'une mise à jour ce fichier pourrait être modifié. Il est donc fortement **déconseillé de le modifier** mais plutôt d'intervenir sur les fichiers inclus dans « **apache2.conf** ».

A chaque modification ou ajout de fichier, on peut vérifier la syntaxe en exécutant la commande :

```
apache2ctl configtest ou apache2ctl -t
```

```
LAMP [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Activités Terminal 23 janv. 17:16
mamadou@debian: ~
ethertypes          localtime          rc5.d              wpa_suppli
firefox-esr         logcheck          rc6.d              X11
fonts               login.defs        rcS.d              xattr.conf
fstab               logrotate.conf    reportbug.conf     xdg
fuse.conf           logrotate.d       resolv.conf        xml
root@debian:/etc#
root@debian:/etc# sudo apache2ctl -t
AH00526: Syntax error on line 5 of /etc/apache2/ports.conf:
Cannot define multiple Listeners on the same IP:port
Action '-t' failed.
The Apache error log may have more information.
root@debian:/etc# nano /etc/apache2/ports.conf
root@debian:/etc# nano /etc/apache2/ports.conf
root@debian:/etc# apache2ctl configtest
bash: apache2ctl : commande introuvable
root@debian:/etc# sudo apache2ctl configtest
AH00526: Syntax error on line 5 of /etc/apache2/ports.conf:
Cannot define multiple Listeners on the same IP:port
Action 'configtest' failed.
The Apache error log may have more information.
root@debian:/etc# nano /etc/apache2/ports.conf
root@debian:/etc# sudo apache2ctl configtest
Syntax OK
root@debian:/etc#
```

[apache2ctl status](#)

OU

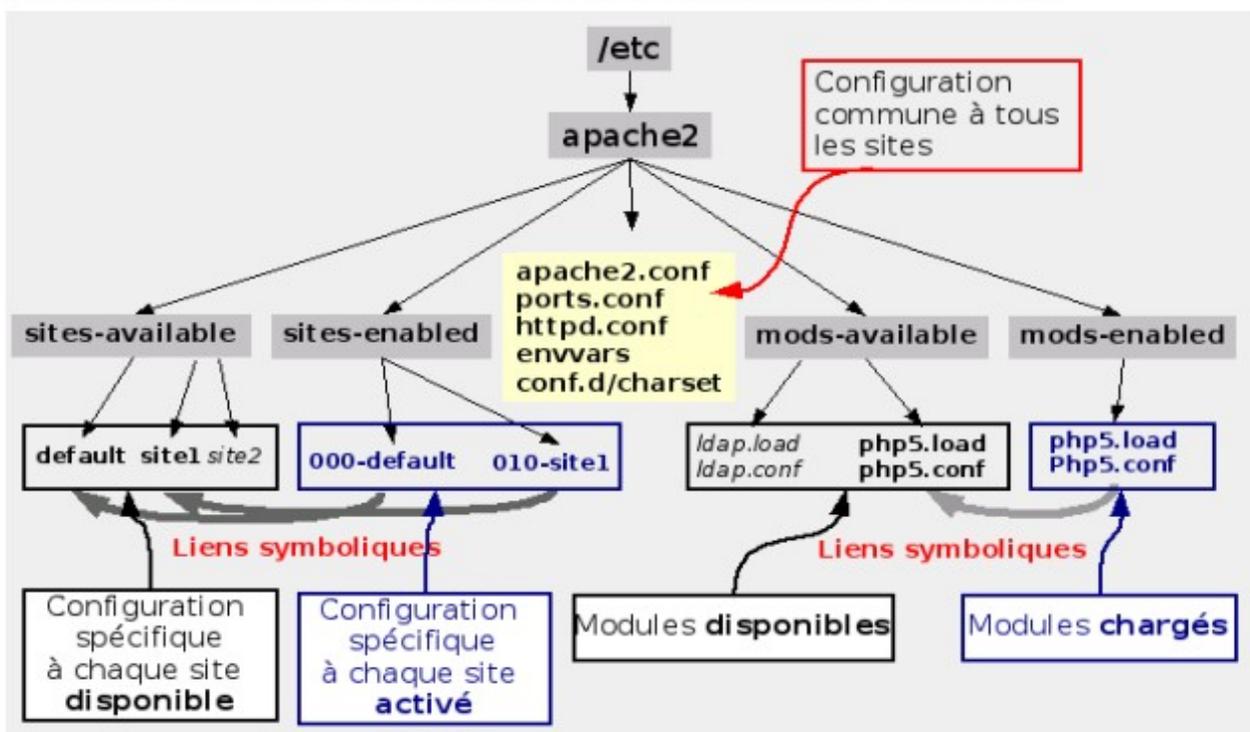
[systemctl status apache2](#)

Cette commande affiche l'état de service apache2

## Etape 2 : Création des Hôtes Virtuels (VHost)

Pour tester le bon fonctionnement du serveur Web, nous allons créer quelques hôtes virtuels sur le serveur. Pour cela, nous allons nous servir sur le site par défaut du serveur.

Apache2 dispose de plusieurs fichiers de configuration dans `/etc/apache2`.



Les directives minimales pour configurer un hôte virtuel sont les suivantes :

<pre>&lt;VirtualHost *:80&gt;   ServerName nom_dns   DocumentRoot     chemin_racine_web_du_site   ... &lt;/VirtualHost&gt;</pre>	<p><b>ServerName</b> nom_dns indique le nom pleinement qualifié à partir duquel le client peut accéder à la ressource désignée par la directive DocumentRoot.</p> <p><b>On doit configurer le service DNS</b> de façon à ce que celui-ci fasse la résolution des noms : nom_dns &lt;==&gt; @IP du serveur</p> <p><b>DocumentRoot</b> désigne donc le chemin absolu où se trouvent les documents du site.</p>
--	--

## 1. Analyse de la configuration du site web par défaut

- Le fichier représentant le site par défaut est **Index.html** et il se trouve dans **/var/www/html**
- Explication d contenu du fichier de configuration du site par défaut
  - **<VirtualHost \* : 80>** : Le serveur écoute sur le port 80 pour tous les adresses IP
  - **ServerName** : nom de domaine du site
  - **DocumentRoot** : le chemin absolue du site
  - **ErrorLog** : les infos des erreurs de log
  - le chemin du dossier qui contient les pages du site par défaut est :  
**/var/www/html**

```
root@debian:/etc/apache2/sites-enabled# cd /var/www
root@debian:/var/www# ls
html  sio1  sio2
root@debian:/var/www#
```

- La page web du site par défaut est:  
**index.html**  
**/var/www/html**

```
LAMP [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
25 janv. 08:14
mamadou@debian: ~
GNU nano 7.2 index.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">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;

        background-color: #D8DBE2;

        font-family: Verdana, sans-serif;
        font-size: 11pt;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div style="text-align: center;>
      <h1>It works!</h1>
    </div>
  </body>
</html>
[ Lecture de 368 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter
^X Quitter   ^R Lire fich. ^\ Remplacer ^U Coller    ^J Justifier
CTRL DROITE
```

- Le chemin du fichier qui stocke les logs liés aux erreurs. .  
</var/log/apache2/error.log>
- Le chemin du fichier qui stocke l'historique des accès au site :  
</var/log/apache2/access.log>

## 2. Création et configuration d'un hôte virtuel (vhost)

Il est très courant d'héberger et de gérer plusieurs sites, dits "**virtuels**" (vhost), par un seul et même serveur (**hébergement mutualisé**).

**Ceux-ci sont appelés par les clients sous différents noms DNS** ; le serveur écoute une seule adresse IP à laquelle sont associés plusieurs noms de sites qui seront utilisés dans les URL clientes.

Les sociétés se partageant un serveur Web peuvent ainsi avoir leurs propres domaines accessibles.

### a. Création d'hôtes virtuels

- En nous basant sur le site par défaut, nous allons ajouter 2 hôtes virtuels pour les sites **sio1.local** et **sio2.local**.

**Important** : Les pages Web de chaque site seront placées dans les dossiers suivants :

- **/var/www/sio1/** : pour le site **sio1.local**.
  - **/var/www/sio2/** : pour le site **sio2.local**.
- Pour activer ces 2 sites : **a2ensite « fichier de configuration du site »...**

- Pour les tests sur le serveur, on configure le fichier **/etc/hosts** en y ajoutant :

127.0.0.1	sio1.local	sio2.local	...
-----------	------------	------------	-----

- Pour les tests à partir d'un autre poste Linux, on configure le fichier **/etc/hosts** en y ajoutant :

Adresse-ip-serveur	sio1.local	sio2.local	...
--------------------	------------	------------	-----

**Remarque** : le fichier hosts est l'ancêtre du système DNS. Il permet la résolution de noms de domaines en adresses IP. Ce fichier est présent dans la plupart des systèmes. Sur le système Windows, l'emplacement est : **C:\Windows\System32\drivers\etc\hosts**.

## b. L'accès sécurisé au serveur Web (https)

- On vérifie si les paquetages « **openssl** » et « **libssl** » sont installés avec la commande :

```
dpkg -l | grep openssl
```

## Configuration d'apache2 avec le module ssl

- Le fichier représentant le site sécurisé par défaut est :  
**/etc/apache2/sites-available/default-ssl.conf**
- >Explication de son contenu :
  - <VirtualHost \* : **443**> : le site écoute sur le port 443 pour tous les IP
  - SSLEngine on : le ssl est activé
  - SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem : chemin absolue du certificat ssl
  - SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key : chemin absolue du clè
- Pour activer le module « ssl » on tape : `a2enmod ssl`
- Ajoute et configuration de 2 hôtes virtuels pour les 2 sites Web sécurisés **sio1.local** et **sio2.local** : `sio1-ssl.conf` et `sio2-ssl.conf`



- Pour renvoyer l'ensemble des requêtes HTTP vers le protocole HTTPS, on ajoute dans les fichiers de configuration des Vhosts la directive suivante :

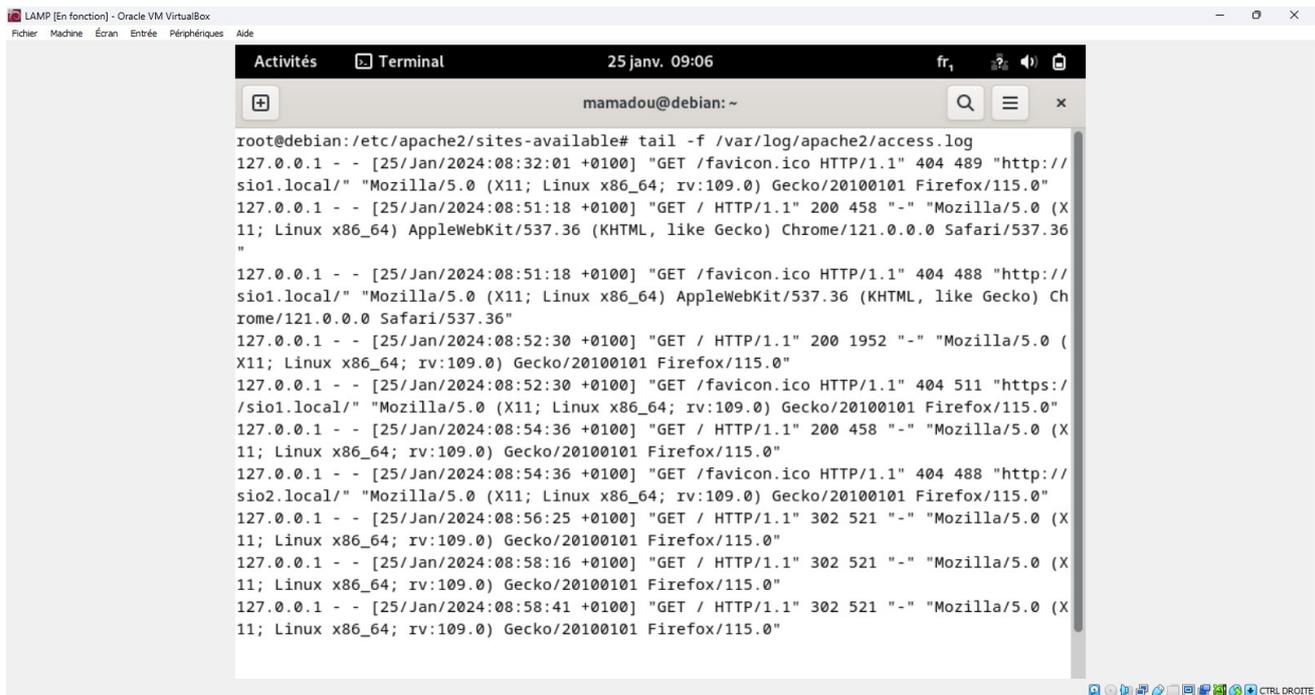
```
<VirtualHost * : 80>  
  
    ...  
  
    Redirect / https://sio1.local  
  
</VirtualHost>
```

- Vérifions en exécutant des requêtes <http://sio1.local> et <http://sio2.local>.



### 3. Analyse de l'activité du serveur Apache

- Tapons la commande : ***tail -f /var/log/apache2/access.log*** et lançons une requête <http://sio1.local>.



The screenshot shows a terminal window titled "Terminal" with the date "25 janv. 09:06". The user is logged in as "mamadou@debian: ~". The terminal output shows the command `tail -f /var/log/apache2/access.log` and its output, which consists of several lines of Apache access logs. Each line represents an HTTP request, including the IP address, timestamp, request method and path, status code, size, and user agent. The user agents include Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0 and AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36.

- Test en effectuant une requête <https://sio1.local> et <https://sio2.local>.

On remarque que le ping vers <https://sio1.local> fonctionne

- Le contenu du fichier `/var/log/apache2/error.log`.

#### 4. Analyse de trames

- Installons le capteur de trames Wireshark ;
- Capturons et analysons des trames à destination de notre serveur web Apache.

Requête du client

Debian clien [En fonction] - Oracle VM VirtualBox

Activités Wireshark 30 janv. 17:40 fr, [?], [Speaker], [Close]

\*eth0

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

http

No.	Time	Source	Destination	Protocol	Length	Info
1044	14.126144827	92.122.166.172	192.168.1.1	OCSP	943	Response
1424	15.074806899	192.168.1.1	92.122.166.180	OCSP	490	Request
1441	15.118323026	92.122.166.180	192.168.1.1	OCSP	943	Response
1834	28.733287417	192.168.1.1	142.250.201.35	OCSP	494	Request
1839	28.891436155	142.250.201.35	192.168.1.1	OCSP	756	Response
3065	32.203782085	192.168.1.1	192.168.1.2	HTTP	425	GET / HT
3067	32.205165292	192.168.1.2	192.168.1.1	HTTP	599	HTTP/1.1
3085	32.261676617	192.168.1.1	34.107.221.82	HTTP	368	GET /car
3095	32.282697133	34.107.221.82	192.168.1.1	HTTP	352	HTTP/1.1
3106	32.326337583	192.168.1.1	34.107.221.82	HTTP	370	GET /suc

Host: www.brocant.fr\r\n  
 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:31.0) Gecko/20100101 Firefox/31.0\r\n  
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8\r\n  
 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.7,en;q=0.3\r\n  
 Accept-Encoding: gzip, deflate\r\n  
 Connection: keep-alive\r\n  
 Upgrade-Insecure-Requests: 1\r\n  
 \r\n  
 [Full request URI: http://www.brocant.fr/]\r\n  
 [HTTP request 1/1]\r\n  
 [Response in frame: 3067]

Réponse du serveur

Debian clien [En fonction] - Oracle VM VirtualBox

Activités Wireshark 1 févr. 08:57 fr, [?], [Speaker], [Close]

Capture en cours de eth0

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

https

No.	Time	Source	Destination	Protocol	Length	Info
12	20.220403762	192.168.1.1	192.168.1.2	HTTP	421	GET / HTT
14	20.221430847	192.168.1.2	192.168.1.1	HTTP	587	HTTP/1.1
51	42.092779695	192.168.1.1	192.168.1.2	HTTP	421	GET / HTT
53	42.094004632	192.168.1.2	192.168.1.1	HTTP	587	HTTP/1.1

Frame 53: 587 bytes on wire (4696 bits), 587 captured (4696 bits) on interface eth0  
 Ethernet II, Src: PcsCompu\_7f:55:d5 (08:00:27:55:d5:7f), Dst: 192.168.1.1  
 Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.1  
 Transmission Control Protocol, Src Port: 80, Dst Port: 80  
 Hypertext Transfer Protocol  
 Line-based text data: text/html (9 lines)  
 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
 <html><head>\n  
 <title>302 Found</title>\n  
 </head><body>\n  
 <h1>Found</h1>\n  
 <p>The document has moved <a href="https://www.brocant.fr/">here</a>.\n  
 <hr>\n  
 <address>Apache/2.4.57 (Debian) Server at 192.168.1.1\n  
 </body></html>\n

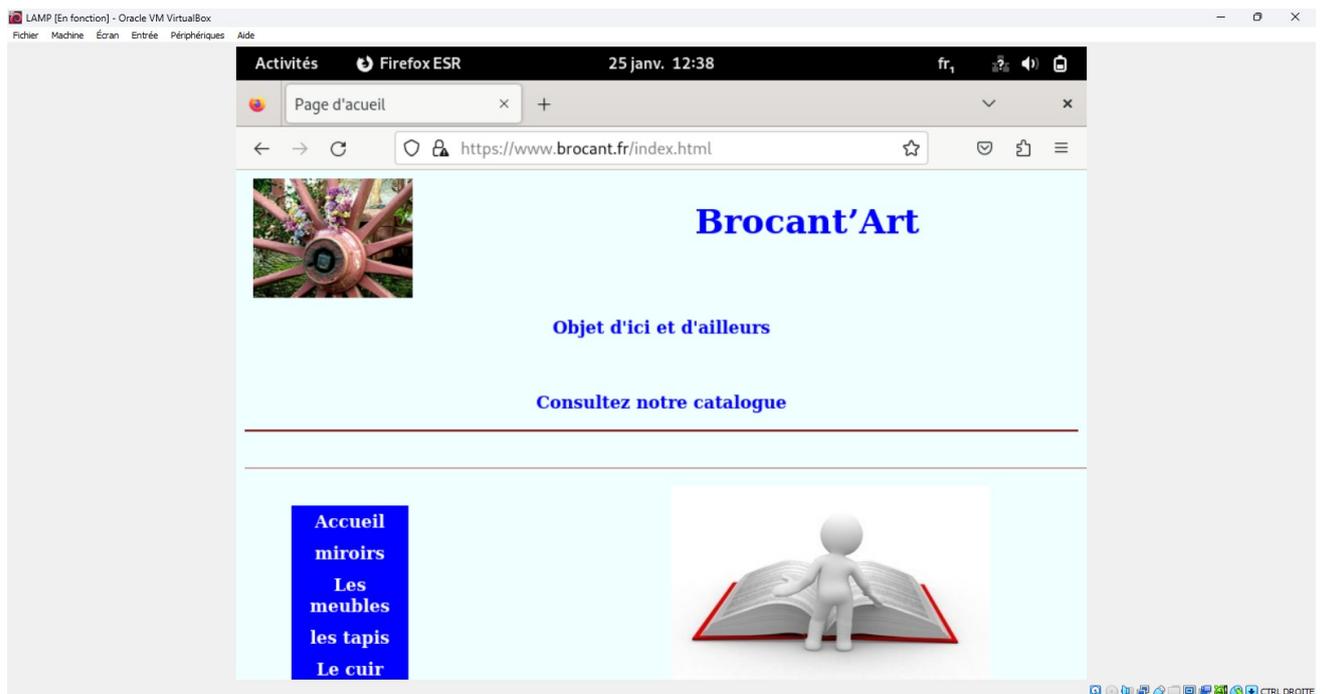
**Le premiers tests sur le serveur Web avec le site « sio1.local » que nous avons créé directement sur le serveur Web fonctionne.**

**Maintenant, nous allons héberger un site web « www.brocant.fr » sur le serveur Web.**

## **5. Hébergement du site Brocant'Art**

**Pour ce test :**

- Nous avons créé un hôte virtuel pour l'hébergement du site Brocant'Art réalisé en AP du 1er semestre. Ce site sera accessible via le nom de domaine *www.brocant.fr*;
- Ensuite, nous créé un dossier pour stocker les pages web du site ;
- Nous avons transféré les pages web vers notre serveur ;
- Enfin, on a crée un hôte virtuel pour l'accès https ;

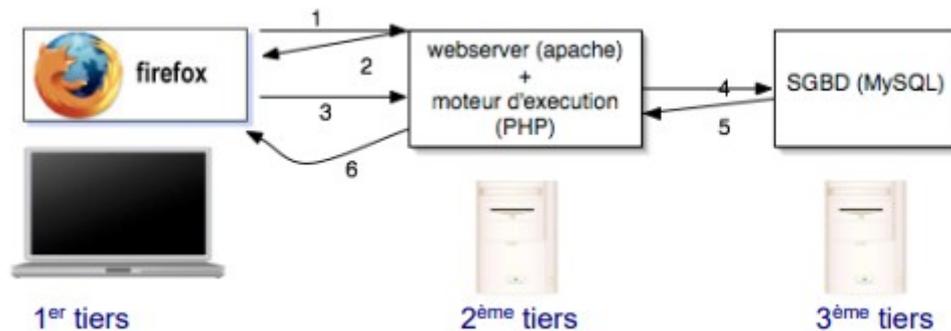


## Apache 2 : ajout de PHP, mariadb et PhpMyAdmin

- |                  |   |
|------------------|---|
| <b>Objectifs</b> | <ul style="list-style-type: none"><li>• Installation du système de gestion de bases de données « mariadb » ;</li><li>• Installation et configuration de phpMyAdmin.</li></ul> |
|------------------|---|

### Etape 2 : Installation de PHP :

#### 2. Architecture web 3 tiers :



### 3. Installation des paquets pour PHP :

```
apt install libapache2-mod-php php-cli php-mysql php-common php-zip  
php-curl php-xml wget -y
```

### Etape 3 : Installation de mariadb :

#### 1. Installation du paquet « mariadb-server » avec la commande :

```
apt install mariadb-server
```

#### 2. Démarrage et activation du service mariadb :

```
systemctl enable --now mariadb
```

#### 3. Sécurisation de mariadb :

## 4. Création d'un utilisateur sur le serveur de bases de données mariadb :

```
Sudo mysql  
CREATE USER 'admin'@localhost IDENTIFIED BY 'password';
```

- **admin** : est le nom de l'utilisateur
- **localhost** : est le nom du serveur (local)

## 5.. Attribution des droits à l'utilisateur admin :

```
GRANT ALL PRIVILEGES ON *.* TO 'admin'@localhost IDENTIFIED BY  
'password';  
FLUSH PRIVILEGES;
```

Pour se connecter à la base de donnée avec l'utilisateur **admin**, on utilise la commande : **mysql -u admin -p**

**Pour** afficher la liste des bases de données présentes sur le serveur, on utilise la commande : **show databases**

## Création d'une nouvelle base de données nommée « db-ventes »

```
create databe db_ventes ;
```

## Utilisation de la base de donnée db\_vente :

```
use db_vente ;
```

## Pour afficher les tables de la base de données « db-ventes » :

```
show tables ;
```

### - Création d'une table nommée produit :

```
create table produits (id int(4), nom varchar(20), prix float(7,2)) ;
```

- Ajouter des produits dans la table « produits »

```
> insert into produits values (1, " Ecran PC ", 100.20),  
                             (2, " Souris ", 30.55),  
                             (3, " Clavier ", 35.99) ;
```

Pour afficher le contenu de la table produit, on utilise la commande :

```
select * from produits ;
```

## Etape 4 : Installation et configuration de « phpMyAdmin »

### Téléchargement du package phpMyAdmin :

```
wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-  
languages.tar.gz
```

wget est un programme en ligne de commande de téléchargement de fichiers depuis le Web. Il supporte les protocoles HTTP, HTTPS et FTP.

Création du dossier suivant sous la racine du serveur web apache 2 :

```
/var/www/html/phpmyadmin
```

### Création d'un fichier de configuration en copiant celui déjà présent dans le dossier phpMyAdmin :

```
cd /var/www/html  
sudo cp phpmyadmin/config.sample.inc.php phpmyadmin/config.inc.php
```

**Créer un dossier temporaire nommé « tmp » pour le stockage, s'il n'existe pas**

```
sudo mkdir /var/www/html/phpmyadmin/tmp
```

### Modification de la configuration de phpMyAdmin :

```
sudo nano /var/www/html/phpmyadmin/config.inc.php
```

**On modifie** le dossier temporaire à la ligne suivante :

```
$cfg['TempDir'] = '/var/www/html/phpmyadmin/tmp';
```

**On donne accès à l'utilisateur Apache pour lire les fichiers :**

```
sudo chown -R www-data:www-data /var/www/html/phpmyadmin
```

## Création d'un fichier de configuration Apache pour phpMyAdmin

```
Alias /phpmyadmin /var/www/html/phpmyadmin

<Directory /var/www/html/phpmyadmin/>
  AddDefaultCharset UTF-8
  <IfModule mod_authz_core.c>
    <RequireAny>
      Require all granted
    </RequireAny>
  </IfModule>
</Directory>

<Directory /var/www/html/phpmyadmin/setup/>
  <IfModule mod_authz_core.c>
    <RequireAny>
      Require all granted
    </RequireAny>
  </IfModule>
</Directory>
```

**On active la configuration :**

```
sudo a2enconf phpmyadmin.conf
```

# Apache 2 : Installation de la dernière version de PHP

Les packages PHP version 8.3 ne sont pas disponibles par défaut sur les dépôts officiels Debian, mais vous pouvez les récupérer à partir du dépôt [sury.org](https://packages.sury.org). Pour commencer, il faut installer quelques dépendances requises :

```
apt install -y lsb-release ca-certificates apt-transport-https  
software-properties-common gnupg2
```

## 1. Téléchargement des clés gpg :

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

## Installation des paquets suivants :

```
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" >  
/etc/apt/sources.list.d/php.list
```

OU

```
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" |  
tee /etc/apt/sources.list.d/sury-php.list
```

## 3. Installation des paquets PHP 8.3 :

```
apt install php8.3
```

installation de la version de PHP 8 fpm et curl :

```
apt install php8.3-fpm
```

```
apt install php8.3-curl
```

# Synthèse :

## 1) Protocole HTTP :

### Fonctionnement du protocole HTTP :

- Le client demande une page web via le navigateur
- Le serveur envoie la réponse (HTML, CSS, javascript, image)
- Le serveur web envoie les requête SQL au SGBD
- Réponse du serveur SGBD
- Page envoyée au client

Remarque : L'interpréteur PHP traduit le code écrit en PHP

## 2) Apache2: Les fichiers de configuration se trouve dans `/etc/apache2`

- Fichier de conf principale : `/etc/apache2.conf`
- Sites disponibles : `/etc/apache2/sites-available`

Activation du Vhost : a2ensite `sio1.conf`

## 3) HTTPS : port 443

HTTPS = HTTP +SSL (TLS)

→ Activation du Vhost : a2ensite `sio1-ssl.conf`

```
apache2ctl configtest ou apache2ctl -t
```

- On affiche l'état de statue du service apache avec la commande suivante :

```
apache2ctl status  
OU  
systemctl status apache2
```

- La directive `ServerName` est le nom du serveur tel qu'il devra être tapé dans la barre d'adresse du navigateur