

Alaoui Smaili Mèjdi

Senelet Amadis

8 mars 2024

Manuel utilisateur - Déploiement site web

Le but de ce manuel est de vous apprendre comment déployer un site Web, en premier lieu sous Linux grâce à Apache, puis grâce à Nginx mais aussi de vous apprendre des notions techniques sur les serveurs.

Apache et Nginx sont les 2 serveurs HTTP les plus connus sous Linux permettant d'héberger des ressources de type web. L'installation et configuration diffère de peu mais le principe reste le même.

Sommaire

Manuel utilisateur - Déploiement site web.....	1
Préalable.....	2
I- Déployer avec Apache.....	4
II- Documentation technique.....	8
a. Affichage des trames HTTP sur le port 80.....	8
b. Changement du port 80 au port 8080.....	11
c. Accéder aux trames via la méthode telnet.....	12
III- Intégrer le principe du DNS sur votre site web en local.....	14
IV- Déployer avec Nginx.....	16

Point Important

Sur une même machine, les deux types de serveurs peuvent être installés mais un seul des deux peut être actif en même temps.

Préalable

1. Si vous n'avez ni Apache ni Nginx installés, veuillez exécuter les deux commandes suivantes pour les installer.

```
sudo apt install apache2
```

```
sudo apt install nginx
```

La commande "sudo" est utilisée pour exécuter une commande avec les privilèges d'un autre utilisateur, généralement l'utilisateur "root" (l'administrateur du système). Certaines commandes nécessitent des privilèges d'administration pour accéder ou modifier des fichiers système, des configurations système (comme notre cas actuel d'où son utilisation ou des ressources protégées).

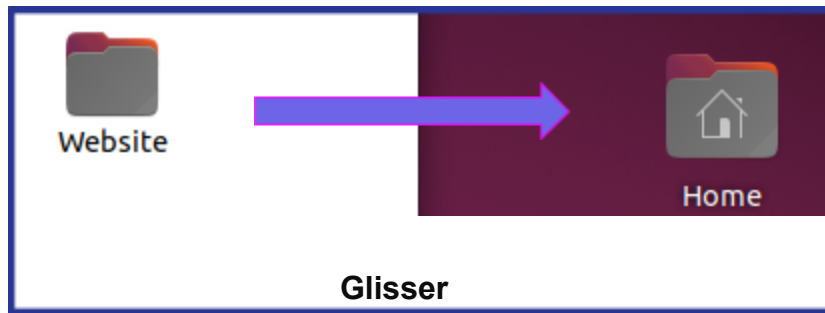
⚠ Après avoir exécuté cette commande, le système vérifie votre identité en tant qu'administrateur et vous demande immédiatement le mot de passe associé à votre session d'administration. De plus, lorsque vous saisissez le mot de passe, il n'apparaît pas à l'écran pour des raisons de sécurité, mais vous pouvez le taper normalement :

```
[sudo] password for devops:
```

Une confirmation est demandée. **Veuillez** taper 'Y' pour confirmer.

```
Do you want to continue? [Y/n] y
```

2. Si votre site web est sur votre clé USB, **veuillez** le glisser dans le répertoire “Home”, (ou dans un emplacement de votre PC dont vous connaissez le chemin d'accès pour les plus expérimentés).



Nous sommes désormais prêts à démarrer. Toutes les opérations se déroulent en ligne de commande, **veuillez** donc ouvrir votre terminal.

I- Déployer avec Apache

Dans cette section, nous vous expliquerons comment déployer votre site web en utilisant le serveur Apache. Tout d'abord, nous vérifierons que Nginx n'est pas actif ([voir Point Important](#)) :

```
systemctl stop nginx
```

1- **Placez** vous dans le dossier d'Apache

```
cd /etc/apache2
```

La commande "cd" permet de changer de répertoire dans le système de fichiers.

2- **Vérifiez** que les dossiers concordent bien avec l'image ci-dessous, pour ce faire tapez "ls" dans la ligne de commande.

```
apache2.conf  conf-enabled  magic          mods-enabled  sites-available  
conf-available  envvars      mods-available ports.conf    sites-enabled
```

La commande "ls" est utilisée pour lister les fichiers et répertoires dans le répertoire actuel

```
cd sites-available
```

3- **Placez** vous désormais dans le fichier "sites-available"

Le répertoire "site-available" est l'endroit où vous pouvez placer des fichiers de configuration pour les sites web disponibles sur votre serveur, mais qui ne sont pas encore activés.

5

4- **Vérifiez** que les dossiers concordent bien avec l'image ci-dessous, pour ce faire tapez "ls" dans la ligne de commande.

```
000-default.conf  default-ssl.conf
```

5- **Créez** une copie du fichier 000-default.conf par un nouveau_nom.conf, nous utiliserons 001-default.conf.

```
devops@ubuntu:/etc/apache2/sites-available$ sudo cp 000-default.conf 001-default.conf
```

La commande "cp" est utilisée pour copier des fichiers et des répertoires d'un emplacement à un autre.

6- **Retournez** dans votre répertoire personnel (grâce à "~").



```
cd ~
```

7- **Dirigez-vous** vers le répertoire où se trouve votre site web. Dans notre cas, il sera situé dans le répertoire "Home", et grâce à la commande précédente, nous y sommes déjà.

8- **Déplacez** le dossier contenant votre site vers le répertoire "/var/www/". Dans notre cas, le dossier s'appelle "SAE-WEB-ALAOUI-SEN" et il sera renommé en "SAE_site" dans le répertoire "/var/www/". Vous pouvez choisir de conserver le même nom que le dossier d'origine ou d'en choisir un nouveau.

```
sudo mv SAE-WEB-ALAOUI-SEN/ /var/www/SAE_site
```

La commande "mv" en Linux est utilisée pour déplacer ou renommer des fichiers et des répertoires.

Le répertoire "/var/www/" est l'endroit où les fichiers de contenu web sont stockés et servis aux utilisateurs via le serveur web.

9- **Retournez** dans le répertoire sites-available, (ci-dessous pour le faire en une commande)

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

10- Nous allons désormais modifier certaines informations pour qu'elles concordent avec notre dossier de notre site web.

```
sudo nano 000-default.conf
```

"Nano" possède plusieurs utilisations, dans notre cas nous l'utilisons pour ouvrir et de modifier des fichiers texte directement dans le terminal (en l'occurrence le fichier 000-default.conf).

Après avoir **tapé** la commande, une page texte s'affiche.

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
Read 31 lines
```

11- Modifiez les informations

```
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
```

```
#ServerName www.nomsitedevotrechoix.com

ServerAdmin adresse_mail_admin
DocumentRoot /var/www/SAE_site
```

Pour la **ligne 3**, “SAE_site” doit correspondre au nom que vous avez indiqué lors de l’étape 8.

12- **Enregistrez** vos modifications en faisant “Ctrl+X”. Vous aurez cela qui s’affiche :

```
Save modified buffer?
Y Yes
N No      ^C Cancel
```

Pour confirmer **tapez** sur la touche “Y”, cela affichera :

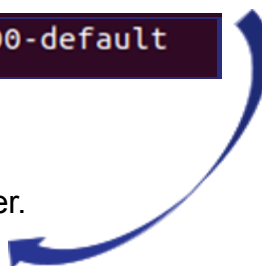
```
File Name to Write: 000-default.conf
^G Help                M-D DOS Format
^C Cancel              M-M Mac Format
```

Enfin, **tapez** sur la touche “Entrée”.

13- **Activez** le site avec :

```
a2ensite 000-default
```

⚠ A ne pas écrire le “.conf” dans le nom du dossier.



```
Site 000-default already enabled
```

Voici la réponse du terminal que vous **devez** avoir.

14- Après chaque modification, le serveur **doit** être relancé.

```
systemctl reload apache2
```

15-**Vérifiez** sur un navigateur web en tapant “localhost” que votre site est bien déployé !

II- Documentation technique

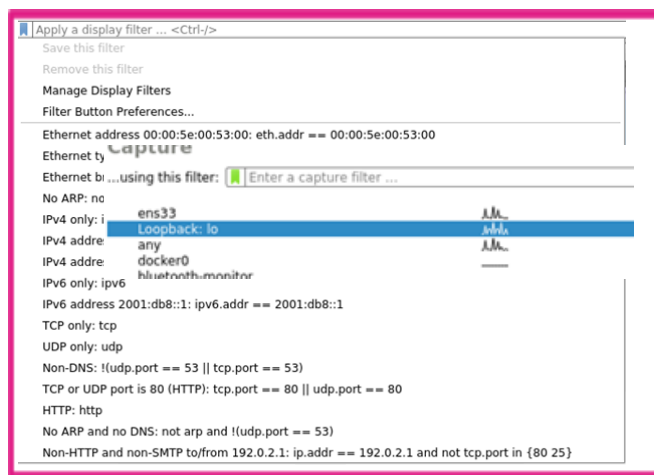
Arrivé ici, votre site est normalement déployé sur Apache. Cependant, il peut être intéressant de savoir plus précisément ce qui a réellement lieu lors des échanges utilisateur-serveur. Nous allons pouvoir étudier ces échanges via la lecture de trames HTTP.

a. Affichage des trames HTTP sur le port 80.

Une fois le site déployé, il vous suffit de rentrer la commande suivante dans le terminal :

```
devops@ubuntu:~$ sudo wireshark
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

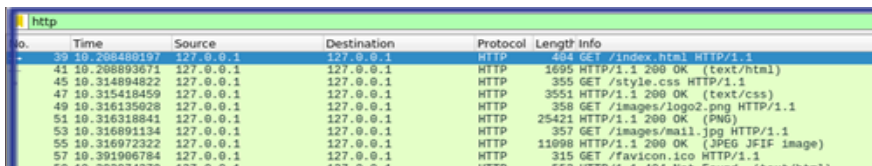
Une fois la commande lancée, le logiciel Wireshark se lance. Lorsqu'il est totalement lancé, **sélectionnez** le menu déroulant présent en haut à gauche du logiciel, il permet de filtrer les trames reçues:



Dans ce menu déroulant, **sélectionnez** "HTTP : http" puisque nous sommes intéressés par les trames HTTP. Une fois cela fait, double-cliquez sur la ligne "Loopback: lo".

Une fois que la nouvelle fenêtre s'affiche, vous **devez** lancer votre site sur le navigateur de votre choix, en entrant dans la barre d'URL : "localhost". Une fois la page de votre site s'étant affichée, **revenez** sur wireshark et vous pourrez maintenant voir différentes lignes s'afficher sur le logiciel.

Dans le bloc rectangulaire du haut, **sélectionnez** la première ligne.



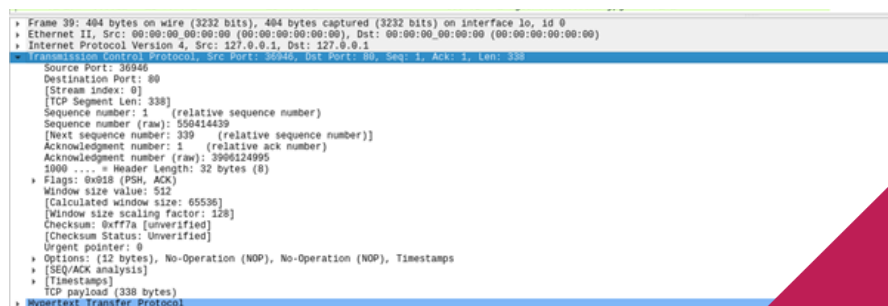
No.	Time	Source	Destination	Protocol	Length	Info
39	0.000000	127.0.0.1	127.0.0.1	HTTP	338	GET /index.html HTTP/1.1
40	0.000000	127.0.0.1	127.0.0.1	HTTP	1695	HTTP/1.1 200 OK (text/html)
41	0.000000	127.0.0.1	127.0.0.1	HTTP	355	GET /style.css HTTP/1.1
42	0.000000	127.0.0.1	127.0.0.1	HTTP	355	HTTP/1.1 200 OK (text/css)
43	0.000000	127.0.0.1	127.0.0.1	HTTP	358	GET /images/logo2.png HTTP/1.1
44	0.000000	127.0.0.1	127.0.0.1	HTTP	25421	HTTP/1.1 200 OK (PNG)
45	0.000000	127.0.0.1	127.0.0.1	HTTP	357	GET /images/mail.jpg HTTP/1.1
46	0.000000	127.0.0.1	127.0.0.1	HTTP	11098	HTTP/1.1 200 OK (JPEG JFIF image)
47	0.000000	127.0.0.1	127.0.0.1	HTTP	315	GET /favicon.ico HTTP/1.1

Cette ligne représente la requête envoyée par l'utilisateur au serveur, lorsque vous avez écrit "localhost" afin d'accéder à votre site. Dans les différents menus déroulants apparaissant dans la partie inférieure du logiciel, développez celui intitulé "Hypertext Transfer Protocol".



On peut voir plusieurs choses sur cette trame. Dans un premier temps, on reconnaît le titre du site ("**Host : localhost\r\n**"), mais on a également le langage attendu pour le site (ligne "**Accept-Language**"), et le type de données acceptées (ligne "**Accept**").

Il peut également être intéressant de développer le menu juste au-dessus, "**Transmission Control Protocol**".



Ici, on peut identifier le port associé à votre site, le port de base étant 80. Cette partie là sera modifiée lorsque nous déplacerons le port du site sur le port 8080.

Une deuxième trame, encore plus importante que celle-ci, est celle représentant la réponse du serveur à votre requête. Elle se situe juste en dessous de la trame précédente.

No.	Time	Source	Destination	Protocol	Length	Info
39	10.208480197	127.0.0.1	127.0.0.1	HTTP	404	GET /index.html HTTP/1.1
40	10.208480201	127.0.0.1	127.0.0.1	HTTP	100	200 OK (text/html)
45	10.314094822	127.0.0.1	127.0.0.1	HTTP	355	GET /style.css HTTP/1.1
47	10.315418459	127.0.0.1	127.0.0.1	HTTP	355	HTTP/1.1 200 OK (text/css)
49	10.316135028	127.0.0.1	127.0.0.1	HTTP	358	GET /images/logo2.png HTTP/1.1
51	10.316318841	127.0.0.1	127.0.0.1	HTTP	25421	HTTP/1.1 200 OK (PNG)
53	10.316891134	127.0.0.1	127.0.0.1	HTTP	357	GET /images/mail.jpg HTTP/1.1
55	10.316972322	127.0.0.1	127.0.0.1	HTTP	11098	HTTP/1.1 200 OK (JPEG JFIF image)
57	10.391906784	127.0.0.1	127.0.0.1	HTTP	315	GET /favicon.ico HTTP/1.1
59	10.392074379	127.0.0.1	127.0.0.1	HTTP	553	HTTP/1.1 404 Not Found (text/html)
61	10.445658214	127.0.0.1	127.0.0.1	HTTP	363	GET /images/backaccueil.jpg HTTP/1.1

De la même manière que précédemment, développer le menu "Hypertext Transfer Protocol".

```

* Frame 41: 1095 bytes on wire (1369 bits), 1095 bytes captured (1369 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: 30946, Seq: 1, Ack: 339, Len: 1629
  * HTTP/1.1 200 OK\r\n
    Date: Thu, 14 Mar 2024 07:32:07 GMT\r\n
    Server: Apache/2.4.41 (Ubuntu)\r\n
    ETag: "c84-60f502d0eaa0-gzip"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Content-Length: 1291\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 1/6]
    [Time since request: 0.000413474 seconds]
    [Request in frame: 39]
    [Next request in frame: 45]
    [Next response in frame: 47]
    [Request URI: http://localhost/index.html]
    Content-encoded entity body (gzip): 1291 bytes -> 3204 bytes
    File Data: 3204 bytes
    * Line-based text data: text/html (89 lines)
  
```

Dans un premier temps, il est important de noter la réponse du serveur, le chiffre situé entre HTTP/1.1 et \r\n. Votre site sera fonctionnel uniquement si ce chiffre est de 200 ou 204 (-> le code de retour 204 signifie que le serveur a reçu la requête mais il n'y a pas d'information à renvoyer). Sinon, une erreur aura lieu à un moment dans le déploiement de votre site. Si vous avez suivi ce manuel à la lettre, le chiffre 200 devrait apparaître et votre site sera fonctionnel.

On peut également voir la date de cette réponse du serveur à la ligne d'en dessous, et enfin le serveur sur lequel votre site est déployé.

Voici donc les deux trames vraiment importantes à regarder pour vérifier que votre site est correctement déployé.

b. Changement du port 80 au port 8080.

Vous avez maintenant vérifié que votre site était correctement déployé sur votre serveur via les trames, vous allez maintenant pouvoir modifier le port destination de base de votre site.

Pour commencer, **retournez** dans le répertoire "sites-available" afin de modifier le fichier correspondant à votre site, en l'occurrence le fichier "000-default.conf". Vous **devrez** répéter les étapes 9 et 10 de la **partie I**.

Pour rappel voici les deux commandes à effectuer :

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

```
sudo nano 000-default.conf
```

Une fois

dans le fichier "000-default.conf", modifiez la ligne "<VirtualHost *:80>" en "<VirtualHost *:8080>". Une fois ceci fait, sortez en **enregistra**nt de la même manière que précédemment (Cf. *partie I* étape 12).

```
GNU nano 4.8                                000-
<VirtualHost *:8080>
    # The ServerName directive se
    # the server uses to identifi
```

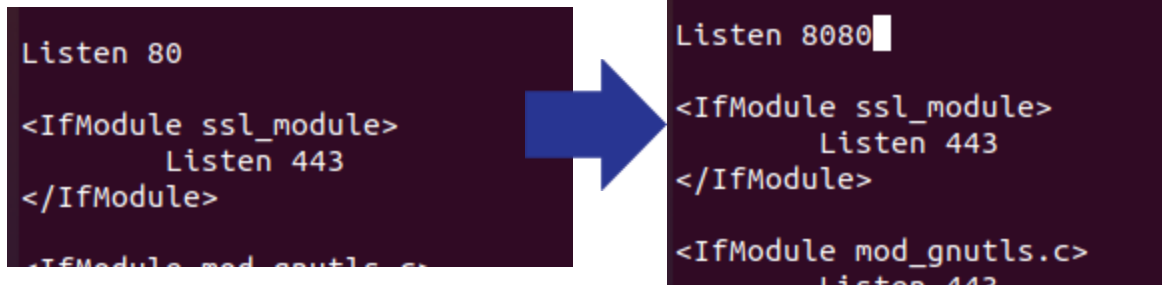
A l'aide de la commande "cd ../", **revenez** au répertoire de base de Apache. **Vérifiez** alors que le fichier "ports.conf" est bien présent (grâce à "ls").

```
devops@ubuntu:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
```

A l'aide de la même commande que précédemment, **ouvrez** ce fichier.

```
devops@ubuntu:/etc/apache2$ sudo nano ports.conf
```

Une fois dans le fichier, **changez** comme précédemment le 80 par 8080.



Une fois cela fini et **enregistré** (Cf. *partie I* étape 12), vous n'aurez plus qu'à recharger Apache avec la commande vue précédemment.

```
systemctl reload apache2
```

Dernière différence, lorsque vous voudrez accéder à votre site sur votre navigateur, il sera maintenant nécessaire d'écrire "localhost:8080" au lieu de simplement "localhost".

c. Accéder aux trames via la méthode telnet.

Il existe une autre méthode pour récupérer les réponses du serveur, sans passer par Wireshark. Cependant, elle nécessite d'avoir passé l'étape "**III- Intégrer le principe du DNS sur votre site web en local**", et d'avoir changé son nom de domaine en un ".local".

```
devops@ubuntu:/etc/apache2$ sudo telnet unidoux.local 8080
```

Supposons l'étape III déjà faite. Dans votre console de commande, **entrez** la commande :

A la place de "unidoux.local", **entrez** votre site, et 8080 correspond au port associé à votre site.

Ensuite, **entrez** l'une des deux commandes suivantes, si vous voulez la requête de l'utilisateur (**GET / HTTP/1.1**) ou la réponse du serveur (**POST / HTTP/1.1**) :

```
Trying 127.0.0.1...
Connected to unidoux.local.
Escape character is '^]'.
GET / HTTP/1.1
```

```
Trying 127.0.0.1...
Connected to unidoux.local.
Escape character is '^]'.
POST / HTTP/1.1
```

Une fois les deux commandes rentrées, **appuyez** une fois sur “Entrée”, puis **rentrez** la

```
Host: unidoux.local
```

deuxième ligne (**Host: unidoux.local**) :

Appuyez ensuite deux fois sur “Entrée”, les deux trames voulues s'affichent alors :

```
Trying 127.0.0.1...
Connected to unidoux.local.
Escape character is '^]'.
GET / HTTP/1.1
Host: unidoux.local

HTTP/1.1 200 OK
Date: Thu, 14 Mar 2024 08:37:07 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Fri, 19 Jan 2024 17:54:48 GMT
ETag: "c84-60f502d0eaa00"
Accept-Ranges: bytes
Content-Length: 3204
Vary: Accept-Encoding
Content-Type: text/html
```

```
Trying 127.0.0.1...
Connected to unidoux.local.
Escape character is '^]'.
POST / HTTP/1.1
Host: unidoux.local

HTTP/1.1 200 OK
Date: Thu, 14 Mar 2024 08:39:00 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Fri, 19 Jan 2024 17:54:48 GMT
ETag: "c84-60f502d0eaa00"
Accept-Ranges: bytes
Content-Length: 3204
Vary: Accept-Encoding
Content-Type: text/html
```

III- Intégrer le principe du DNS sur votre site web en local

Une fois que vous avez déployé votre site web sur Apache, nous allons maintenant ajouter la fonctionnalité DNS en le faisant apparaître sous un domaine différent de celui par défaut d'Apache en local.

Pour commencer, **retournez** dans le répertoire "sites-available" afin de modifier le fichier correspondant à votre site, en l'occurrence le fichier "000-default.conf". Vous **devrez** répéter les étapes 9 et 10 de la **partie I**.

Pour rappel voici les deux commandes à effectuer :

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

```
sudo nano 000-default.conf
```

1. **Rajoutez** cette **ligne** avec **nomdusite.local**, et **enregistrez** les modifications (Cf. *partie I* étape 12).

```
ServerAdmin webmaster@localhost
ServerName unidoux.local
DocumentRoot /var/www/SAE_site
```

2. **Modifiez** le fichier "hosts".

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

Le fichier "/etc/hosts" est un fichier de configuration système utilisé pour associer des adresses IP à des noms de domaine.

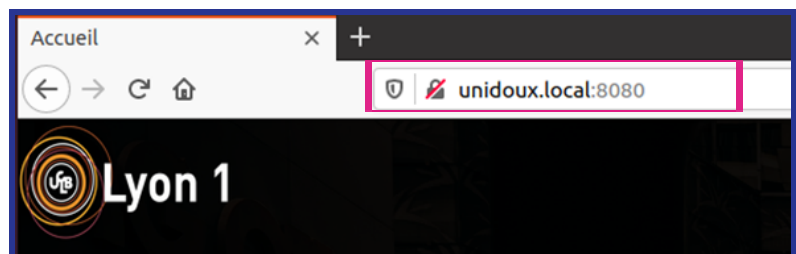
```
127.0.0.1      localhost
127.0.1.1      ubuntu

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

Cela vous amènera à ce fichier :

3. **Modifiez** '*localhost*' par le nom que vous avez mis dans l'étape 1 (à nouveau avec le **.local** à la fin). **Enregistrez** de la même manière que précédemment.

Testez et le tour est joué.



IV- Déployer avec Nginx

Dans cette section, nous vous expliquerons comment déployer votre site web en utilisant le serveur Nginx. Tout d'abord, nous vérifierons que Apache2 n'est pas actif ([voir Point Important](#)), et nous allons activer dans la même occasion Nginx :

```
systemctl stop apache2
```

```
systemctl start nginx
```

Avant de commencer, je vous invite à réaliser l'étape 8 de la **partie I**, qui est similaire pour le déploiement de votre site web sur Nginx.

1. **Placez** vous dans le répertoire Nginx avec :

```
cd /etc/nginx
```

Vérifiez que vous avez la même chose qu'avec ci-dessous (en tapant ls).

```
devops@ubuntu:/etc/nginx$ ls
conf.d          koi-win        nginx.conf     sites-enabled
fastcgi.conf    mime.types     proxy_params   snippets
fastcgi_params  modules-available  scgi_params    uwsgi_params
koi-utf          modules-enabled  sites-available  win-utf
```

2. **Dirigez-vous** vers le répertoire "sites-available".

3. Grâce à la commande 'ls', **vérifiez** que le fichier texte 'default' est bien existant.
4. **Ouvrez** le fichier pour le modifier en tant qu'administrateur.

```
sudo nano default
```

En descendant tout en bas, vous trouverez un exemple de modification du fichier.

```
# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
#    listen 80;
#    listen [::]:80;
#
#    server_name example.com;
#
#    root /var/www/example.com;
#    index index.html;
#
#    location / {
#        try_files $uri $uri/ =404;
#    }
#}
```

5. **Modifiez** le premier encadré ci-dessous pour que "SAE_SITE" représente le même nom que vous avez indiqué lorsque vous avez réalisé l'étape 8 de la **partie I**.

Pour le deuxième encadré, **saisissez** le nom que vous souhaitez pour votre site web.

```
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/SAE_SITE;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name www.unidoux.com;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}
```

6. **Enregistrez** précédemment vos modifications et **saisissez** la commande ci-dessous pour vérifier que votre syntaxe est bonne.

```
sudo nginx -t
```

Vous aurez ceci comme réponse.

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

7. Enfin, **redémarrez** le serveur Nginx pour appliquer les modifications que vous avez apportées au fichier de configuration.

```
sudo systemctl restart nginx
```

8. **Vérifiez** sur un navigateur web en tapant “localhost” que votre site est bien déployé !

Nous vous remercions d'avoir suivi ce manuel d'utilisateurs. Bonne continuation avec vos projets !

Alaoui Smaili Mèjdi

Senelet Amadis