

## SAE Réseaux Partie 1 – Biabiany Thomas ; Hadjras Mohcine



Partie 1.1 – installation d'Apache2 .....	p2
Partie 1.2 – Importation et installation du site internet sur le serveur Apache2 .....	p2
Partie 1.3 – Capture de trames sur le port 80 .....	p3
Partie 1.4 – Changement de port pour l'hébergement du site .....	p5
Partie 1.5 – Capture de trames sur le port 8080 .....	p6
Partie 1.6 – Principe DNS .....	p7
Partie 1.7 – Nginx .....	p7
Partie 1.8 – Commande Telnet .....	p9

# Partie 1 : Création d'un serveur web en local.

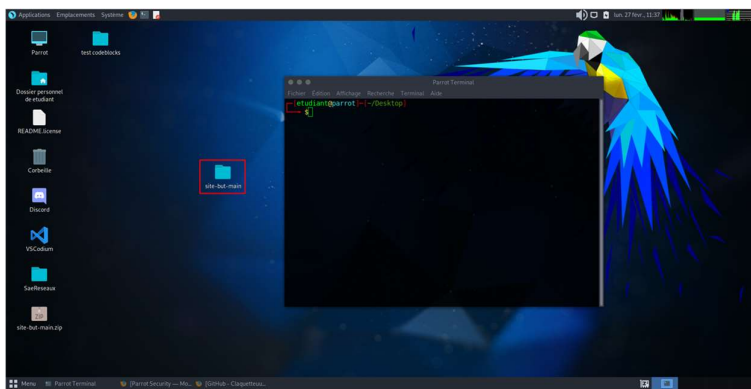
## Partie 1.1 – installation de Apache2

Installer Apache avec la commande ***sudo apt-get install apache2***:

```
[etudiant@parrot]~[/Desktop]
$ sudo apt-get install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.54-1-deb11u1).
apache2 passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
```

## Partie 1.2 – Importation et installation du site internet sur le serveur Apache2

Importer notre projet sur la VM.



À présent nous allons passer en superadmin afin de nous permettre l'accès aux commandes que nous allons faire à l'aide de la commande : ***sudo su***  
Il faut entrer notre mot de passe.

Déplacer le projet à la fin du chemin « */var/www/html* »

- il faut que notre index.html soit directement dans le dossier « html », pensez à retirer les autres fichiers html par défaut\* pour que seul votre site soit pris en compte.

\* :      ***rm -rf ./\****              Dans le dossier : */var/www/html*

```
[root@parrot]~/var/www/html
#rm -rf ./*
[root@parrot]~/var/www/html
#ls
[root@parrot]~/var/www/html
#
```

Une fois le dossier vide, nous déplaçons le contenu du projet dans le repertoire html à l'aide de la commande :

***cp /home/etudiant/Desktop/site-but\_main/\* -r ./***

La commande « *cp* » ci-dessus permet de copier notre projet dans le répertoire.  
D'abord nous spécifions le **chemin vers le contenu du projet**, l'option ***-r*** spécifie qu'il s'agit d'un répertoire et ***./*** pour indiquer que la destination du cp est celle où l'on se trouve déjà.

```

[root@parrot]~#
#cd /var/www/html codeblocks
[root@parrot]~/var/www/html#
#ls
[root@parrot]~/var/www/html#
#cp /home/etudiant/Desktop/site-but-main/* -r ./
[root@parrot]~/var/www/html#
#ls
assets index.html maquette pages src

```

Pensez à donner les droits au serveur web pour qu'il puisse accéder aux fichiers du projet :

chown www-data -R /var/www/html  
(faire attention à la casse)

```

[root@parrot]~/var/www/html#
#chown www-data -r /var/www/html
chown : option invalide -- 'r'
Saisissez « chown --help » pour plus d'informations.
[root@parrot]~/var/www/html#
#chown www-data -R /var/www/html
[root@parrot]~/var/www/html#
#

```

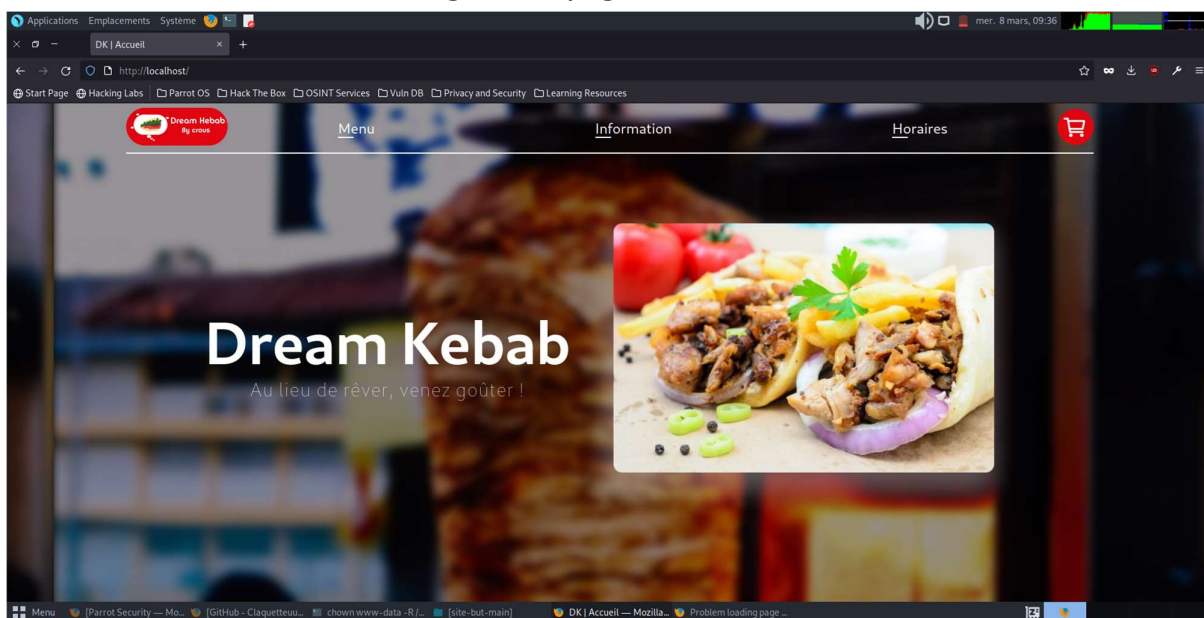
À présent il faut lancer le serveur apache2 :  
systemctl start apache2

```

[root@parrot]~/var/www/html#
#systemctl start apache2

```

Le site internet devrait être chargé sur la page localhost :



## Partie 1.3 – Capture de trames sur le port 80

Nous allons observer les échanges entre notre machine et le port 80 du serveur apache.

Pour faire cela, nous allons lancer WireShark et commencer à capturer les trames en appuyant sur l'aileron en haut à gauche.

Nous pouvons observer toutes les trames circulant sur notre ordinateur mais nous voulons nous concentrer sur tout ce qui concerne le port 80, nous allons donc écrire dans la barre de filtres :

tcp.port == 80 || udp.port == 80.

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide					
tcp.port == 80    udp.port == 80					
Liste de Paquet		UTF-8 / ASCII / UTF-16		Sensible à la casse	
				Chaine de Caractères	
				keep-alive	
No.	Time	Source	Destination	Protocol	Length Info
418	20.286497866	10.0.2.15	54.39.182.171	TCP	54 38666 → 443 [ACK] Seq=596 Ack=531283 Win=65535 Len=0
419	21.234394352	10.0.2.15	142.250.179.67	TCP	54 [TCP Keep-Alive] 53752 → 80 [ACK] Seq=425 Ack=703 Win=63882 Len=0
420	21.234784922	142.250.179.67	10.0.2.15	TCP	60 [TCP Keep-Alive ACK] 80 → 53752 [ACK] Seq=783 Ack=426 Win=65535 Len=0
421	23.621595612	10.0.2.15	142.250.179.74	TLSv1.2	167 Application Data
422	23.622154971	142.250.179.74	10.0.2.15	TCP	60 443 → 39544 [ACK] Seq=62440 Ack=2332 Win=65535 Len=0
423	23.624939512	10.0.2.15	54.39.182.171	TLSv1.2	138 Application Data
<ul style="list-style-type: none"> <li>Frame 420: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp0s3, id 0</li> <li>Ethernet II, Src: RealtekU12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_55:62:83 (08:00:27:55:62:83)</li> <li>Internet Protocol Version 4, Src: 142.250.179.67, Dst: 10.0.2.15</li> <li>Transmission Control Protocol, Src Port: 80, Dst Port: 53752, Seq: 703, Ack: 426, Len: 0 <ul style="list-style-type: none"> <li>Source Port: 80</li> <li>Destination Port: 53752</li> <li>[Stream index: 4]</li> <li>[TCP Segment Len: 0]</li> <li>Sequence Number: 703 (relative sequence number)</li> <li>Sequence Number (raw): 756224704</li> <li>[Next Sequence Number: 703 (relative sequence number)]</li> <li>Acknowledgment Number: 426 (relative ack number)</li> <li>Acknowledgment number (raw): 2376927545</li> <li>0101 ... = Header Length: 20 bytes (5)</li> <li>Flags: 0x010 (ACK)</li> <li>Window: 65535</li> <li>[Calculated window size: 65535]</li> <li>[Window size scaling factor: -2 (no window scaling used)]</li> <li>Checksum: 0xb8b5 [unverified]</li> <li>[Checksum Status: Unverified]</li> <li>Urgent Pointer: 0</li> </ul> </li> <li>[SEQ/ACK analysis] <ul style="list-style-type: none"> <li>[RTT: 0.014337660 seconds]</li> <li>[TCP Analysis Flags] <ul style="list-style-type: none"> <li>[Expert Info (Note/Sequence): ACK to a TCP keep-alive segment]</li> <li>[ACK to a TCP keep-alive segment]</li> <li>[Severity level: Note]</li> <li>[Group: Sequence]</li> </ul> </li> </ul> </li> <li>[Timestamps]</li> </ul>					

Nous pouvons repérer ici l'utilisation du mécanisme « keep-alive » qui sert à maintenir la connexion entre le site et le client tant que l'on considère qu'elle est nécessaire (il sert également à vérifier si la connexion est toujours effective).

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide					
tcp.port == 80    udp.port == 80					
Liste de Paquet		UTF-8 / ASCII / UTF-16		Sensible à la casse	
				Chaine de Caractères	
				get	
No.	Time	Source	Destination	Protocol	Length Info
417	27.688757435	54.39.182.171	10.0.2.15	TLSv1.2	9805 Application Data, Application Data, Application Data, Application Data
478	27.688783319	10.0.2.15	54.39.182.171	TCP	54 38666 → 443 [ACK] Seq=834 Ack=743797 Win=65535 Len=0
479	31.259720800	10.0.2.15	142.250.179.67	TCP	54 [TCP Keep-Alive] 53752 → 80 [ACK] Seq=425 Ack=703 Win=63882 Len=0
480	31.260808961	142.250.179.67	10.0.2.15	TCP	60 [TCP Keep-Alive ACK] 80 → 53752 [ACK] Seq=703 Ack=426 Win=65535 Len=0
481	41.290912613	10.0.2.15	142.250.179.67	TCP	54 [TCP Keep-Alive] 53752 → 80 [ACK] Seq=425 Ack=703 Win=63882 Len=0
482	41.291344349	142.250.179.67	10.0.2.15	TCP	60 [TCP Keep-Alive ACK] 80 → 53752 [ACK] Seq=703 Ack=426 Win=65535 Len=0
485	47.054334946	10.0.2.15	217.70.184.38	TCP	74 36310 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3691445764 TSecr=0 WS=128
486	47.081301229	217.70.184.38	10.0.2.15	TCP	60 80 → 36316 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
487	47.081341339	10.0.2.15	217.70.184.38	TCP	54 36316 → 80 [ACK] Seq=0 Ack=1 Win=64240 Len=0
488	47.081307395	10.0.2.15	217.70.184.38	HTTP	527 GET /enseignement/intro-web/_static/formproc/debug?kebab=on&ignon=on&caviar=on&truffe=on&sauce=on&boisson=on HTTP/1.1\r\n
<ul style="list-style-type: none"> <li>Window: 64240</li> <li>[Calculated window size: 64240]</li> <li>[Window size scaling factor: -2 (no window scaling used)]</li> <li>Checksum: 0xb9f6 [unverified]</li> <li>[Checksum Status: Unverified]</li> <li>Urgent Pointer: 0</li> <li>[SEQ/ACK analysis] <ul style="list-style-type: none"> <li>[RTT: 0.027066392 seconds]</li> <li>[Bytes in flight: 473]</li> <li>[Bytes sent since last PSH flag: 473]</li> </ul> </li> <li>[Timestamps]</li> <li>TCP payload (473 bytes)</li> <li>Myperext Transfer Protocol <ul style="list-style-type: none"> <li>GET /enseignement/intro-web/_static/formproc/debug?kebab=on&amp;ignon=on&amp;caviar=on&amp;truffe=on&amp;sauce=on&amp;boisson=on HTTP/1.1\r\n</li> <li>[Expert Info (Chat/Sequence): GET /enseignement/intro-web/_static/formproc/debug?kebab=on&amp;ignon=on&amp;caviar=on&amp;truffe=on&amp;sauce=on&amp;boisson=on HTTP/1.1\r\n]</li> <li>[GET /enseignement/intro-web/_static/formproc/debug?kebab=on&amp;ignon=on&amp;caviar=on&amp;truffe=on&amp;sauce=on&amp;boisson=on HTTP/1.1\r\n]</li> <li>[Severity level: Chat]</li> <li>[Group: Sequence]</li> <li>Request Method: GET</li> <li>Request URI: /enseignement/intro-web/_static/formproc/debug?kebab=on&amp;ignon=on&amp;caviar=on&amp;truffe=on&amp;sauce=on&amp;boisson=on</li> <li>Request Version: HTTP/1.1</li> <li>Host: champion.net\r\n</li> <li>User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0\r\n</li> <li>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n</li> <li>Accept-Language: en-US,en;q=0.5\r\n</li> <li>Accept-Encoding: gzip, deflate\r\n</li> <li>Referer: http://localhost/r/n</li> <li>DNT: 1\r\n</li> <li>Connection: keep-alive\r\n</li> <li>Upgrade-Insecure-Requests: 1\r\n</li> <li>\r\n</li> <li>[Full request URI: http://champion.net/enseignement/intro-web/_static/formproc/debug?kebab=on&amp;ignon=on&amp;caviar=on&amp;truffe=on&amp;sauce=on&amp;boisson=on]</li> <li>[HTTP request 1/1]</li> <li>[Response in frame: 490]</li> </ul> </li></ul>					

Ici, nous pouvons repérer que la méthode GET est utilisée sur le site hébergé au port 80, nous pouvons aussi voir à quelle adresse est envoyée la requête GET dans le lien tout en bas de la capture d'écran.

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000000	10.0.2.15	142.250.179.74	TLSv1.2	167 Application Data
2	0.000028276	142.250.179.74	10.0.2.15	TCP	60 443 → 39544 [ACK] Seq=1 Ack=114 Win=65535 Len=0
3	0.022154288	142.250.179.74	10.0.2.15	TLSv1.2	1696 Application Data, Application Data, Application Data, Application Data
4	0.022217930	10.0.2.15	142.250.179.74	TCP	54 39544 → 443 [ACK] Seq=114 Ack=1553 Win=65535 Len=0
5	0.024622318	10.0.2.15	142.250.179.74	TLSv1.2	93 Application Data
6	0.025062741	142.250.179.74	10.0.2.15	TCP	60 443 → 39544 [ACK] Seq=1553 Ack=153 Win=65535 Len=0
7	0.089704656	10.0.2.15	54.39.182.171	TLSv1.2	138 Application Data
8	0.090136754	54.39.182.171	10.0.2.15	TCP	60 443 → 38666 [ACK] Seq=1 Ack=85 Win=65535 Len=0
9	0.215173629	54.39.182.171	10.0.2.15	TLSv1.2	13759 Application Data, Application Data, Application Data
10	0.215201868	10.0.2.15	54.39.182.171	TCP	54 38666 → 443 [ACK] Seq=85 Ack=13706 Win=65535 Len=0

Dans cette section, nous pouvons voir quels sont les expéditeurs et destinataires concernés.

## Partie 1.4 – Changement de port pour l'hébergement du site

Nous sommes sur le port **80** et nous voulons héberger le site sur le port **8080**, nous allons donc voir comment faire cela.

Premièrement il faut se rendre sur le fichier qui contient les informations du serveur web apache.

```
cd /etc/apache2/
```

```
[root@parrot]-[~]
#cd /etc/apache2
[root@parrot]-[/etc/apache2]
#ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled
```

Et modifier le fichier ports.conf avec la commande « nano »

```
[root@parrot]-[/etc/apache2]
#nano ports.conf
```

Le port par défaut est 80, il faut simplement aller à la ligne « Listen » et changer 80 par 8080.

```
GNU nano 5.4 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 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Ne pas oublier de redémarrer le serveur apache afin de mettre à jour les informations.

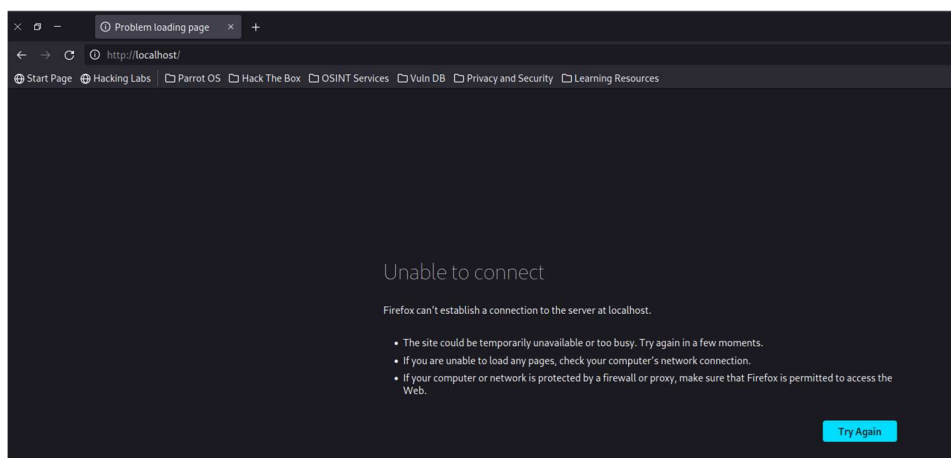
```
[root@parrot]-[/etc/apache2]
#systemctl restart apache2
[root@parrot]-[/etc/apache2]
#
```

Le localhost ne devrait plus fonctionner et si il fonctionne toujours alors il faut désactiver nginx avec les commandes suivantes :

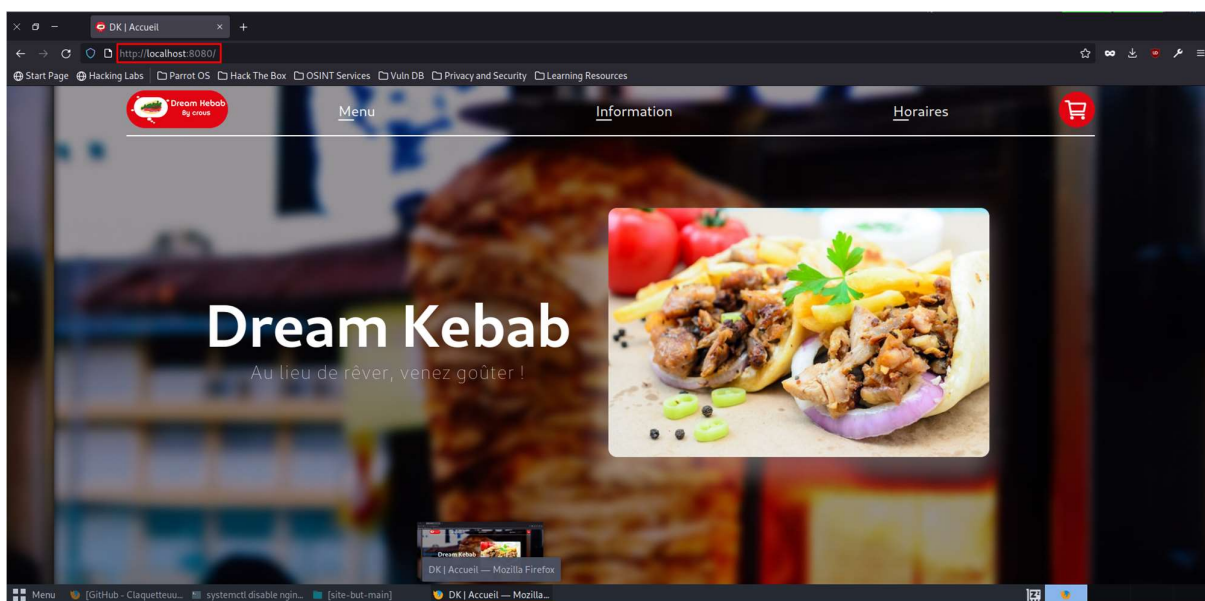
**systemctl stop nginx**  
**systemctl disable nginx**

```
[root@parrot]-[/etc/apache2]
#systemctl stop nginx
[root@parrot]-[/etc/apache2]
#systemctl disable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable nginx
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
[root@parrot]-[/etc/apache2]
#
```





Si nous allons sur le port 8080 en écrivant localhost:8080 dans la barre de lien, nous devrions retrouver notre site internet :



## **Partie 1.5 – Capture de trames sur le port 8080**

La capture de trames sur le port 8080 donne le même résultat que la capture de trames sur le port 80. Seulement, le port 8080 est utilisé par simplicité par rapport au port 80 (deux fois 80 : rien ne nous empêche de choisir un autre port libre). Il est utilisé pour :

- Héberger un deuxième site internet sur une même machine quand le port 80 est déjà utilisé.
- Faire de la maintenance sur le même site internet tout en ayant l'original toujours en ligne.

La seule différence est que le port 80 est le port par défaut de Apache, il sera donc utilisé en « localhost » tandis que pour aller sur le port 8080 il faut le renseigner dans la barre de lien.

## Partie 1.6 – Principe DNS

Maintenant que notre serveur apache2 tourne correctement, nous allons mettre en place un nom de domaine pour accéder plus facilement à notre application.

Pour ce faire, effectuez la commande :

***nano /etc/hosts***

Créez une nouvelle ligne puis complétez la de cette manière :

**<mon\_ip> <le\_nom\_de\_domaine>**

Vous pouvez ensuite enregistrer les modifications et constater que votre nom domaine fonctionne correctement en navigant sur celui-ci depuis votre navigateur.

## Partie 1.7 – Nginx

Nous avons vu comment héberger notre site grâce à apache2, maintenant nous allons voir comment l'héberger grâce à nginx, une alternative d'apache2.

### Partie 1.7.1 – Désactiver apache2

Pour utiliser correctement nginx, on va d'abord devoir désactiver apache2 pour éviter tout conflit.

Pour ce faire, on va effectuer les commandes suivantes :

- ***Systemctl stop apache2***
- ***Systemctl disable apache2***

Ensuite, nous allons activer et lancer nginx :

- ***Systemctl enable nginx***
- ***Systemctl start nginx***

Voilà, apache2 est maintenant désactivé, nous pouvons passer à la suite.

## Partie 1.7.2 – Configurer Nginx

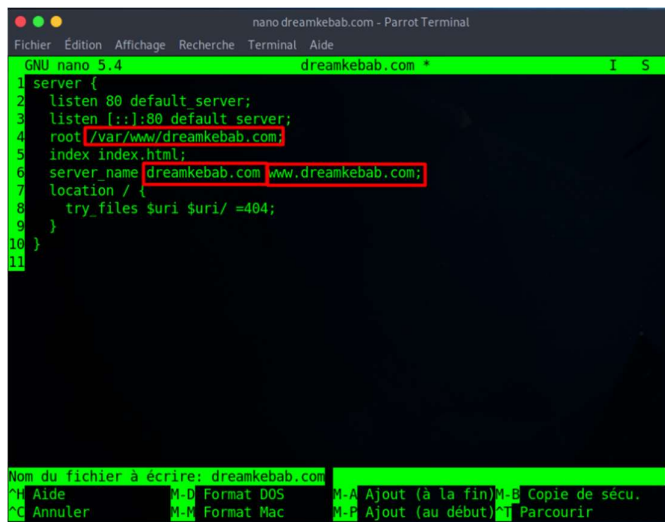
Assurez-vous d'avoir votre dossier de projet dans `/var/www`.

Avec nginx deux dossiers sont très importants :

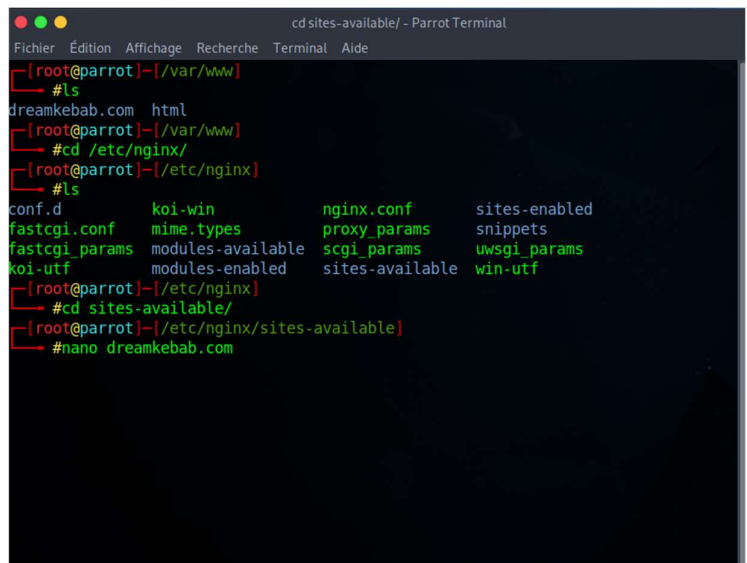
- Sites-available : répertorie les fichiers de configuration de nos sites pouvant être exécutés.
- Site-enabled, répertorie les liens vers les fichiers de configuration des sites qui tournent actuellement.

On va donc commencer par créer le fichier de configuration de notre site dans `nginx/sites-available`.

Dans ce fichier, nous allons écrire ceci :



```
GNU nano 5.4 dreamkebab.com *
1 server {
2   listen 80 default_server;
3   listen [::]:80 default_server;
4   root /var/www/dreamkebab.com;
5   index index.html;
6   server_name dreamkebab.com www.dreamkebab.com;
7   location / {
8     try_files $uri $uri/ =404;
9   }
10 }
11
```



```
cd sites-available/ - Parrot Terminal
Fichier Édition Affichage Recherche Terminal Aide
[root@parrot]~/var/www
#ls
dreamkebab.com html
[root@parrot]~/var/www
#cd /etc/nginx/
[root@parrot]~/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
[root@parrot]~/etc/nginx
#cd sites-available/
[root@parrot]~/etc/nginx/sites-available
#nano dreamkebab.com
```

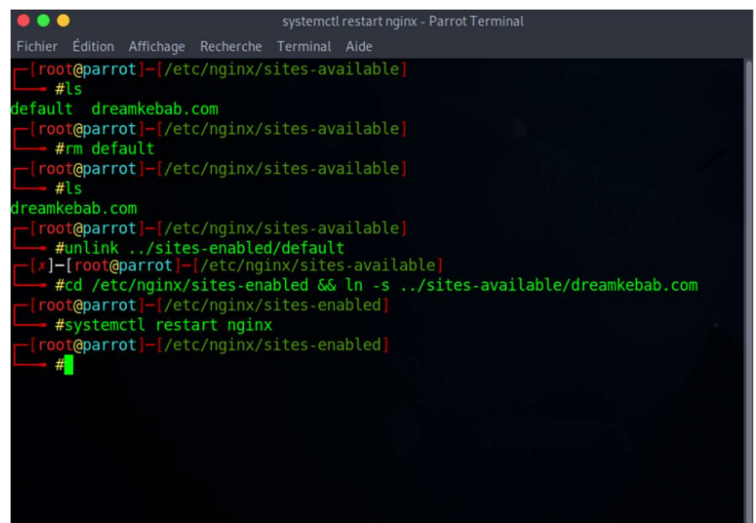
```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/dreamkebab.com;
    index index.html;
    server_name dreamkebab.com www.dream-
kebab.com;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Remarque : Changez `/var/www/dreamkebab.com` par le chemin vers le répertoire où se trouve votre site.

Vous pouvez constater l'existence d'un fichier « default » dans le dossier « `nginx/sites-available` », supprimez le et déliez le dans le dossier « `nginx/sites-enabled` ».

Comme dit précédemment dans le dossier « `nginx/sites-enabled` », nous allons placer les liens vers nos fichiers de configuration.

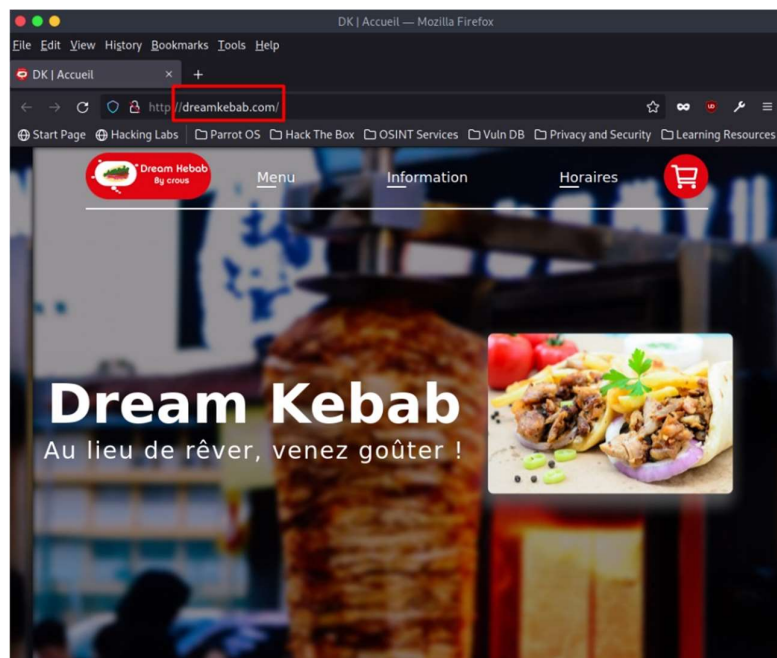
Pour cela, effectuez la commande :  
**`cd /etc/nginx/sites-enabled && ln -s ../sites-available/dreamkebab.com`**



```
systemctl restart nginx - Parrot Terminal
Fichier Édition Affichage Recherche Terminal Aide
[root@parrot]~/etc/nginx/sites-available
#ls
default dreamkebab.com
[root@parrot]~/etc/nginx/sites-available
#rm default
[root@parrot]~/etc/nginx/sites-available
#ls
dreamkebab.com
[root@parrot]~/etc/nginx/sites-available
#unlink ../sites-enabled/default
[x]-[root@parrot]~/etc/nginx/sites-available
#cd /etc/nginx/sites-enabled && ln -s ../sites-available/dreamkebab.com
[root@parrot]~/etc/nginx/sites-enabled
#systemctl restart nginx
[root@parrot]~/etc/nginx/sites-enabled
#
```



Maintenant, on peut redémarrer nginx et constater que le serveur fonctionne correctement.



## Partie 1.8 – Commande Telnet

Telnet est un protocole qui a pour but de communiquer avec un système distant avec un CLI.

### Partie 1.8.1 – Requête GET

Pour effectuer une requête GET avec telnet, on va commencer par effectuer la commande **telnet <domaine\_cible> <port>**

Une fois la commande exécuté, écrivez votre requête :

**GET <route> HTTP/1.1**  
**Host: <domaine\_cible>**

Appuyez 2 fois sur la touche entrer et constatez que votre requête à été effectuée et que le serveur vous a renvoyé les données correspondantes.

