

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

SERVEUR LAMP DEBIAN

SOMMAIRE :

I)	Objectif.....	2
II)	Prérequis.....	2
III)	Définitions.....	2
IV)	Installation et configuration du service Apache.....	2-3
V)	Création du répertoire personnel des utilisateurs.....	3-4
VI)	Accès privé au répertoire.....	4-5
VII)	Installation et configuration du service « php5 ».....	6-7
VIII)	Hébergement virtuel.....	7-13
	a) Hébergement virtuel par l'adresse IP.....	7-9
	b) Hébergement virtuel par le nom.....	10-13
IX)	Conclusion.....	13

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

I) Objectif

Dans cette procédure, nous allons montrer comment installer et configurer un serveur Web en utilisant les paquets **Apache2**, **MySQL** et **PHP5** sous Debian.

II) Prérequis

Pour réaliser cette procédure, nous avons besoin des éléments suivants :

OS	Distribution	Version	C/S
Debian Jessie	Linux	8.5	S : Serveur Web Apache C : Client Windows 7

Nom du serveur Web	Adresse IP du serveur Web	Adresse IP virtuelle
serveurweb	192.168.1.132	192.168.1.133

III) Définitions

- Le service **Apache** est un serveur **HTTP (HyperText Transfer Protocol)** qui permet de gérer la communication avec le client. Ce serveur Web est le plus populaire de **WWW (World Wide Web)**.
- Une **adresse IP virtuelle** est une adresse IP non connectée à un poste ou une carte réseau. Celle-ci permet de faire de la redondance de connexion entre les serveurs en cas de défaillances sur le réseau. C'est-à-dire que si le site Web en question n'est plus accessible, nous pouvons y accéder par l'intermédiaire de cette **adresse IP**.
- Le service **PHP** est un langage de programmation de pages web dynamiques pouvant être utilisé dans du code **HTML**.

IV) Installation et configuration du service Apache

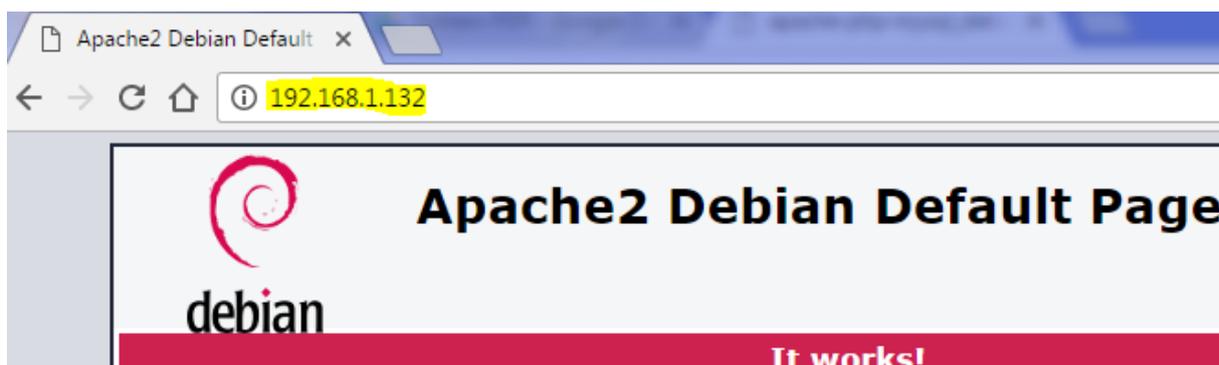
- Tout d'abord, nous mettons à jour les paquets :

```
root@serveurweb:~# apt-get update
```

- Nous installons le service « **apache2** » :

```
root@serveurweb:~# apt-get install apache2_
```

- Nous vérifions son bon fonctionnement via un navigateur :



ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Ensuite, dans le fichier « **/etc/apache2/apache2.conf** », nous ajoutons la ligne (en jaune) pour nommer le serveur Web :

```
#ServerRoot "/etc/apache2"
ServerName serveurweb
```

- Dans le fichier « **/etc/hosts** », nous ajoutons la ligne en jaune pour préciser l'adresse IP et le nom du serveur :

```
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
127.0.1.1 serveurweb
192.168.1.132 serveurweb
```

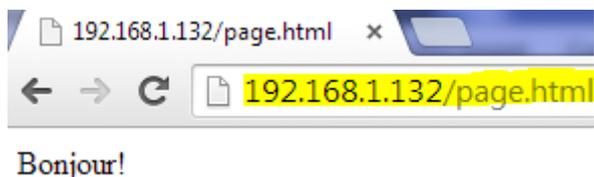
- Nous tapons cette commande afin de vérifier la configuration d'**Apache** et constatons que celle-ci est correcte :

```
root@serveurweb:~# apache2ctl -t
Syntax OK
root@serveurweb:~# _
```

- Nous créons et éditons un nouveau fichier **HTML** nommé « **page.html** » dans le répertoire « **/var/www/html** » et saisissons du code **HTML** :

```
GNU nano 2.2.6 Fichier : /var/www/html/page.html
Bonjour!_
```

- Nous vérifions via un navigateur et constatons que nous accédons à la page :



V) Création du répertoire personnel des utilisateurs

- Maintenant, nous créons un nouvel utilisateur nommé « **toto** » :

```
root@serveurweb:~# adduser toto_
```

- Nous nous connectons en tant qu'utilisateur « **toto** » et créons le répertoire personnel « **public_html** » dans le dossier de cet utilisateur :

```
toto@serveurweb:~$ mkdir /home/toto/public_html
toto@serveurweb:~$ _
```

- Nous nous reconnectons en tant que « **root** », créons dans le dossier de l'utilisateur un fichier **HTML** nommé « **cv.html** », par exemple et vérifions sa présence :

```
root@serveurweb:~# ls /home/toto/public_html/
cv.html
root@serveurweb:~# _
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Pour pouvoir visualiser le contenu de ce dossier, nous devons activer le module permettant de publier les fichiers de l'utilisateur créé :

```
root@serveurweb:/home/toto/public_html# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  service apache2 restart
root@serveurweb:/home/toto/public_html# _
```

- Nous redémarrons le service « **apache2** » pour prendre en compte les modifications :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

- Nous vérifions via un navigateur en tapant l'URL « http://@IP_serveurweb/~toto/ » et constatons que le contenu du dossier personnel de l'utilisateur « **toto** » s'affiche :



VI) Accès privé au répertoire

- Nous nous connectons en tant qu'utilisateur et lui attribuons un mot de passe :

```
toto@serveurweb:~/public_html$ htpasswd -c .privpasswd toto
New password:
Re-type new password:
Adding password for user toto
toto@serveurweb:~/public_html$ _
```

- Nous sécurisons son dossier personnel en créant le fichier « **.htaccess** » qui permet d'afficher l'interface de connexion dans le dossier « **/home/toto/public_html** » :

```
toto@serveurweb:~/public_html$ nano .htaccess.
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

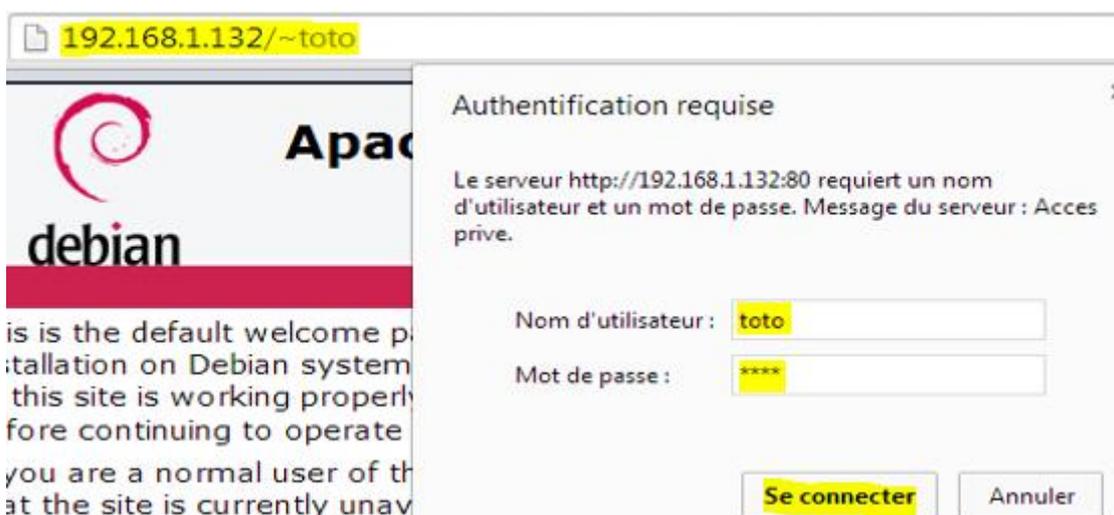
- Nous saisissons le contenu suivant permettant l'authentification de l'utilisateur « **toto** » sur son dossier personnel :

```
GNU nano 2.2.6  Fichier : /home/toto/public_html/.htaccess
AuthType Basic
AuthName "Bienvenue, veuillez vous connecter"
AuthUserFile /home/toto/public_html/.privpasswd
AuthName "Acces prive"
#<limit GET>
require valid-user
#</limit>
```

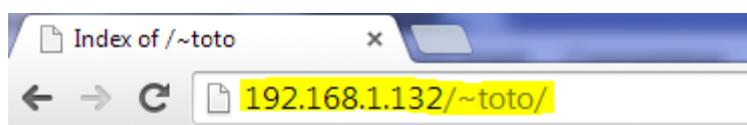
- Nous redémarrons le service « **apache2** » :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

- Nous testons la connexion au dossier personnel de l'utilisateur :



- Et, nous accédons bien à son espace personnel avec son contenu :



Index of /~toto

Name	Last modified	Size	Description
 Parent Directory		-	
 cv.html	2016-09-06 09:25	7	

Apache/2.4.10 (Debian) Server at 192.168.1.132 Port 80

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

VII) Installation et configuration du service « php5 »

- Nous installons les services suivants pour **PHP** :

```
root@serveurweb:~# apt-get install libapache2-mod-fcgid php5-cgi php5-cli
```

- Nous activons le module « **fcgid** » :

```
root@serveurweb:~# a2enmod fcgid
```

- Maintenant, dans le fichier « **/etc/apache2/sites-available/000-default.conf** », nous ajoutons le contenu suivant (en jaune) :

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme
    # the server uses to identify itself. This is use
    # redirection URLs. In the context of virtual hos
    # specifies what hostname must appear in the requ
    # match this virtual host. For the default virtua
    # value is not decisive as it is used as a last r
    # However, you must set it for any further virtua
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/html/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        AddHandler fcgid-script .php
```

```
    <Directory /usr/lib/cgi-bin/>
        Options +ExecCGI
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
```

```
        Allow from all
    </Directory>
```

- ⇒ « **<Directory/** » définit les options par défaut du répertoire.
- ⇒ « **FollowSymLinks** » permet de suivre les liens symboliques.
- ⇒ « **Indexes** » permet de lister les fichiers d'un dossier n'ayant pas d'index.
- ⇒ « **AllowOverride** » permet de surcharger quelques options en utilisant des fichiers « **.htaccess** » dans le dossier du site.

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

⇒ La ligne « **AddHandler** » permet de spécifier l'extension de fichier à exécuter via **FCGI**. Il suffit d'indiquer où se situe l'application concernant cette extension. Cela se fait avec « **FCGIWrapper** ». La ligne « **options +ExecCGI** » permet d'exécuter les **CGI**.

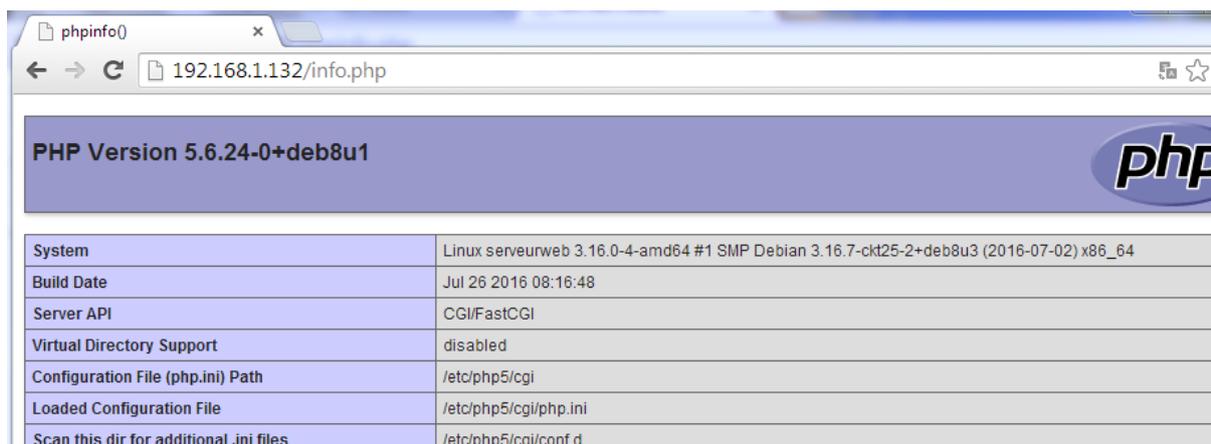
- Maintenant, nous créons un fichier nommé « **/var/www/html/info.php** » et saisissons le contenu suivant :

```
GNU nano 2.2.6 Fichier : /var/www/html/info.php
<?php
phpinfo()
?>
```

- Nous redémarrons le service « **apache2** » pour prendre en compte les modifications :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

- Nous vérifions le fonctionnement de **PHP** via un navigateur en tapant l'adresse IP du serveur avec le nom du fichier **PHP** et constatons que nous accédons bien à la page du service **PHP** :



VIII) Hébergement virtuel

a) Hébergement virtuel par l'adresse IP

- Nous ajoutons un alias IP dans le fichier « **/etc/network/interfaces** » :

```
auto 0:0
iface eth0:0 inet static
address 192.168.1.133
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous désactivons et activons la nouvelle interface et constatons que cela fonctionne :

```
root@serveurweb:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:98:64:74
          inet addr:192.168.1.132  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe98:6474/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20009 errors:0 dropped:344 overruns:0 frame:0
          TX packets:8727 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:21965345 (20.9 MiB)  TX bytes:608178 (593.9 KiB)

eth0:0    Link encap:Ethernet  HWaddr 08:00:27:98:64:74
          inet addr:192.168.1.133  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

- Nous redémarrons le service réseau « **networking** » :

```
root@serveurweb:~# systemctl restart networking.service
root@serveurweb:~# _
```

- Maintenant, nous pouvons faire un **ping** de la nouvelle adresse et la communication fonctionne :

```
root@serveurweb:~# ping 192.168.1.133
PING 192.168.1.133 (192.168.1.133) 56(84) bytes of data.
64 bytes from 192.168.1.133: icmp_seq=1 ttl=64 time=0.038 ms
64 bytes from 192.168.1.133: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 192.168.1.133: icmp_seq=3 ttl=64 time=0.043 ms
```

- Nous créons 2 répertoires pour les 2 hébergements virtuels :

```
root@serveurweb:~# mkdir /var/www/html/sio2A /var/www/html/web
root@serveurweb:~# _
```

- Nous créons dans ces 2 répertoires un fichier nommé « **logs** » :

```
root@serveurweb:~# mkdir /var/www/html/sio2A/logs /var/www/html/web/logs
root@serveurweb:~# _
```

- Maintenant, nous créons un fichier nommé « **sites-web.conf** » dans le répertoire « **/etc/apaches2/sites-available** » et saisissons tout le contenu suivant permettant l'accès aux 2 serveurs Web :

```
<VirtualHost 192.168.1.132>
  ServerName sio2A
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html/sio2A
  ErrorLog /var/www/html/sio2A/logs/error_log
  TransferLog /var/www/html/sio2A/logs/access_log
</VirtualHost>

<VirtualHost 192.168.1.133>
  ServerName web
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html/web
  ErrorLog /var/www/html/web/logs/error_log
  TransferLog /var/www/html/web/logs/access_log
</VirtualHost>
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous désactivons le fichier « **000-default.conf** » :

```
root@serveurweb:~# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  service apache2 reload
root@serveurweb:~# _
```

- Nous redémarrons le service « **apache2** » pour prendre en compte les modifications :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

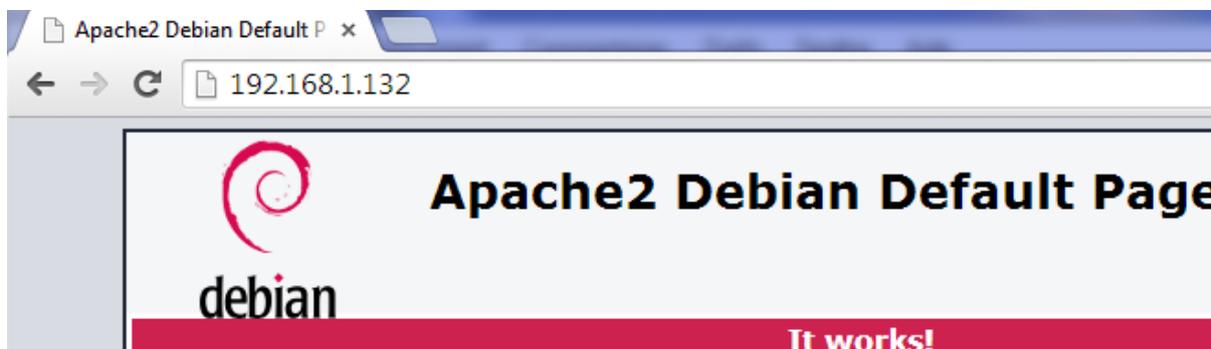
- Nous activons le répertoire virtuel :

```
root@serveurweb:~# a2ensite sites-web.conf
Enabling site sites-web.
To activate the new configuration, you need to run:
  service apache2 reload
root@serveurweb:~# _
```

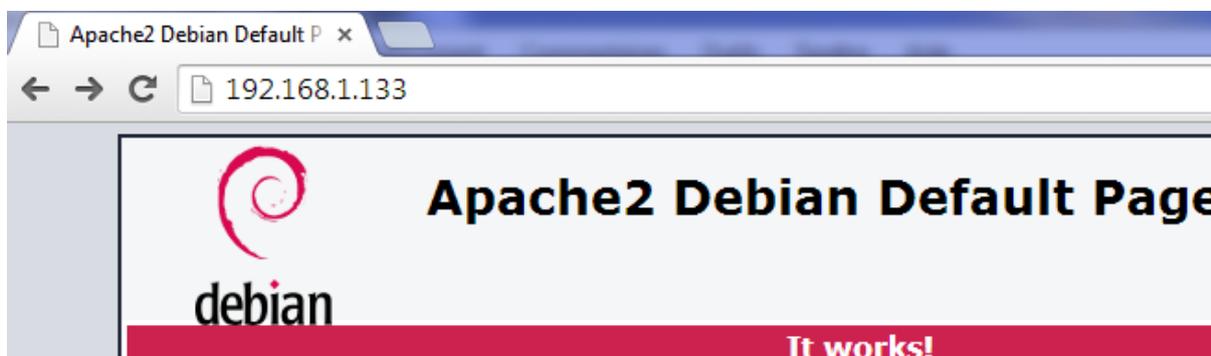
- Nous redémarrons de nouveau le service « **apache2** » pour prendre en compte les modifications :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

- Nous testons l'adresse IP du serveur et constatons que cela fonctionne :



- Nous testons l'adresse IP virtuelle et constatons que cela fonctionne également :



ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

b) Hébergement virtuel par le nom

- Nous créons les 2 répertoires respectifs pour l'hébergement des noms des sites « **sisr.local** » et « **slam.local** » :

```
root@serveurweb:~# mkdir /var/www/html/sisr /var/www/html/slam
root@serveurweb:~# _
```

- Nous créons dans ces 2 répertoires un fichier nommé « **logs** » :

```
root@serveurweb:~# mkdir /var/www/html/sisr/logs
root@serveurweb:~# mkdir /var/www/html/slam/logs
root@serveurweb:~# _
```

- Nous créons et configurons un fichier nommé « **sites-sio.conf** » dans le répertoire « **/etc/apache2/sites-available** » et saisissons tout le contenu suivant permettant l'accès à ces 2 sites :

```
#NameVirtualHost *:80
<VirtualHost *:80>
DocumentRoot /var/www/html/sisr
ServerName sisr.local
ServerAlias www.sisr.local
ErrorLog /var/www/html/sisr/logs/error_log
TransferLog /var/www/html/sisr/logs/access_log
</VirtualHost>

<VirtualHost *:80>
DocumentRoot /var/www/html/slam
ServerName slam.local
ServerAlias www.slam.local
#Error_log ${APACHE_LOG_DIR}/logs/error_log
ErrorLog /var/www/html/slam/logs/error_log
TransferLog /var/www/html/slam/logs/access_log
</VirtualHost>
```

- Nous désactivons le fichier « **sites-web.conf** » pour pouvoir ensuite activer le fichier « **sites-sio.conf** » :

```
root@serveurweb:/etc/apache2/sites-available# a2dissite sites-web.conf
Site sites-web disabled.
To activate the new configuration, you need to run:
  service apache2 reload
root@serveurweb:/etc/apache2/sites-available# _
```

- Nous redémarrons le service « **apache2** » pour prendre en compte les modifications :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

- Donc, nous activons le fichier « **sites-sio.conf** » :

```
root@serveurweb:/etc/apache2/sites-available# a2ensite sites-sio.conf
Enabling site sites-sio.
To activate the new configuration, you need to run:
  service apache2 reload
root@serveurweb:/etc/apache2/sites-available# _
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous redémarrons de nouveau le service « **apache2** » pour prendre en compte les modifications :

```
root@serveurweb:~# systemctl restart apache2.service
root@serveurweb:~# _
```

- Ensuite, nous éditons le fichier « **/etc/hosts** » et ajoutons le nom des 2 sites avec l'adresse IP du serveur Web :

```
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
127.0.1.1 serveurweb
192.168.1.132 sivr.local
192.168.1.132 slam.local
```

- Nous faisons de même sur la machine cliente. Pour ce faire, nous nous rendons dans le dossier « **C:\Windows\System32\drivers\etc** », éditons le fichier « **hosts** » et ajoutons le même contenu (en jaune) que le fichier « **/etc/hosts** » sous Debian :

```
hosts - Bloc-notes
Fichier Edition Format Affichage
# Copyright (c) 1993-2009 M
#
# This is a sample HOSTS fi
#
# This file contains the ma
# entry should be kept on a
# be placed in the first co
# The IP address and the hc
# space.
#
# Additionally, comments (s
# lines or following the ma
#
# For example:
#
# 102.54.94.97 rhi
# 38.25.63.10 x.a
#
# localhost name resolutior
# 127.0.0.1 loc
# ::1 loc
127.0.0.1 localhost
192.168.1.132 sivr.local
192.168.1.132 slam.local
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Enfin, nous créons un fichier **HTML** nommé « **index.html** » dans les répertoires respectifs « **/var/www/html/sisr** » et « **/var/www/html/slam** » et saisissons du code **HTML** pour vérifier le fonctionnement :

- o Voici le contenu du fichier **HTML** du site **SISR** :

```
GNU nano 2.2.6      Fichier : /var/www/html/sisr/index.html
<!DOCTYPE html>
<html>
<head>
  <title>Site SISR</title>
</head>
<body>
<h1>Site SISR</h1>
</body>
</html>
```

- o Voici le contenu du fichier **HTML** du site **SLAM** :

```
GNU nano 2.2.6      Fichier : /var/www/html/slam/index.html
<!DOCTYPE html>
<html>
<head>
  <title>Site SLAM</title>
</head>
<body>
<h1>Site SLAM</h1>
</body>
</html>
```

- Nous vérifions via un navigateur, l'accès aux 2 sites en tapant leur nom qui sont « **sisr.local** » et « **slam.local** » :



Donc, nous pouvons constater que les 2 sites sont bien accessibles.

ETTORI Bastien	BTS SIO 2^{ème} année
22 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

NB : Pour vérifier leur bon fonctionnement, nous devons désactiver l'interface virtuelle.

IX) Conclusion

En conclusion, nous pouvons dire que le serveur Web est fonctionnel et que celui-ci permet bien de gérer la communication entre le client et le serveur et permet l'accès aux différents sites Web mis en place.