

SAE – 2.03

Installation de services réseaux

Manuel Utilisateur

Table des matières

I)	Configuration d'un serveur web Apache local	2
1)	Mise en place des outils de travail	2
3)	Mettre son site dans le répertoire html.....	3
4)	Démarrage du serveur Apache	4
5)	Modification du nom de domaine	5
6)	Modification des ports.....	7
7)	Commentaire de trame	9
II)	Mise en place d'un serveur Nginx.....	11
1)	Installation de Nginx.....	11
2)	Démarrage du serveur Nginx	11
3)	Mise en place du site web	13
III)	Installation d'un serveur Apache sous Windows.....	15
1)	Mise en place des outils	15
2)	Installation du serveur sur windows	16
IV)	Commande HTTP via telnet.....	18

I) Configuration d'un serveur web Apache local

Dans ce manuel utilisateur nous allons vous apprendre à configurer un serveur web Apache. Dans le cadre de ce manuel nous allons configurer le serveur web sur une machine virtuelle et dans un environnement Ubuntu, qui est une distribution de Linux.

Puis dans un second temps, nous commenterons deux captures de trames, nous configurerons un serveur web Nginx, un serveur Apache sous Windows, et enfin nous afficherons une page à l'aide des méthodes GET et POST

1) Mise en place des outils de travail

Dans un premier temps il faut déterminer si Apache est installé sur votre machine. Pour ce faire il vous suffit de vous rendre dans le répertoire apache2 qui se trouve lui-même dans le répertoire etc. Il vous suffit donc d'exécuter la commande suivante :

```
cd /etc/apache2
```

Si cette commande vous retourne l'erreur suivante c'est qu'il faut installer Apache sur votre machine :

```
gabin@gabin-virtual-machine:~$ cd /etc/apache2
bash: cd: /etc/apache2: No such file or directory
```

Dans ce cas il vous faut exécuter la commande suivante :

```
sudo apt install apache2
```

Ce qui va vous permettre d'installer Apache. Ainsi si vous vous rendez dans le répertoire apache2 il ne devrait plus y avoir d'erreur, et devrait vous donner le résultat suivant :

```
gabin@gabin-virtual-machine:/etc/apache2$
```

Désormais en utilisant la commande ls les fichiers et les répertoires suivants devraient s'afficher à votre écran :

```
gabin@gabin-virtual-machine:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
```

2) Le fichier de configuration

Maintenant que vous vous trouvez dans le répertoire `apache2`, nous allons nous rendre dans le répertoire `sites-available` à l'aide de la commande `cd` pour aller y créer un fichier de type `votre_nom_de_site.conf`. Pour ce faire exécutez la suite de commande suivante :

```
gabin@gabin-virtual-machine:/etc/apache2$ cd sites-available/
gabin@gabin-virtual-machine:/etc/apache2/sites-available$ touch ecodoua.fr
touch: cannot touch 'ecodoua.fr': Permission denied
gabin@gabin-virtual-machine:/etc/apache2/sites-available$ sudo touch ecodoua.conf
[sudo] password for gabin:
gabin@gabin-virtual-machine:/etc/apache2/sites-available$
```

Il faudra ensuite écrire dans ce fichier que nous venons de créer à l'aide de la commande `gedit`. Il faudra donc exécuter la commande suivante, et lorsque la fenêtre de l'éditeur de texte sera ouverte, il faut écrire les éléments suivants :

```
gabin@gabin-virtual-machine:/etc/apache2/sites-available$ sudo gedit ecodoua.conf
```

Puis :

```
1 <VirtualHost *:80>
2     ServerAdmin ecodoua@crozet.com
3     DocumentRoot /var/www/html
4     ServerName ecodoua.com
5     ServerAlias www.ecodoua.com
6
7     <Directory /var/www/html>
8         Options Indexes FollowSymLinks
9         AllowOverride All
10        Require all granted
11    </Directory>
12
13    ErrorLog ${APACHE_LOG_DIR}/error.log
14    CustomLog ${APACHE_LOG_DIR}/access.log combined
15 </VirtualHost>
```

Enfin cliquez en haut à droite sur "Save" avant de fermer l'éditeur de texte.

3) Mettre son site dans le répertoire html

Rendez-vous dans le répertoire `html` à l'aide de la commande suivante :

```
cd /var/www/html
```

Lorsque vous utilisez la commande `ls` dans le répertoire `html`, vous devriez obtenir le résultat suivant :

```
gabin@gabin-virtual-machine:/etc/apache2/sites-available$ cd /var/www/html
gabin@gabin-virtual-machine:/var/www/html$ ls
index.html
gabin@gabin-virtual-machine:/var/www/html$
```

Nous allons ensuite supprimer le fichier `index.html`, pour ensuite copier les fichiers de notre site créé préalablement. Pour ce faire nous allons d'abord utiliser la commande `rm` pour supprimer le fichier `index.html` :

```
gabin@gabin-virtual-machine:/var/www/html$ sudo rm index.html
[sudo] password for gabin:
gabin@gabin-virtual-machine:/var/www/html$
```

Puis, pour copier les fichiers de votre site rendez-vous dans le répertoire dans lequel il se trouve grâce à la commande `cd`, puis copiez le(s) dans le répertoire `html` comme ci-dessous :

```
gabin@gabin-virtual-machine:/var/www/html$ cd /home/gabin/Desktop/ecodoua-master/public
gabin@gabin-virtual-machine:~/Desktop/ecodoua-master/public$ sudo cp -r css images index.html pages /var/www/html
gabin@gabin-virtual-machine:~/Desktop/ecodoua-master/public$
```

Enfin si vous retournez dans le répertoire `html`, et que vous utilisez la commande `ls` vous devriez voir le(s) nom(s) de(s) fichier(s)/répertoire(s) de votre site, comme ci-dessous :

```
gabin@gabin-virtual-machine:~/Desktop/ecodoua-master/public$ cd /var/www/html
gabin@gabin-virtual-machine:/var/www/html$ ls
css  images  index.html  pages
gabin@gabin-virtual-machine:/var/www/html$
```

4) Démarrage du serveur Apache

Pour démarrer votre serveur, il faut que vous vous rendiez de nouveau dans le répertoire `apache2`, pour s'assurer que le port écouté par Apache est bien le port n°80, pour ce faire il faut exécuter la commande suivante :

```
cd /etc/apache2
sudo cat ports.conf
```

Il faut ainsi obtenir ce résultat :

```
gabin@gabin-virtual-machine:/var/www/html$ cd /etc/apache2
gabin@gabin-virtual-machine:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
gabin@gabin-virtual-machine:/etc/apache2$ cat ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
gabin@gabin-virtual-machine:/etc/apache2$
```

Dans le cas où le port écouté n'est pas le port n°80, vous pouvez utiliser la commande `sudo gedit ports.conf` et modifier le numéro de port.

Désormais il ne reste plus qu'à lancer le serveur Apache grâce à la commande `sudo systemctl start apache2`.

Dans le cas où à la suite de cette commande, vous obtenez une erreur, c'est peut être car un autre serveur est déjà lancé par exemple Nginx, dans ce cas il vous suffit d'exécuter la commande `sudo systemctl stop nginx` pour l'arrêter.

Pour observer votre site vous pouvez lancer votre navigateur préféré, et taper `localhost/` dans la barre de recherche pour observer votre site.

5) Modification du nom de domaine

Si vous souhaitez modifier le nom de domaine, c'est-à-dire accéder à dire au lieu de taper "localhost" dans votre barre des tâches taper "ecodoua.local" par exemple, il vous faut suivre les étapes suivantes :

Dans un premier temps modifier le fichier "hosts" qui se trouve dans le répertoire "etc". Pour ce faire vous devrez exécuter la commande suivante : `sudo gedit /etc/hosts`

À la suite de cette commande une fenêtre d'édition de texte va s'ouvrir affichant le contenu suivant :

```
1 127.0.0.1      localhost
2 127.0.1.1      gabin-virtual-machine
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1           ip6-localhost ip6-loopback
6 fe00::0       ip6-localnet
7 ff00::0       ip6-mcastprefix
8 ff02::1       ip6-allnodes
9 ff02::2       ip6-allrouters
```

Vous pourrez ensuite modifier la partie encadrée en rouge pour donner le nom de domaine que vous souhaitez. Dans ce cas nous allons le modifier pour le renommer "ecodoua.local".

Dans notre cas nous obtenons ainsi le résultat suivant :

```
1 127.0.0.1      ecodoua.local
2 127.0.1.1      gabin-virtual-machine
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1           ip6-localhost ip6-loopback
6 fe00::0       ip6-localnet
7 ff00::0       ip6-mcastprefix
8 ff02::1       ip6-allnodes
9 ff02::2       ip6-allrouters
```

Il ne faut pas oublier de cliquer sur le bouton "Save" avant de fermer l'éditeur de texte.

Dans un second temps vous modifierez le fichier nommé "000-default.conf" en utilisant la commande suivante :

```
sudo gedit /etc/apache2/sites-available/000-default.conf
```

Le contenu du fichier avants la modification doit être le suivant :

```
1 VirtualHost *:80
2 # The ServerName directive sets the request scheme, hostname and port that
3 # the server uses to identify itself. This is used when creating
4 # redirection URLs. In the context of virtual hosts, the ServerName
5 # specifies what hostname must appear in the request's Host: header to
6 # match this virtual host. For the default virtual host (this file) this
7 # value is not decisive as it is used as a last resort host regardless.
8 # However, you must set it for any further virtual host explicitly.
9 #ServerName www.example.com
10
11 ServerAdmin webmaster@localhost
12 DocumentRoot /var/www/html
13
14 # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
15 # error, crit, alert, emerg.
16 # It is also possible to configure the loglevel for particular
17 # modules, e.g.
18 #LogLevel info ssl:warn
19
20 ErrorLog ${APACHE_LOG_DIR}/error.log
21 CustomLog ${APACHE_LOG_DIR}/access.log combined
22
23 # For most configuration files from conf-available/, which are
24 # enabled or disabled at a global level, it is possible to
25 # include a line for only one particular virtual host. For example the
26 # following line enables the CGI configuration for this host only
27 # after it has been globally disabled with "a2disconf".
28 #Include conf-available/serve-cgi-bin.conf
29 </VirtualHost>
30
31 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Encore une fois il faudra modifier la partie encadrée en rouge, il faut modifier cette partie par la même chose que dans le fichier "host" vu précédemment.

Dans notre cas ça sera donc "ecodoua.local". On obtiendra ainsi le résultat suivant :

```

1 <VirtualHost *:80>
2     # The ServerName directive sets the request scheme, hostname and port that
3     # the server uses to identify itself. This is used when creating
4     # redirection URLs. In the context of virtual hosts, the ServerName
5     # specifies what hostname must appear in the request's Host: header to
6     # match this virtual host. For the default virtual host (this file) this
7     # value is not decisive as it is used as a last resort host regardless.
8     # However, you must set it for any further virtual host explicitly.
9     #ServerName www.example.com
10
11     ServerAdmin ecodoua.local
12     DocumentRoot /var/www/html
13
14     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
15     # error, crit, alert, emerg.
16     # It is also possible to configure the loglevel for particular
17     # modules, e.g.
18     #LogLevel info ssl:warn
19
20     ErrorLog ${APACHE_LOG_DIR}/error.log
21     CustomLog ${APACHE_LOG_DIR}/access.log combined
22
23     # For most configuration files from conf-available/, which are
24     # enabled or disabled at a global level, it is possible to
25     # include a line for only one particular virtual host. For example the
26     # following line enables the CGI configuration for this host only
27     # after it has been globally disabled with "a2disconf".
28     #Include conf-available/serve-cgi-bin.conf
29 </VirtualHost>
30
31 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

N'oubliez pas de cliquer sur le bouton "Save" en haut à droite de la fenêtre d'édition de texte avant de la fermer.

6) Modification des ports

Dans cette partie nous allons admettre que nous voulons modifier le numéro de port, c'est à dire qu'on souhaite passer du port n°80 au port n°8080 par exemple.

Dans un premier temps il vous faudra vous rendre dans le répertoire "sites-available" pour modifier le fichier .conf que vous avez créé pour votre site.

Dans un second temps c'est le fichier "ports.conf" qu'il faudra modifier.

Pour modifier le fichier .conf que vous avez créée, exécutez la commande suivante :

```
cd /etc/apache2/sites-available
```

Modifiez le fichier `.conf` que vous avez créée plus tôt, dans notre cas c'est le fichier `"ecodoua.conf"`, et modifiez le numéro de port encadré en rouge, grâce à la commande `sudo gedit ecodoua.conf` :

```
1 VirtualHost *:8080
2     ServerAdmin ecodoua@crozet.com
3     DocumentRoot /var/www/html
4     ServerName ecodoua.com
5     ServerAlias www.ecodoua.com
6
7     <Directory /var/www/html>
8         Options Indexes FollowSymLinks
9         AllowOverride All
10        Require all granted
11    </Directory>
12
13    ErrorLog ${APACHE_LOG_DIR}/error.log
14    CustomLog ${APACHE_LOG_DIR}/access.log combined
15 </VirtualHost>
```

Rendez vous ensuite dans le répertoire `"apache2"` à l'aide de la commande `cd ..` :

Vous allez maintenant devoir modifier le fichier `"ports.conf"` avec la commande `sudo gedit ports.conf` :

```
1 # If you just change the port or add more ports here, you will likely also
2 # have to change the VirtualHost statement in
3 # /etc/apache2/sites-enabled/000-default.conf
4
5 Listen 8080
6
7 <IfModule ssl_module>
8     Listen 443
9 </IfModule>
10
11 <IfModule mod_gnutls.c>
12     Listen 443
13 </IfModule>
14
15 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Enfin il faut arrêter, et relancer Apache. Pour ce faire exécutez successivement les commandes suivantes :

```
systemctl stop apache 2
systemctl start apache 2
```

Désormais si vous tapez le nom de votre site, suivi de deux points et du numéro de port, votre site s'affichera. Dans notre cas on tapera donc dans notre barre de recherche : `ecodoua.local:8080`

7) Commentaire de trame

À la suite de la mise en place du serveur web, nous allons utiliser le logiciel de capture de trame Wireshark, et les commenter. Nous allons commenter les captures lorsque le serveur était configuré sur le port 80, et lorsqu'il était configuré sur le port 8080 :

Port 80 :

Requête http :

The image shows a Wireshark capture of an HTTP GET request and its response. The packet list at the top shows two packets: a GET request (frame 14) and a 200 OK response (frame 16). The packet details pane for the selected response packet is shown, with annotations pointing to specific fields:

- Protocole et sa version**: Points to the `HTTP/1.1` field in the request line.
- Nom du site**: Points to the `ecodoua.local` field in the Host header.
- Format des données**: Points to the `gzip` field in the Accept-Encoding header.
- retour attendu**: Points to the `200` status code in the status line.
- Type de données retour attendu**: Points to the `text/html` field in the Content-Type header.

The packet details pane shows the following fields for the selected packet (frame 16):

- Frame 16: 2900 bytes on wire (23200 bits), 2900 bytes captured (23200 bits) on interface lo, id 0**
- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)**
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1**
- Transmission Control Protocol, Src Port: 52782, Dst Port: 80, Seq: 1, Ack: 1, Len: 437**
- Hypertext Transfer Protocol**
 - GET / HTTP/1.1**
 - Host: ecodoua.local**
 - User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:123.0) Gecko/20100101 Firefox/123.0**
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8**
 - Accept-Language: en-US,en;q=0.5**
 - Accept-Encoding: gzip, deflate**
 - Connection: keep-alive**
 - Upgrade-Insecure-Requests: 1**
 - If-Modified-Since: Wed, 13 Mar 2024 13:44:43 GMT**
 - If-None-Match: "1b4b-6138afa28c139-gzip"**
 - [Full request URI: http://ecodoua.local/]**
 - [HTTP request 1/1]**
 - [Response in frame: 16]**

Réponses http :

*Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
14	4.605000617	127.0.0.1	127.0.0.1	HTTP	503	GET / HTTP/1.1
16	4.606214018	127.0.0.1	127.0.0.1	HTTP	2900	HTTP/1.1 200 OK (text/html)

Frame 16: 2900 bytes on wire (23200 bits), 2900 bytes captured (23200 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 80, Dst Port: 52782, Seq: 1, Ack: 438, Len: 2834

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Date: Sun, 24 Mar 2024 15:05:24 GMT\r\n

Server: Apache/2.4.52 (Ubuntu)\r\n

Last-Modified: Wed, 13 Mar 2024 13:44:43 GMT\r\n

Etag: "1b4b-6138afa28c139-gzip"\r\n

Accept-Ranges: bytes\r\n

Vary: Accept-Encoding\r\n

Content-Encoding: gzip\r\n

Content-Length: 2495\r\n

Keep-Alive: timeout=5, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: text/html\r\n

\r\n

[HTTP response 1/1]

[Time since request: 0.001213401 seconds]

[Request in frame: 14]

[Request URI: http://ecodoua.local/]

Content-encoded entity body (gzip): 2495 bytes -> 6987 bytes

File Data: 6987 bytes

Line-based text data: text/html (140 lines)

Port 8080 :

Requête http :

Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000870857	127.0.0.1	127.0.0.1	HTTP	508	GET / HTTP/1.1
6	0.002084223	127.0.0.1	127.0.0.1	HTTP	2900	HTTP/1.1 200 OK (text/html)

Frame 4: 508 bytes on wire (4064 bits), 508 bytes captured (4064 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 48520, Dst Port: 8080, Seq: 1, Ack: 1, Len: 442

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]

Request Method: GET

Request URI: /

Request Version: HTTP/1.1

Host: ecodoua.local:8080\r\n

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:123.0) Gecko/20100101 Firefox/123.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

If-Modified-Since: Wed, 13 Mar 2024 13:44:43 GMT\r\n

If-None-Match: "1b4b-6138afa28c139-gzip"\r\n

\r\n

[Full request URI: http://ecodoua.local:8080/]

[HTTP request 1/1]

[Response in frame: 6]

Réponse http :

Code retour et statut de ce dernier

Date et heure de réponse à la requête

Type de serveur

Format des données renvoyées

La taille de la trame

Type des données retour

II) Mise en place d'un serveur Nginx

1) Installation de Nginx

Tout d'abord assurez-vous que Nginx est installé, pour ce faire exécutez la commande suivante : `systemctl status nginx`

Si l'erreur retournée est la suivante, il faut installer Nginx :
Unit nginx.service could not be found.

Vous pourrez ainsi installer Nginx à l'aide de la commande suivante :
`sudo apt install nginx`

2) Démarrage du serveur Nginx

Avants de démarrer le serveur Nginx il faut s'assurer qu'il n'y ait pas d'autres serveur activé, comme un serveur Apache par exemple. On

effectue donc la commande suivante :
systemctl status apache2

Dans notre cas on obtient le résultat suivant :

```
gabin@gabin-virtual-machine:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-03-25 13:02:39 CET; 16min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 909 (apache2)
      Tasks: 55 (limit: 4556)
    Memory: 8.1M
       CPU: 242ms
    CGroup: /system.slice/apache2.service
            └─909 /usr/sbin/apache2 -k start
              └─912 /usr/sbin/apache2 -k start
                └─913 /usr/sbin/apache2 -k start

mars 25 13:02:38 gabin-virtual-machine systemd[1]: Starting The Apache HTTP Server...
mars 25 13:02:39 gabin-virtual-machine apache2[877]: AH00558: apache2: Could not reliably
mars 25 13:02:39 gabin-virtual-machine systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

On voit qu'un serveur Apache est actif, il faut donc le désactiver avant d'activer le serveur Nginx. Dans le cas où le serveur n'est pas actif il n'y aura pas besoin d'exécuter la commande suivante :
systemctl stop apache2

Maintenant que le serveur est désactivé vous allez pouvoir activer le serveur Nginx à l'aide de la commande suivante :
systemctl start nginx

Pour vous assurer que le serveur Nginx est bien actif vous pouvez exécuter la commande `systemctl status nginx`. Vous devrez ainsi obtenir le résultat suivant :

```
guillaume@guillaume-virtual-machine:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:
   Active: active (running) since Sun 2024-03-24 10:27:20 CET; 13min ago
```

Désormais si vous tapez dans la barre de recherche de votre moteur de recherche "127.0.0.1" vous devrez obtenir le résultat suivant :

🔍 127.0.0.1

☆

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

3) Mise en place du site web

A la suite de l'installation de Nginx, un fichier nommé "index.nginx-debian.html" a été créé. Il va falloir le supprimer, pour ce faire rendez-vous dans le répertoire de ce fichier, et supprimez-les en exécutant successivement les commandes suivantes :

```
cd /var/www/html
sudo rm index.nginx-debian.html
```

```
guillaume@guillaume-virtual-machine:~$ cd /var/www/html
guillaume@guillaume-virtual-machine:/var/www/html$ sudo rm -r index.nginx-debian.html
```

Ensuite pour que votre site s'affiche il faut que vous copiez les fichiers et les répertoires de votre site dans le répertoire "html" dans lequel vous vous trouvez à la suite des commandes précédentes, sans oublier qu'il faut qu'il y ait un fichier "index.html" dans les fichiers qui composent votre site, ce sera la page chargée en premier.

Pour ce faire déplacez-vous dans le répertoire de votre site à l'aide de la commande "rm" comme ci-dessous :

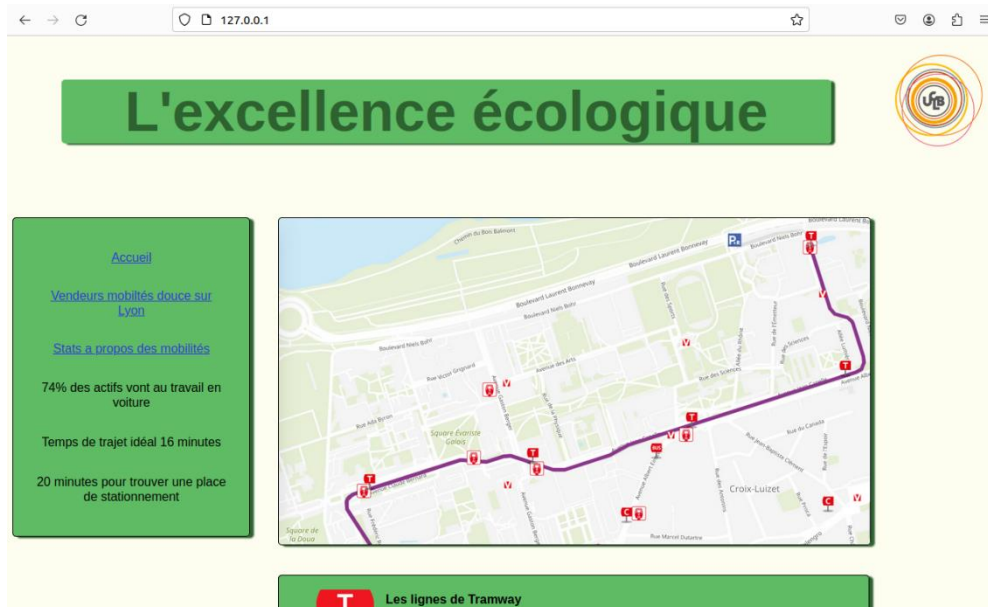
```
rm /Desktop/mon_site_web
```

Puis exécutez la commande suivante :

```
sudo cp -r votre_index.html votre_page2.html votre_page3.html
votre_style.css /var/www/html
```

```
guillaume@guillaume-virtual-machine:~/Desktop$ cd ~/Desktop/mon_site_web/
guillaume@guillaume-virtual-machine:~/Desktop/mon_site_web$ sudo cp -r image/ index.html page2.html page3.html style.css style-p2.css style-p3.css /var/www/html
```

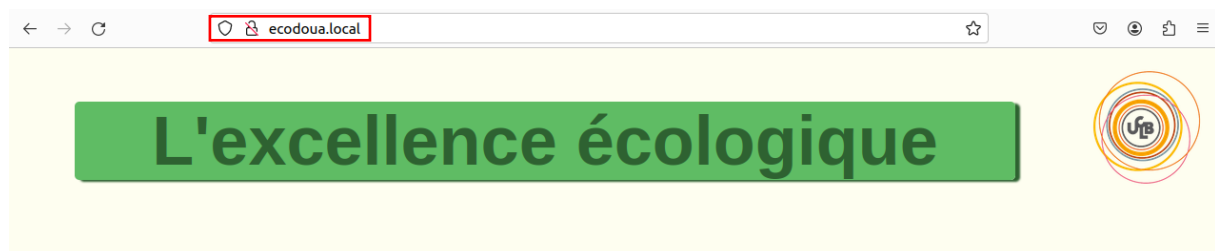
Félicitation vous venez de paramétrer votre serveur web Nginx.
Maintenant si vous tapez dans la barre de recherche de votre moteur de recherche "127.0.0.1", vous pourrez voir votre site s'afficher.



Nous allons ensuite modifier le nom de domaine il faut faire la commande
`sudo gedit /etc/hosts` puis ajouter cette ligne au fichier hosts.

```
1 127.0.0.1    ecodoua.local
2 127.0.1.1    guillaume-virtual-machine
3 0.0.0.0      ecodoua.local www.ecodoua.local
4
5 # The following lines are desirable for IPv6 capable hosts
6 ::1          ip6-localhost ip6-loopback
7 fe00::0      ip6-localnet
8 ff00::0      ip6-mcastprefix
9 ff02::1      ip6-allnodes
10 ff02::2      ip6-allrouters
```

Rafraichissez la page, votre nom de domaine apparaîtra

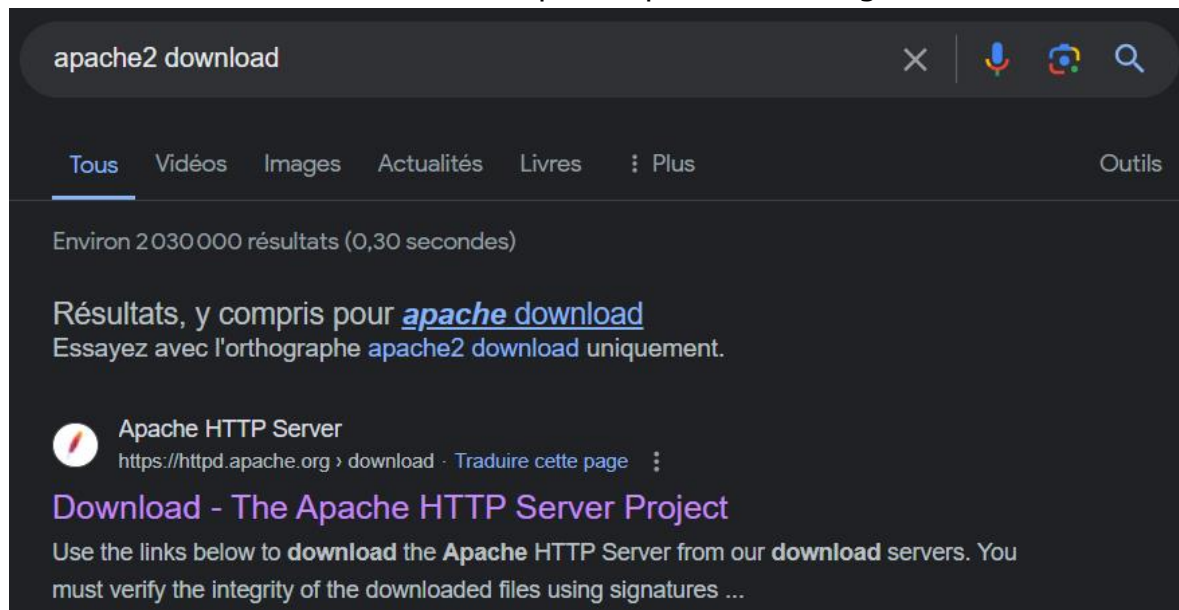


Bravo vous avez configuré un serveur local !

III) Installation d'un serveur Apache sous Windows

1) Mise en place des outils

Rendez-vous sur le site d'apache pour télécharger ce dernier :



Puis sélectionnez la dernière version d'Apache pour Windows, sélectionnez la version "Lounge" qui correspond à votre architecture :

Apache 2.4 binaries VS17

[Info & Changelog](#)

Apache 2.4.58 Win64

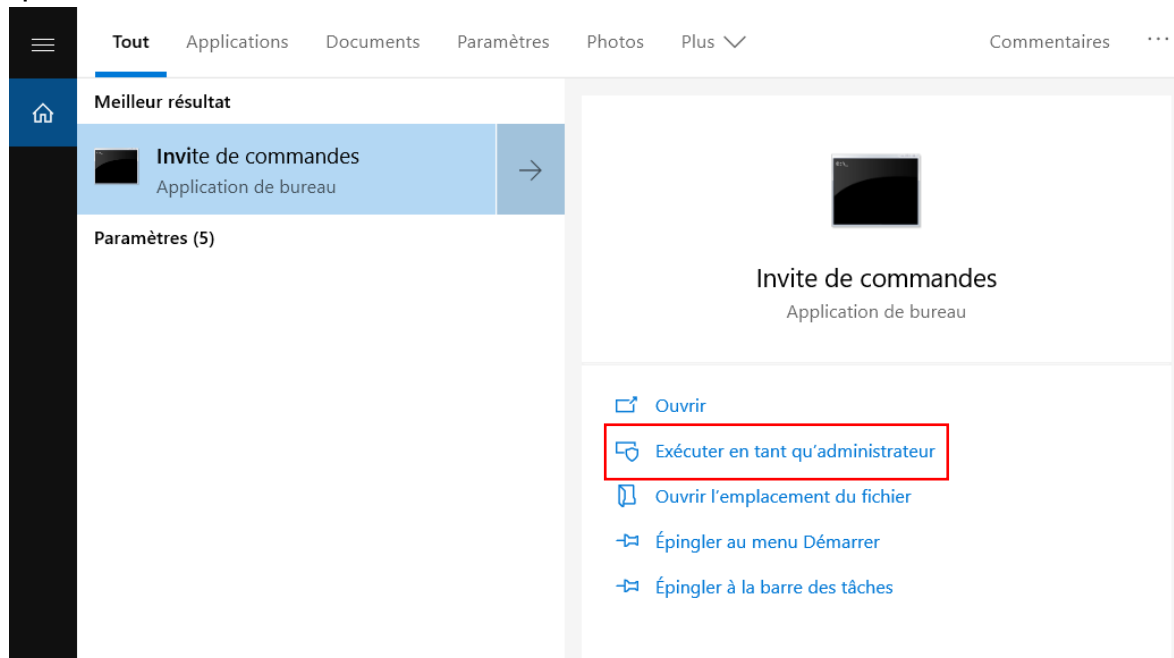
● [httpd-2.4.58-240131-win64-vs17.zip](#) 31 Jan '24 11.398k
[PGP Signature](#) (Public [PGP key](#)), SHA1-SHA512 [Checksums](#)

Apache 2.4.58-240131 Win32

● [httpd-2.4.58-240131-win32-vs17.zip](#) 31 Jan '24 10.244k
[PGP Signature](#) (Public [PGP key](#)), SHA1-SHA512 [Checksums](#)

To be sure that a download is intact and has not been tampered with, use PGP, see [PGP Signature](#)

Une fois téléchargé, extrayez le contenu du fichier .zip obtenu dans le répertoire de votre choix, puis ouvrez votre invite de commande en tant qu'administrateur :



Lorsque l'invite de commande est ouverte rendez-vous dans le répertoire "bin" de apache24 avec la commande suivante :

```
cd c:\Apache24\bin.
```

Exécutez ensuite la commande `httpd -k install` pour installer Apache.

Vous devriez ainsi obtenir le résultat suivant :

```
C:\Windows\system32>cd c:\Apache24\bin

c:\Apache24\bin>httpd -k install
Installing the 'Apache2.4' service
The 'Apache2.4' service is successfully installed.
```

2) Installation du serveur sur windows

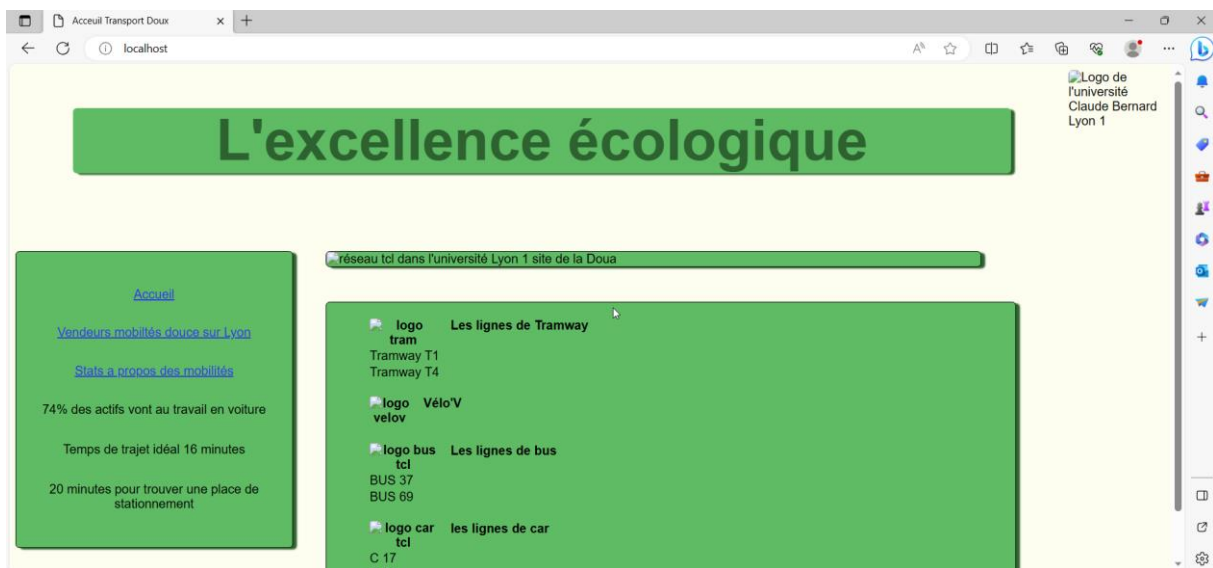
Allez ensuite dans votre dossier où est situé votre site internet avec la commande `cd` puis faites la copie des fichiers et des répertoires de votre site dans le répertoire "htdocs" à l'aide de la commande suivante :

```
copy * c:\Apache24\htdocs
```


Vous devriez obtenir le résultat suivant :

```
c:\site>copy * c:\Apache24\htdocs
index.html
Remplacer c:\Apache24\htdocs\index.html (Oui/Non/Tous) : oui
page2.html
page3.html
style-p2.css
style-p3.css
style.css
        6 fichier(s) copié(s).
```

Pour terminer si vous tapez dans votre barre de recherche "localhost", vous pourrez alors voir votre site.



Bravo vous venez de paramétré votre serveur web Apache2 sur Windows !

IV) Commande HTTP via telnet

Installation de Telnet avec la commande : `sudo apt-get install telnet`. Il faut préalablement démarrer votre serveur (se référer IV)Démarrage du serveur Apache2) avec votre site web sur un serveur connu (par exemple le 80).

```
guillaume@guillaume-virtual-machine:~$ sudo apt-get install telnet
```

Puis dans un second temps, il faut se connecter au serveur Apache2 grâce à la commande `telnet localhost 80`, localhost est le nom de domaine de votre site ou alors son adresse IP ce qui permettra de nous connecter à celui-ci. Son port (ici est le port 80).

```
guillaume@guillaume-virtual-machine:~$ telnet localhost 80
Trying ::1...
Connected to localhost.localdomain.
Escape character is '^['.
```

A la suite de ceci, vous devrez écrire

```
GET / HTTP/1.1
Host: localhost
```

Cela va envoyer une requête au serveur et il va alors donner le code html de la page.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Accueil Transport Doux</title>
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <header>
      <h1>L'excellence écologique</h1>
      
    </header>

    <aside id="aside">

      <a href="index.html">Accueil</a>
      <a href="page3.html">Vendeurs mobiltés douce sur Lyon</a>
      <a href="page2.html">Stats a propos des mobilités</a>
```