

SAMONTE Jean-Vincent
SESSOU A.S Alfred
BTS SIO 1 RD

Atelier Pro - Apache

On souhaite héberger notre site dans le serveur HTTP Apache pour que ce dernier soit accessible via un navigateur internet. Nous allons donc installer un serveur web Apache sur notre serveur Ubuntu sur VirtualBox et pratiquer dans le terminal.

1 / Installation d'Apache

Premièrement, nous allons utiliser la commande suivante :

```
sudo apt update
```

Nous devons mettre à jour la liste des packages locaux. Il est important de mettre à jour cette liste afin de garantir qu'on installe la dernière version d'Apache.

Après avoir mis à jour les packages, on peut désormais installer **Apache2** via cette commande :

```
sudo apt install apache2
```

On confirme l'installation ce qui va installer Apache ainsi que toutes ses dépendances.

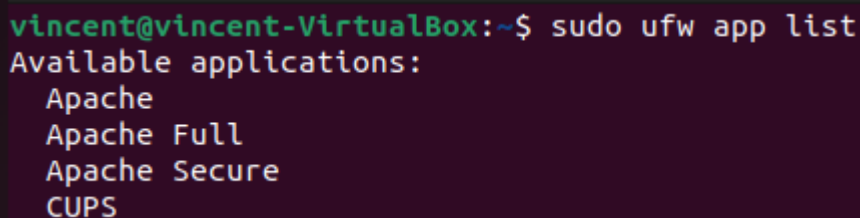
2 / Réglage du pare-feu

Il est nécessaire de régler le pare-feu pour autoriser l'accès extérieur aux ports Web par défaut. On doit disposer d'un pare-feu UFW configuré pour restreindre l'accès à notre serveur.

Apache s'enregistre auprès de UFW pour fournir quelques profils d'application pour activer / désactiver l'accès à Apache via le pare-feu.

```
sudo ufw app list
```

On répertorie la liste des profils d'application :



```
vincent@vincent-VirtualBox:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

Le résultat indique :

- Apache : Le profil n'ouvre que le port 80 (Trafic Web Normal non chifré)
- Apache Full : Ce profil ouvre à la fois le port 80 et le port 443 (Trafic chiffré TLS/SSL)

- Apache Secure : Ce profil ouvre uniquement le port 443.
- CUPS (Common Unix Printing System) : Système d'impression open source pour fournir un ensemble de pilotes d'imprimante et de services d'impression.

Comme nous n'avons pas encore configuré SSL (Secure Sockets Layer), nous allons autoriser le trafic sur le port 80 via cette commande ci-dessous :

```
sudo ufw allow 'Apache'
```

Le profil a donc été activé pour permettre l'accès au serveur Web Apache.

3 / Vérification de notre serveur Web

Pour vérifier auprès du système d'initialisation que le service est en cours d'exécution, nous allons taper la commande suivante dans le terminal :

```
sudo systemctl status apache2
```

On a pour résultat :

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pre
   Active: active (running) since Tue 2023-04-04 10:22:21 CEST; 44min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2752 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/
  Main PID: 2756 (apache2)
    Tasks: 55 (limit: 2288)
   Memory: 7.0M
      CPU: 278ms
   CGroup: /system.slice/apache2.service
           └─2756 /usr/sbin/apache2 -k start
             └─2757 /usr/sbin/apache2 -k start
               └─2758 /usr/sbin/apache2 -k start
```

Cette commande permet de vérifier si le service Apache2 est en marche. Ce qui nous permettra de continuer la configuration de notre serveur.

Nous avons eu la page par défaut en mettant dans la barre de recherche d'un navigateur l'adresse IP de notre machine.

4 / Configuration des hôtes virtuels

Dans cette partie, nous allons créer un domaine appelé **site_web**.
D'abord, il faudra créer le répertoire pour notre site web.

```
sudo mkdir /var/www/site_web
```

On crée le répertoire dans l'emplacement commun "/var/www" sur les serveurs Web Linux où sont stockés les fichiers du site Web. Ce dossier est généralement utilisé par le serveur Web pour stocker les fichiers de contenu Web.

On attribue la propriété de l'annuaire avec la variable d'environnement : \$USER

```
sudo chown -R $USER:$USER /var/www/site_web
```

La commande modifie le propriétaire et le groupe du répertoire /var/www/site_web et de tous ses fichiers de façon à ce qu'ils appartiennent à l'utilisateur actuel.

Le "chown" est important : Abréviation de "change owner" pour modifier le propriétaire d'un fichier ou répertoire.

Ensuite, on affecte les permissions du répertoire /var/www/site_web et de tous ses fichiers pour qu'ils soient accessibles en lecture et en écriture par le propriétaire du fichier, et en lecture et en exécution par tous les autres utilisateurs avec cette commande :

```
sudo chmod -R 755 /var/www/site_web
```

On crée un exemple de page en utilisant notre éditeur index.html.

```
sudo nano /var/www/site_web/index.html
```

```
GNU nano 6.2 /var/www/site_web/index.html
<html>
  <head>
    <title>Bienvenue à Paris !</title>
  </head>
  <body>
    <h1>Le site web est en marche.</h1>
  </body>
</html>
```

On enregistre le fichier avec **Contrôle + X**, on appuie sur **Y(Yes)** pour confirmer et on appuie sur **Entrée** pour sortir du fichier.

Il est nécessaire de créer un fichier hôte virtuel avec les directives correctes.
On utilisera donc cette commande :

```
sudo nano /etc/apache2/sites-available/site_web.conf
```

```
GNU nano 6.2 /etc/apache2/sites-available/site_web.conf
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName site_web
    ServerAlias www.site_web
    DocumentRoot /var/www/site_web
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Ici nous avons mis à jour notre nouveau répertoire et l'e-mail auquel l'administrateur du site "site_web" peut accéder.

On active le fichier avec l'outil a2ensite :

```
sudo a2ensite site_web.conf
```

On désactive le site par défaut défini dans : 000-default.conf

```
sudo a2dissite 000-default.conf
```

Après avoir rentré ces commandes, on redémarre Apache.

```
sudo systemctl restart apache2
```

On constate que www.site_web ne fonctionne pas, mais uniquement en localhost. Pour régler ce problème, nous allons rentrer cette commande dans le terminal :

```
sudo nano /etc/hosts
```

Avec cette commande on va faire en sorte que l'adresse IP de notre machine puisse accéder au nom du domaine www.site_web.

On remplace le champ localhost par www.site_web.

Ensuite, on redémarre à nouveau Apache2 :

```
sudo systemctl restart apache2
```



On en conclut que le site fonctionne, il est accessible via le nom du domaine plutôt que l'adresse IP.

