

TP Configuration serveur Apache sous Ubuntu

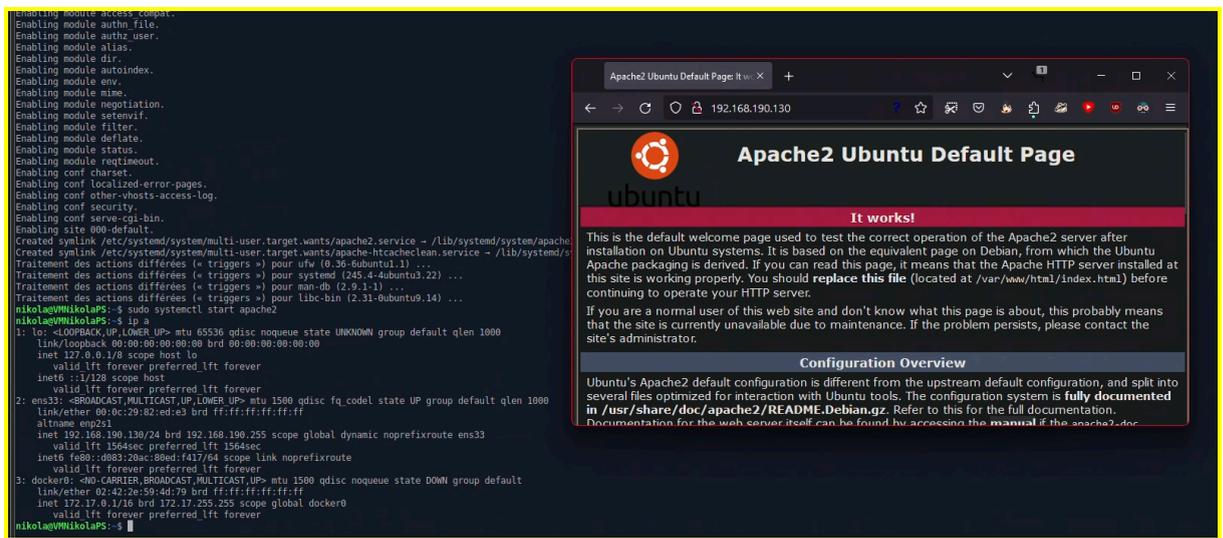
Partie A :Vérification installation d'Apache

1. Lancer la VM Ubuntu et ouvrir une session Terminal

Mise à jour des packages :

`sudo apt update`

Vérifier que le service Apache est bien démarré, pour cela référez-vous à l'annexe A



Le service est bien démarrée, j'ai bien accès au serveur web depuis Firefox

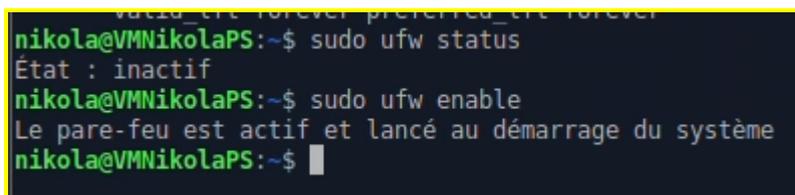
2. Réglage du pare-feu UFW

Vérifiez que le pare-feu est bien actif :

`sudo ufw status`

s'il est inactif, démarrer le via la commande

`sudo ufw enable`

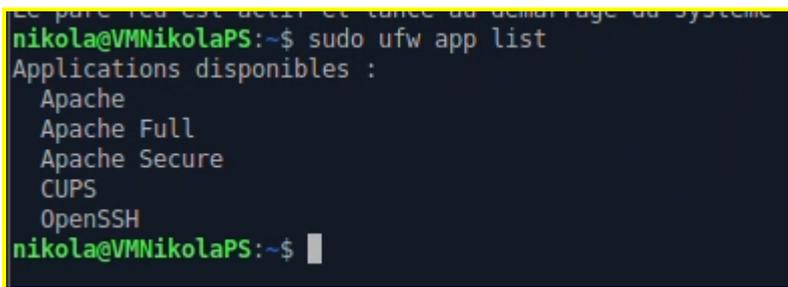


TP Configuration serveur Apache sous Ubuntu

3. Lors de l'installation, Apache s'enregistre auprès de l'UFW pour fournir quelques profils d'application qui peuvent être utilisés pour activer ou désactiver l'accès à Apache à travers le pare-feu.

Listez les profils d'application ufw en tapant :

```
sudo ufw app list
```



```
Le pare-feu est actif et lancé au démarrage du système
nikola@VMNikolaPS:~$ sudo ufw app list
Applications disponibles :
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
nikola@VMNikolaPS:~$
```

Il existe trois profils pour Apache :

Apache : ce profil n'ouvre que le port 80 (trafic web normal, non crypté).

Apache Full : ce profil ouvre à la fois le port 80 (trafic web normal, non crypté) et le port 443 (trafic crypté TLS/SSL).

Apache Secure : ce profil n'ouvre que le port 443 (trafic crypté TLS/SSL).

Pour le moment, on configure l'accès de base pour n'autoriser que le port 80 :

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

Vérifiez que le pare-feu a bien pris en compte le profil via la commande de vérification du statut de UFW

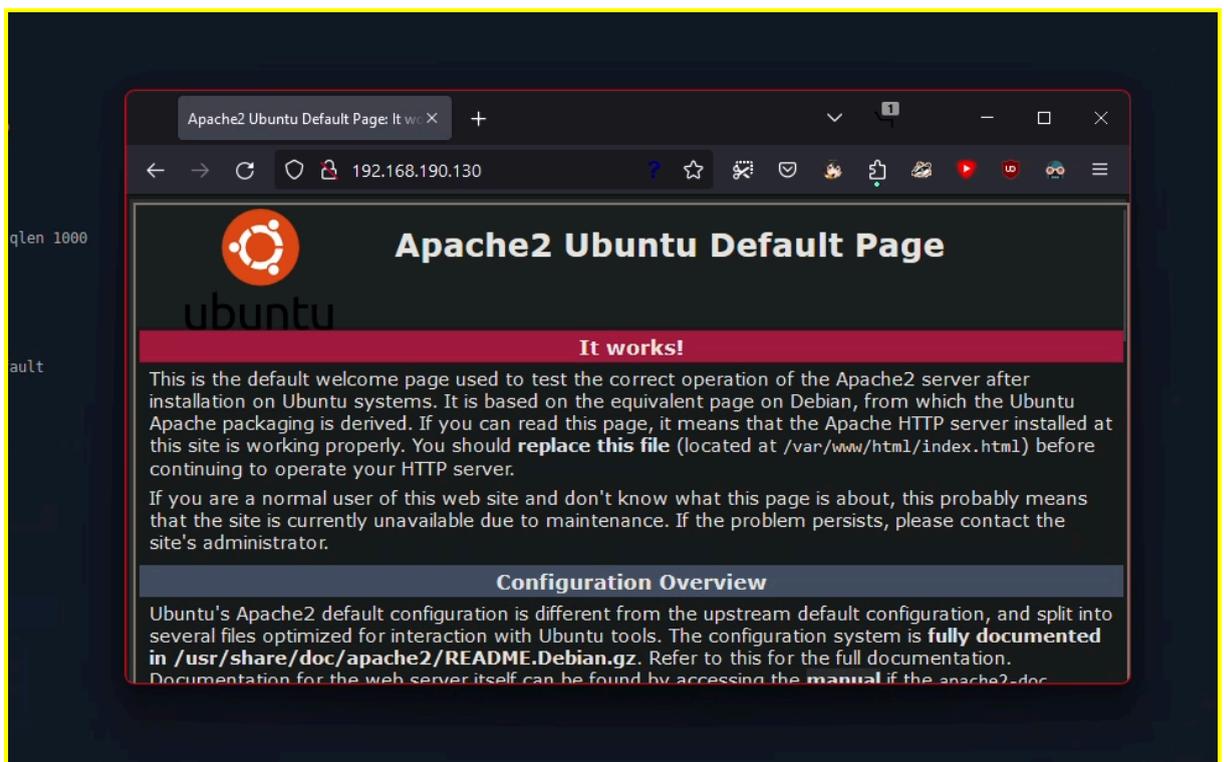
TP Configuration serveur Apache sous Ubuntu

```
OpenSSH
nikola@VMNikolaPS:~$ sudo ufw allow Apache
La règle a été ajoutée
La règle a été ajoutée (v6)
nikola@VMNikolaPS:~$ sudo ufw status
État : actif

Vers          Action      De
----          -
Apache        ALLOW      Anywhere
Apache (v6)   ALLOW      Anywhere (v6)

nikola@VMNikolaPS:~$ █
```

4. Vérifier le fonctionnement du serveur web en ouvrant un navigateur web et en allant sur l'adresse : `http://localhost`
la page "Apache2 Ubuntu Default Page" doit s'afficher



TP Configuration serveur Apache sous Ubuntu

Partie B : Configuration des hôtes virtuels

Lorsque vous utilisez le serveur web Apache, vous pouvez utiliser des hôtes virtuels pour encapsuler les détails de la configuration et héberger plusieurs domaines à partir d'un seul serveur. Nous allons configurer un domaine appelé *Mon_Nom_De_Domaine*, mais vous devez le remplacer par le nom de votre domaine. Ce nom de domaine sera constitué par :

1ère lettre de votre prénom+nom.edu

exemple pour Pierre Martin son domaine serait : pmartin.edu

Apache sur Ubuntu 20.04 a un bloc serveur activé par défaut qui est configuré pour servir des documents à partir du répertoire `/var/www/html`. Même si cela fonctionne bien pour un seul site, cela peut devenir ingérable si vous hébergez plusieurs sites. Au lieu de modifier `/var/www/html`, créons une structure de répertoire au sein de `/var/www` pour un *Mon_Nom_De_Domaine*, en laissant `/var/www/html` en place comme répertoire par défaut qui sera utilisé si une demande du client ne correspond à aucun autre site.

1. Créer le répertoire pour votre domaine dans `/var/www/` (Dans le cas de Pierre Martin, il aurait le répertoire `pmartin.edu` dans `/var/www/`). Vérifier le contenu de `/var/www/`

```
root@VMNikolaPS:/var/www# ls
html  nphilipsinibaldi.edu
root@VMNikolaPS:/var/www#
```

2. Ensuite, changer, pour ce répertoire et de façon récursive, les attributs User+Group afin qu'ils soient égale à votre identifiant (vous pouvez utiliser la variable d'environnement `$USER`)

```
root@VMNikolaPS:/var/www# exit
nikola@VMNikolaPS:~$ sudo chown -R nikola:nikola /var/www/nphilipsinibaldi.edu/
nikola@VMNikolaPS:~$
```

TP Configuration serveur Apache sous Ubuntu

3. Assurez-vous que vous seul avez tous les droits et que le group et others ont seulement les droits en lecture et exécution. Modifier les droits le cas échéant.
4. Vérifier le contenu détaillé de /var/www/

```
nikola@VMNikolaPS:~$ ls -l /var/www
total 8
drwxr-xr-x 2 root  root  4096 janv. 19 11:12 html
drwxr-xr-x 2 nikola nikola 4096 janv. 19 11:17 nphilipsinibaldi.edu
nikola@VMNikolaPS:~$
```

5. Créer un page index.html dans ce répertoire en ouvrant nano par exemple et en ajoutant le code html suivant :

```
<html>
  <head>
    <title>Bienvenue sur mon domaine !</title>
  </head>
  <body>
    <h1> Félicitations ! Ce domaine virtuel fonctionne parfaitement !</h1>
  </body>
</html>
```

6. Pour qu'Apache puisse servir ce contenu, il est nécessaire de créer un fichier d'hôte virtuel avec les directives correctes. Au lieu de modifier directement le fichier de configuration par défaut situé dans /etc/apache2/sites-available/000-default.conf, créer un nouveau fichier *Mon_Nom_De_Domaine.conf* (avec nano par exemple) dans /etc/apache2/sites-available/ et ajouter le contenu suivant en mettant à jour la valeur de *Mon_Nom_De_Domaine* :

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName Mon_Nom_De_Domaine
  ServerAlias www.Mon_Nom_De_Domaine
  DocumentRoot /var/www/Mon_Nom_De_Domaine
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

TP Configuration serveur Apache sous Ubuntu

```
nikola@VMNikolaPS:/etc/apache2$ cat sites-available/nphilipsinibaldi.edu.conf
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/nphilipsinibaldi.edu

    ServerName nphilipsinibaldi.edu
    ServerAlias www.nphilipsinibaldi.edu

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

nikola@VMNikolaPS:/etc/apache2$ █
```

7. Maintenant, il faut activer le site via la commande :

```
sudo a2ensite Mon_Nom_De_Domaine.conf
```

8. Et désactiver le site par défaut défini dans 000-default.conf via la commande :

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

9. Ensuite, faire un test à la recherche d'éventuelles erreurs de configuration via la commande :

```
sudo apache2ctl configtest
```

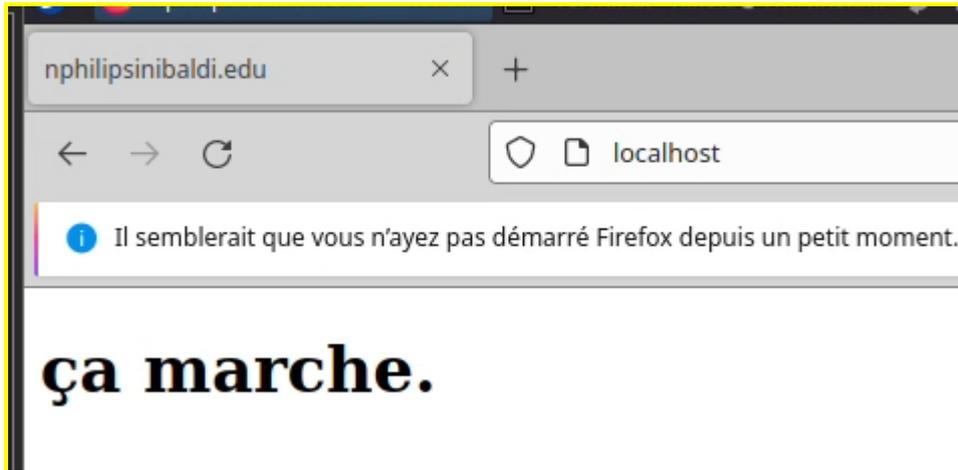
```
nikola@VMNikolaPS:/etc/apache2$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
nikola@VMNikolaPS:/etc/apache2$ █
```

10. Redémarrez Apache pour implémenter les modifications (cf Annexe A)

11. Vérifier la prise en compte des modifications en ouvrant un navigateur web et en allant sur l'adresse : <http://localhost>

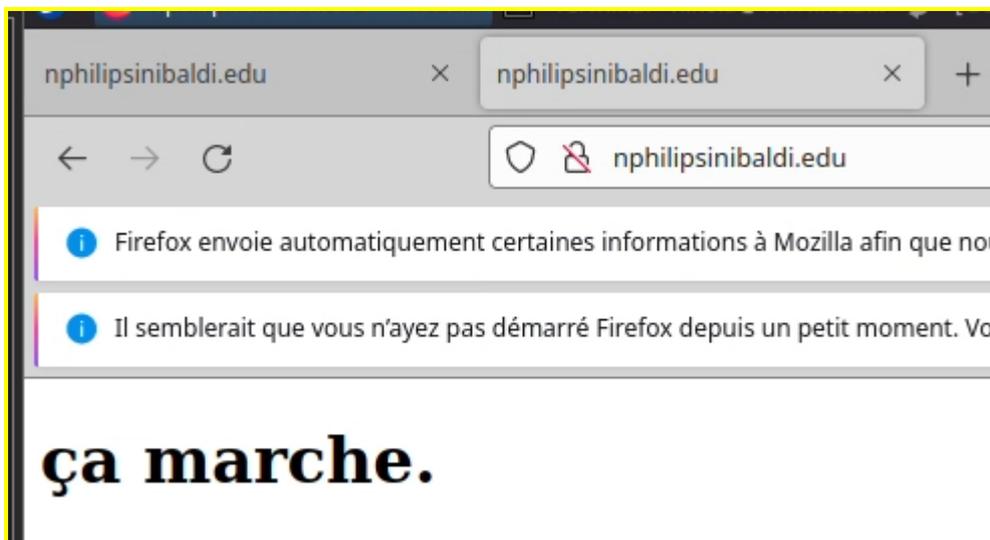
Votre page doit s'afficher

TP Configuration serveur Apache sous Ubuntu



12. Etape bonus, faites les modifications nécessaire afin de pouvoir afficher la page web http://Mon_nom_De_Domaine.edu

vim /etc/hosts puis j'ai insérer cette ligne : 127.0.0.1 nphilipsinibaldi.edu



Annexe A : commandes de gestion de base en utilisant systemctl.

Pour arrêter votre serveur Web, tapez :

```
sudo systemctl stop apache2
```

TP Configuration serveur Apache sous Ubuntu

Pour démarrer le serveur web lorsqu'il est arrêté, tapez :

```
sudo systemctl start apache2
```

Pour arrêter puis redémarrer le service, tapez :

```
sudo systemctl restart apache2
```

Si vous procédez uniquement à des modifications de configuration, il se peut qu'Apache recharge souvent sans interrompre les connexions.

Pour ce faire, utilisez cette commande :

```
sudo systemctl reload apache2
```

Par défaut, Apache est configuré pour un lancement automatique au démarrage du serveur. Si ce n'est pas ce que vous souhaitez, désactiver ce comportement en tapant :

```
sudo systemctl disable apache2
```

Pour réactiver le service de lancement automatique au démarrage, tapez :

```
sudo systemctl enable apache2
```

Désormais, Apache devrait démarrer automatiquement au redémarrage du serveur.