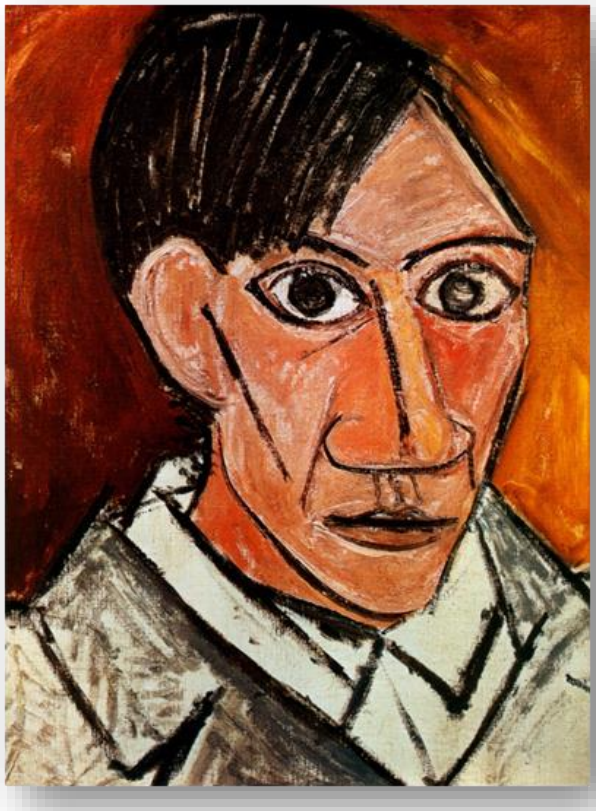
Rapport de mise en œuvre de l'infrastructure



17/03/2023

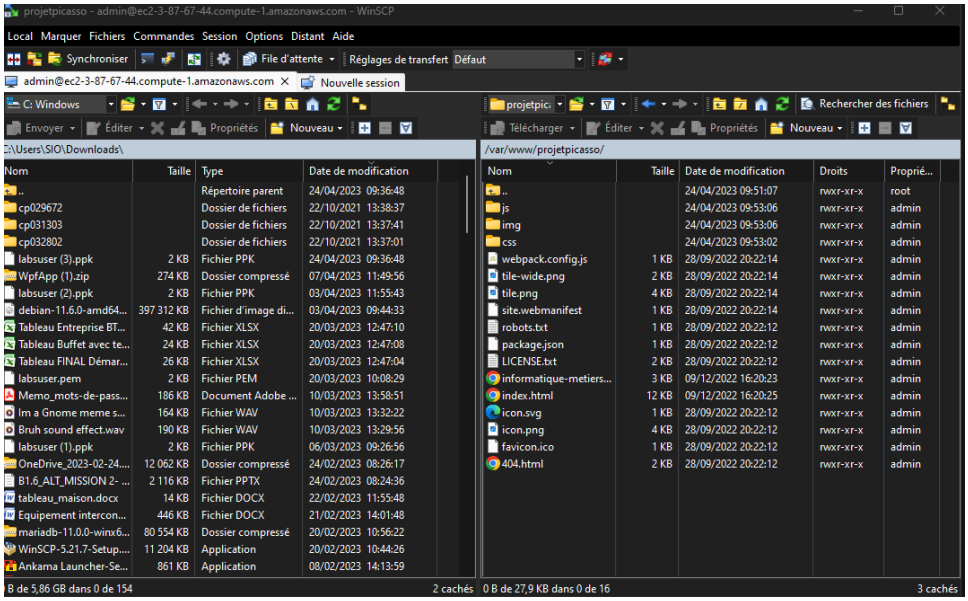
Lycée Saint Michel
Auteur : ZIANI Théo

SAINT MICHEL
Annecy Haute-Savoie

Infrastructure réseau et systèmes

Sommaire :

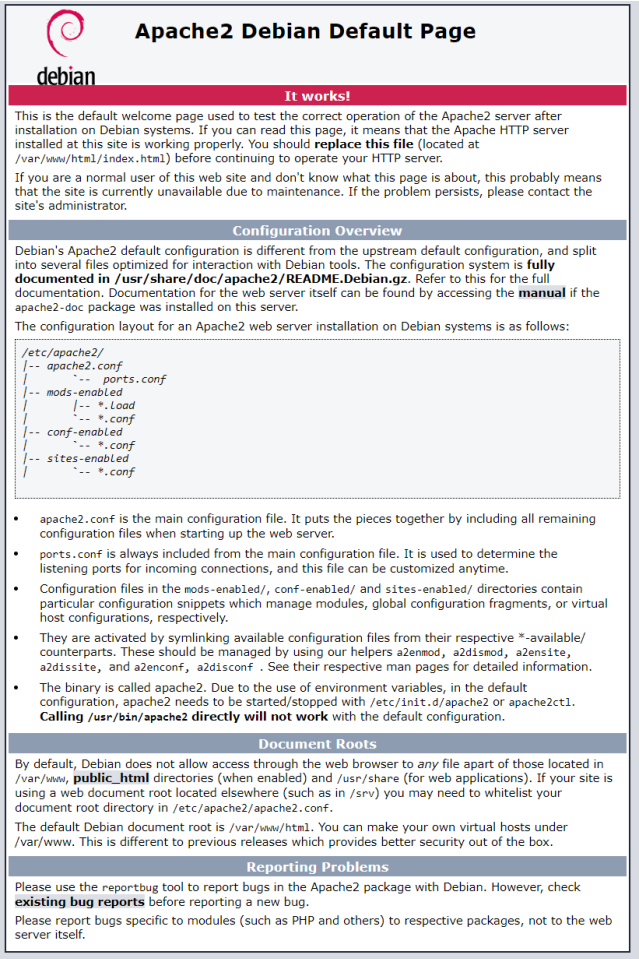
Il reste à aller sur le logiciel WinSCP pour transférer les fichiers du html du site Bootstrap dans le dossier "***var/www/projetpicasso***"



Ensuite on utilise la commande "***sudo mkdir /var/www/html/*** [***www.projetpicasso.com***](http://www.projetpicasso.com)***"***
"sudo cp /var/www/html/index.html /var/www/html/www.projetpicasso.com"

Puis recharger la configuration Apache
"sudo systemctl reload apache2"

Aller sur navigateur et copier l'adresse du site



Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Debian does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Debian document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `reportbug` tool to report bugs in the Apache2 package with Debian. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

5. Installation Firewall

Pour gérer les droits administrateurs et pour la sécurité du site il faut installer un firewall. Cela se fait aussi sur Putty avec des lignes de commandes

En premier il faut installer le Firewall : "***sudo apt install ufw***"

Grace au firewall nous pouvons autoriser différents moyens de connexion comme avec le port ***80***, on peut créer des règles, des statues

6. Installation PHP/MySQL

Cela va permettre à nos collègues en SLAM de pouvoir codé sur notre site en intégrant du PHP donc du code :
"sudo apt install mariadb-server" puis faire la commande "***sudo mysql_secure_installation"***

Pour définir un mot de passe administrateur

"mysql -h localhost -u root –p"

- ***mysql***, c'est le nom du client texte qui permet de se connecter à MySQL

- ***h localhost***, précise le nom de l'hôte auquel vous voulez vous connecter. Localhost est l'hôte par défaut, donc vous pouvez vous passer de cette option pour les connexions locales

- ***u root***, indique le nom de l'utilisateur avec lequel vous voulez vous connecter

- ***p***, demande d'afficher un prompt pour entrer le mot de passe de manière interactive

7. Sécurité mise en œuvre

a. Certificat auto-signé

Cela permet d'avoir un protocole un peu mieux qu'un site normal et donc un peu plus référencé

b. Les mots de passes administrateurs

Pour avoir accès au droit administrateur en-haut nous avons mis en place un mot de passe "***root***"

c. Firewall

Autorise l'accès à certains ports et la connexion avec une ip précise.

d. Let's Encrypt

Nous avons téléchargé avec la ligne de commande
sudo apt-get install letsencrypt

En utilisant la commande ***certonly*** cela demande de générer un certificat mais sans l'installer sur le serveur web
Pour pouvoir utiliser les certificats dans Apache, il faut modifier notre VirtualHost

Slide présentation :