



Royaume du Maroc
Ministère de l'Éducation Nationale de la Formation Professionnelle
de l'Enseignement Supérieur et de la Recherche Scientifique
Université Sultan Moulay Slimane
Faculté Polydisciplinaire – BENI MELLAL
Master Système de Télécommunication et Réseaux Informatiques



Exposé sous le thème :

HTTPs

Réalisé par :

EL KHABLI OMAR
AYOK THON AYOK AWOL
IBOURK Othman

Encadré par:

Pr: Anouar DARIF

PLAN

I	Présentation de serveur HTTPs
II	Fonctions du serveur HTTPs
III	Configuration de l'environnement global
IV	Configuration du serveur principal
V	Démarrage et arrêt du serveur
VI	Modules et serveur HTTPs
VII	Directives de configuration dans le fichier de configuration
VIII	Configuration d'hôtes virtuels
IX	Conclusion

Présentation du serveur HTTPs

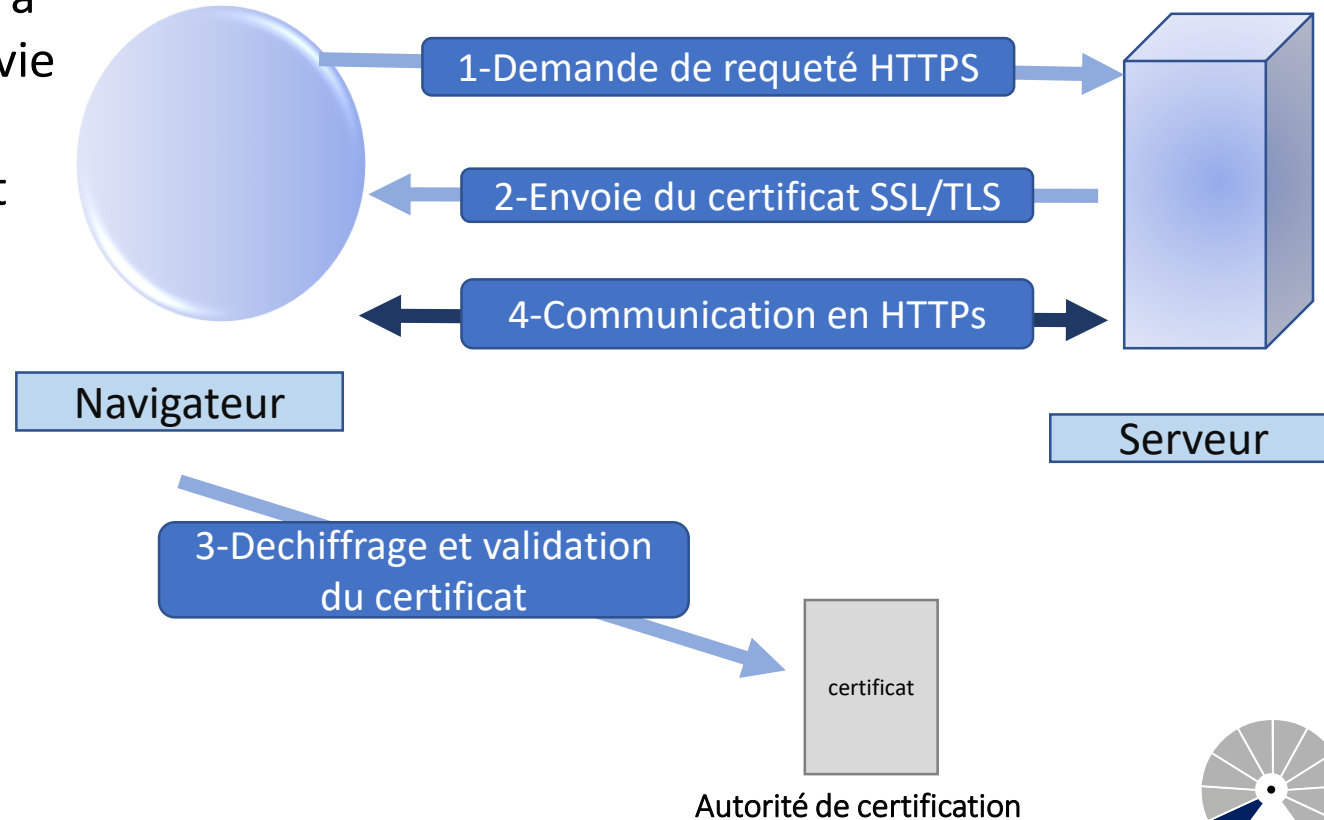
Définition de HTTPs

- le **protocole HTTPS** (*HyperText Transfer Protocol Secure*) est un protocole de transfert hypertexte sécurisé, utilisé pour envoyer des documents sur le web. Les données transférées sont des fichiers HTML, des fichiers d'image, des vidéos, etc.
- Les transmissions client/serveur sont en fait cryptées par un protocole de sécurisation des données de type **SSL** (*Secure Sockets Layer*) ou **TLS** (*Transport Layer Security*), qui se base sur l'installation de **certificats**.
- On peut dire que **HTTPs** est une connexion **HTTP** dans un tunnel chiffre SSL/TLS.

Présentation du serveur HTTPs

Principe (simplifié) du protocole HTTPs

- ❑ Le client et le serveur doivent se reconnaître grâce à un certificat d'authentification qui a une durée de vie limitée.
- ❑ Le client envoie une requête HTTP, mais celle-ci est d'abord chiffrée avant d'être envoyée au serveur.
- ❑ Le serveur déchiffre le paquet d'informations chiffrées. La réponse du serveur est alors chiffrée, puis envoyée au client.
- ❑ Le client reçoit la réponse chiffrée du serveur. Il la déchiffre.





Présentation du serveur HTTPs

À quoi sert le protocole HTTPS ?

Le **protocole HTTPS** sert à :

- Chiffrer les données ;
- Vérifier l'identité du serveur sur lequel le navigateur se connecte;
- Empêcher la modification des données par la perte de paquets;
- Améliorer la sécurité de la navigation sur le Web;



Présentation du serveur HTTPs

La différence entre HTTPs et HTTP

	HTTP	HTTPS
URL	http://	https://
Sécurité	Non sécurisé	Sécurité renforcée
Numéro de port	80	443
Couche OSI	Couche Application	Couche Application
Certificats SSL/TLS	Non	Oui
Validation de domaine	Non requis	Validation de domaine
Chiffrement	Non	Oui

Les étapes du fonctionnement du cryptage SSL/TLS:

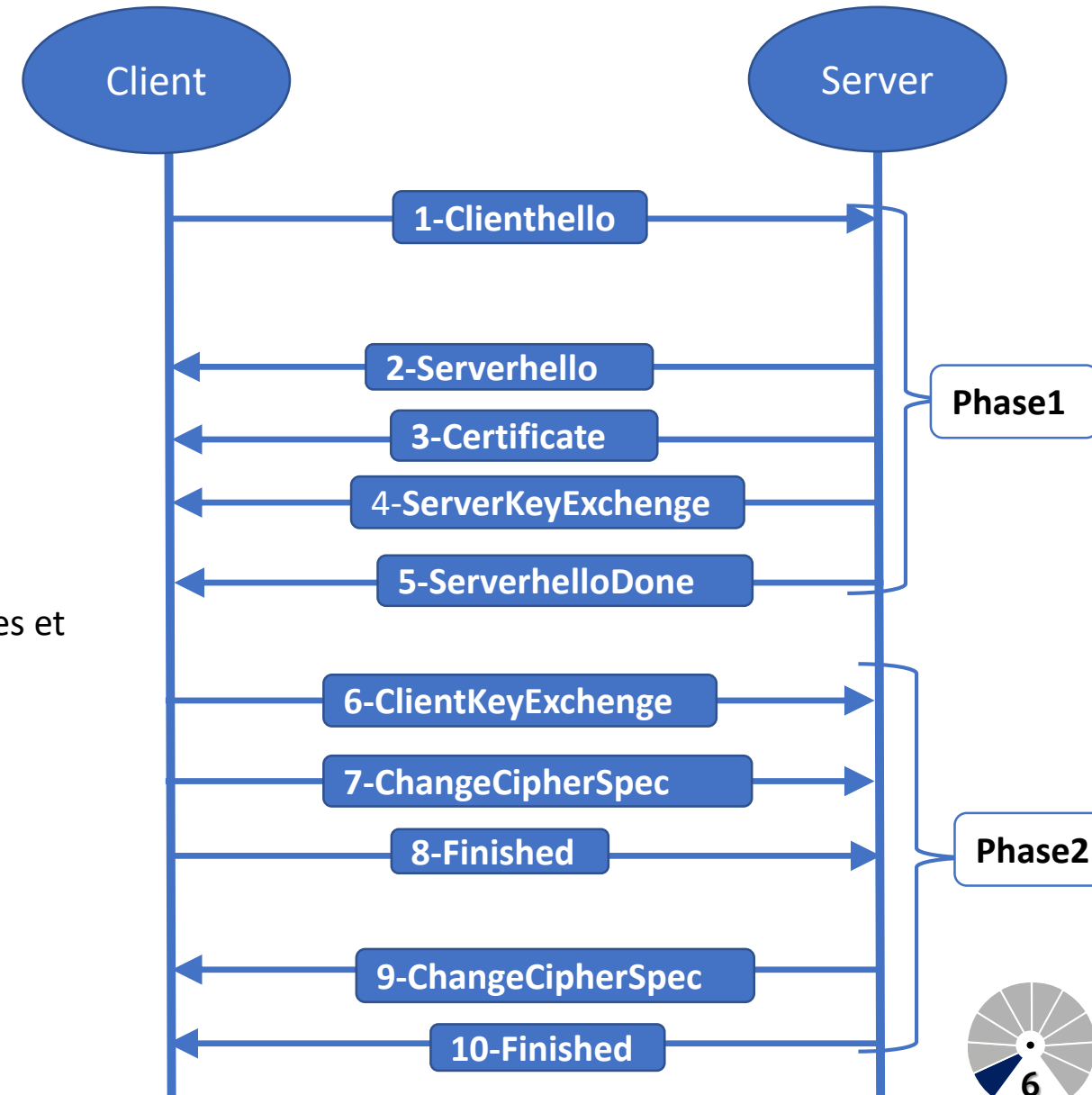
➤ SSL se déroule en deux phases:

1. Phase 1: authentification du serveur

- Requête client,
- Le serveur envoie son certificat ,
- Le client vérifie le certificat du serveur à l'aide de la clé publique ,
- Le client génère un pré-master secret (PMS)qui sera utilisé pour générer le master-key,
- PMS est chiffré avec la clé publique du serveur,
- Les données échangées entre le client et le serveur seront chiffrées et authentifiées avec des clés dérivées du master-secret.

2. Phase 2: authentification du client

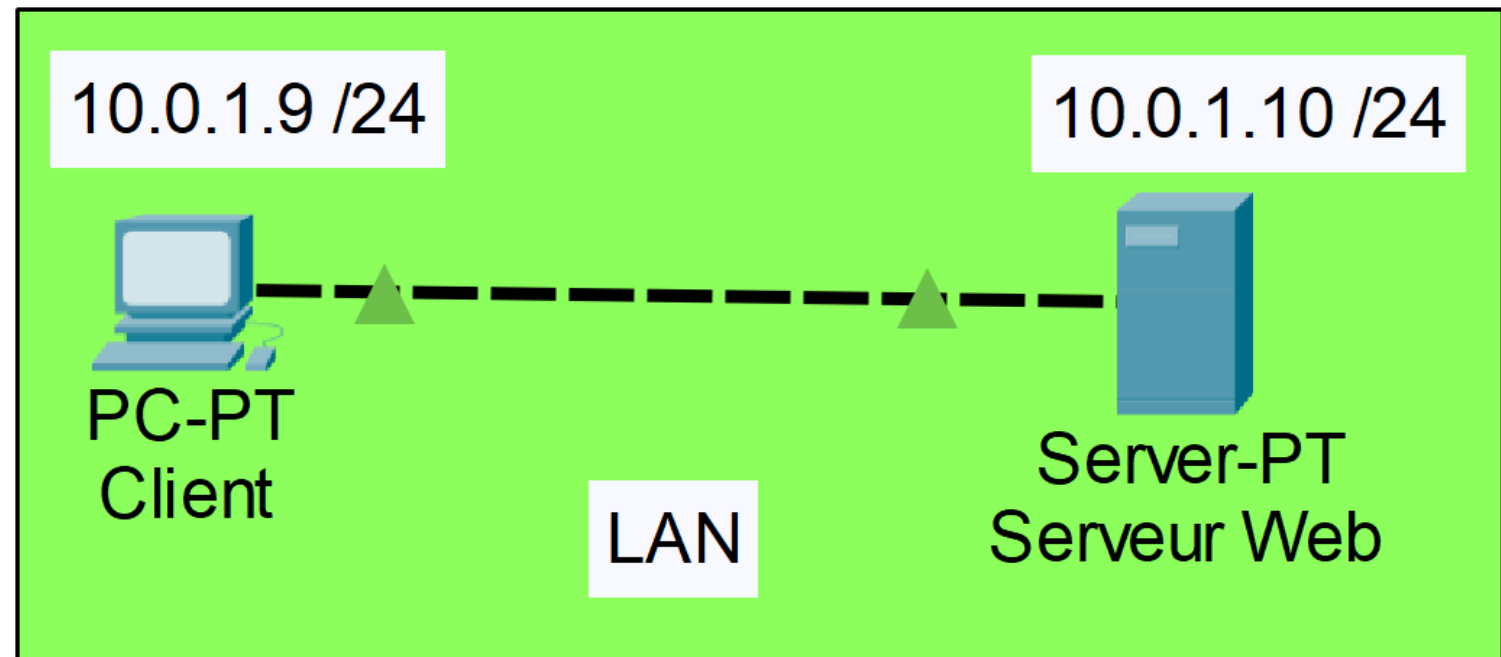
- Le serveur peut demander au client de s'authentifier en lui demandant son certificat,
- Le client répond en envoyant son certificat puis en signant un message avec sa clé.



Configuration de l'environnement global

La première chose est de configurer un hyperviseur où nous allons installer nos machines virtuelles, et les images iso de ces deux machines virtuelles.

- ❑ **virtualBox** (hyperviseur)
- ❑ **Kali linux** (serveur)
- ❑ **Ubuntu** (client)



Présentation et
fonctions de HTTPs



Configuration de
l'environnement et
du serveur principal



Modules, Démarrage
et arrêt de serveur
HTTPs



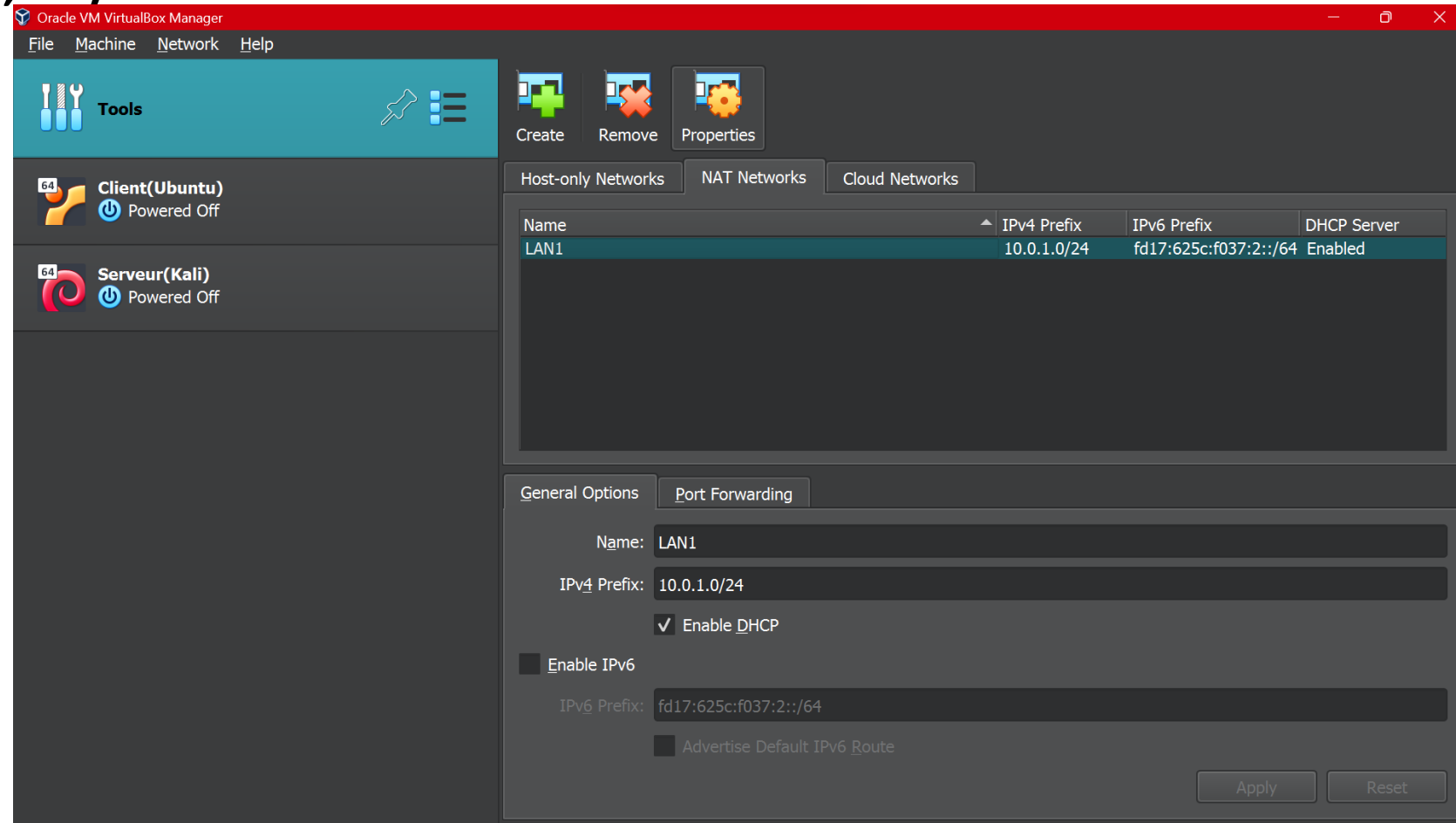
Directives de
configuration et hôtes
virtuels



Conclusion

Configuration de l'environnement global

- créations de LAN dans virtualbox
- mod de connexion (NAT, Bridge, ...)



Présentation et
fonctions de HTTPs

Configuration de
l'environnement et
du serveur principal

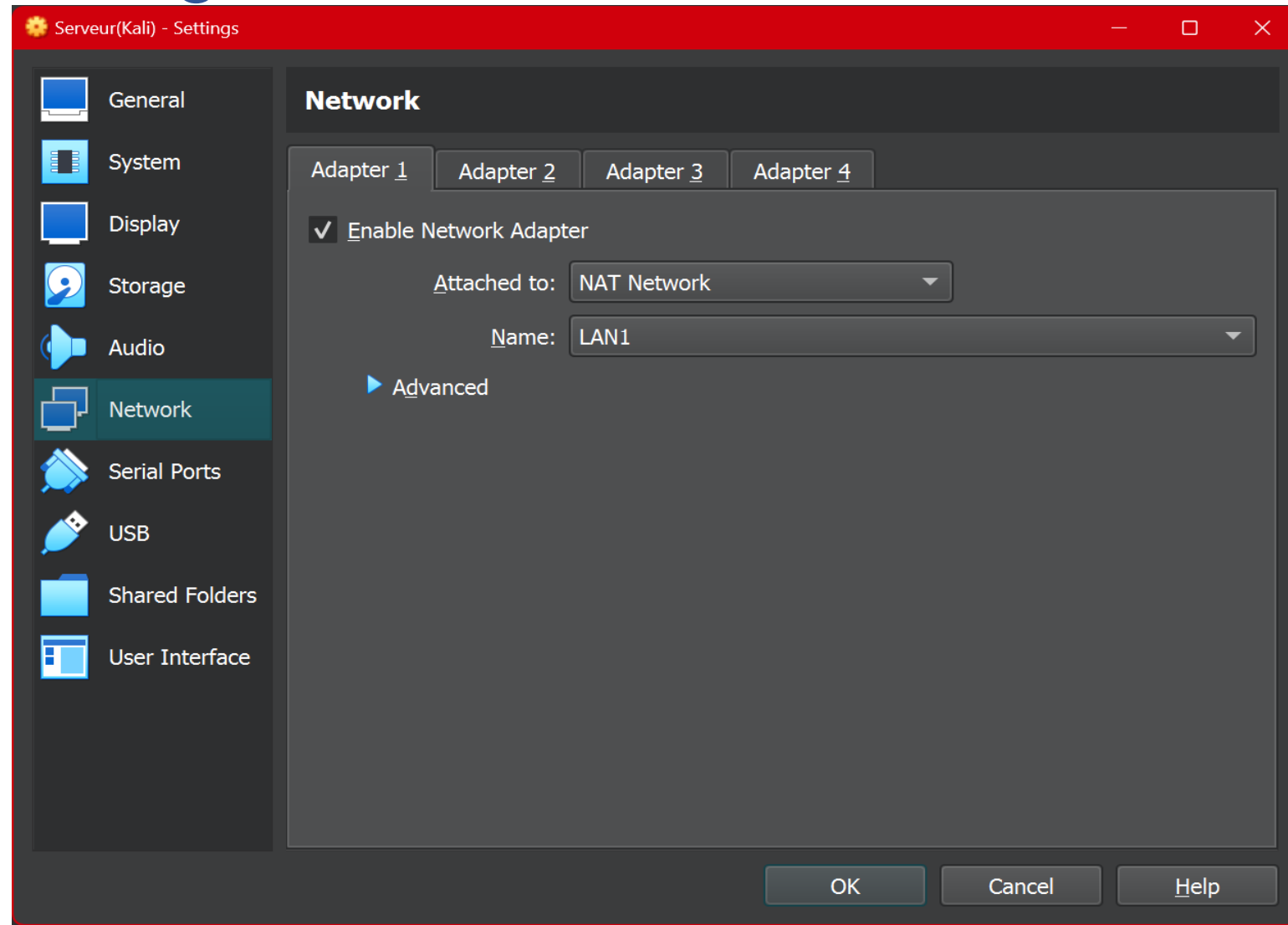
Modules, Démarrage
et arrêt de serveur
HTTPs

Directives de
configuration et hôtes
virtuels

Conclusion

Configuration de l'environnement global

- créations de LAN dans virtualbox
- mod de connexion (NAT, Bridge, ...)
- **choix de réseau dans la machine**



Configuration du serveur principal

- ❑ Le serveur web que nous utilisons est Apache2, qui est gratuit et open source, il peut être facilement installé sur le site officiel ou via la commande suivante :

```
sudo apt install apache2
```

- ❑ Apache2 est préinstallé sur certaines distribution linux.
- ❑ Après avoir installé et démarré le serveur, nous pouvons accéder au serveur en tapant l'adresse IP dans le navigateur , il va afficher une page par défaut.
- ❑ Pour mettre notre fichier HTML on doit le placer dans le chemin **/var/www/**
- ❑ NB: apache2 est configuré pour la connexion HTTP par défaut.

Démarrage et arrêt du serveur

- ❑ Pour démarrer le serveur ,on utilise la commande suivante:

```
sudo systemctl start apache2
```

- ❑ Pour voir l'état du serveur s'il est actif ou non, il affiche également des informations sur la configuration du serveur, qui sont utiles en cas d'erreur.

```
sudo systemctl status apache2
```

- ❑ Pour arrêter le serveur on exécute la commande suivante:

```
sudo systemctl stop apache2
```

- ❑ Pour configurer le serveur pour qu'il démarre automatiquement au démarrage de la machine.

```
sudo systemctl enable apache2
```

Modules et serveur HTTPs

- ❑ Les modules sont des programmes de service qui peuvent être liés et chargés dynamiquement pour étendre la nature du serveur HTTPs.
- ❑ Tous les modules configurés par défaut pour HTTP seront la même pour HTTPs, de plus nous aurons besoin du SSL pour le cryptage.
- ❑ Pour activer le module SSL. `sudo a2enmod ssl`
- ❑ **Remarque:** avant d'activer ssl, il prendra en compte certaines dépendances comme **setenvif**, **mime** et **socache_shmcb**.

Présentation et
fonctions de HTTPs

Configuration de
l'environnement et
du serveur principal

Modules, Démarrage
et arrêt de serveur
HTTPs

Directives de
configuration et hôtes
virtuels

Conclusion

Modules et serveur HTTPs

File Actions Edit View Help

(kali@kali)-[/etc/apache2]

\$ ls

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

(kali@kali)-[/etc/apache2]

\$ ls mods-available

access_compat.load	authz_groupfile.load	dav_lock.load	include.load	mpm_prefork.load	proxy_http.load	socache_dbm.load
actions.conf	authz_host.load	dbd.load	info.conf	mpm_worker.conf	proxy.load	socache_memcache.load
actions.load	authz_owner.load	deflate.conf	info.load	mpm_worker.load	proxy_scgi.load	socache_redis.load
alias.conf	authz_user.load	deflate.load	lbmethod_bybusyness.load	negotiation.conf	proxy_uwsgi.load	socache_shmcb.load
alias.load	autoindex.conf	dialup.load	lbmethod_byrequests.load	negotiation.load	proxy_wstunnel.load	speling.load
allowmethods.load	autoindex.load	dir.conf	lbmethod_bytraffic.load			ssl.conf
asis.load	brotli.load	dir.load	lbmethod_heartbeat.load			ssl.load
auth_basic.load	buffer.load	dump_io.load	ldap.conf	proxy_ajp.load	remoteip.load	status.conf
auth_digest.load	cache_disk.conf	echo.load	ldap.load	proxy_balancer.conf	reqtimeout.conf	status.load
auth_form.load	cache_disk.load	env.load	log_debug.load	proxy_balancer.load	reqtimeout.load	substitute.load
authn_anon.load	cache.load	expires.load	log_forensic.load	proxy.conf	request.load	suexec.load
authn_core.load	cache_socache.load	ext_filter.load	lua.load	proxy_connect.load	rewrite.load	unique_id.load
authn_dbd.load	cern_meta.load	file_cache.load	macro.load	proxy_express.load	sed.load	userdir.conf
authn_dbm.load	cgid.conf	filter.load	md.load	proxy_fcgi.load	session_cookie.load	userdir.load
authn_file.load	cgid.load	headers.load	mime.conf	proxy_fdpass.load	session_crypto.load	usertrack.load
authn_socache.load	cgi.load	heartbeat.load	mime.load	proxy_ftp.conf	session_dbd.load	vhost_alias.load
authnz_fcgi.load	charset_lite.load	heartmonitor.load	mime_magic.conf	proxy_ftp.load	session.load	xml2enc.load
authnz_ldap.load	data.load	http2.conf	mime_magic.load	proxy_hcheck.load	setenvif.conf	
authz_core.load	dav_fs.conf	http2.load	mpm_event.conf	proxy_html.conf	setenvif.load	
authz_dbd.load	dav_fs.load	ident.load	mpm_event.load	proxy_html.load	slotmem_plain.load	
authz_dbm.load	dav.load	imagemap.load	mpm_prefork.conf	proxy_http2.load	slotmem_shm.load	

(kali@kali)-[/etc/apache2]

\$

Directives de configuration dans le fichier de configuration

```
kali@kali: /etc/apache2
File Actions Edit View Help

(kali@kali)-[~]
$ cd /etc/apache2

(kali@kali)-[/etc/apache2]
$ ls
apache2.conf  conf-enabled  magic  mods-enabled  sites-available
conf-available  envvars  mods-available  ports.conf  sites-enabled

(kali@kali)-[/etc/apache2]
$
```

- ❑ Dans le dossier sites-available il y a deux fichiers: **000-default.conf** pour l'accès de http, et le fichier **default-ssl.conf** pour l'accès de https.
- ❑ mais seulement le fichier **000-default.conf** est actif dans sites-enabled.

Directives de configuration dans le fichier de configuration

Création de certificat:

// clé du client.

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -  
pkeyopt rsa_keygen_pubexp:65537 -out cakey.pem
```

// signature .

```
openssl req -new -x509 -key cakey.pem -out cacert.pem -days 1095
```

```
Country Name (2 letter code) [AU]:Ma  
State or Province Name (full name) [Some-State]:Beni mellal  
Locality Name (eg, city) []:beni mellal  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:fpbm  
Organizational Unit Name (eg, section) []:.  
Common Name (e.g. server FQDN or YOUR name) []:www.masterstri.ma  
Email Address []:admin@masterstri.ma
```




Directives de configuration dans le fichier de configuration

Création de certificat:

création des répertoires nécessaires au certificat.

```
mkdir demoCA
```

```
mkdir demoCA/certs
```

```
mkdir demoCA/crl
```

```
mkdir demoCA/newcerts
```

```
mkdir demoCA/private
```

```
touch demoCA/index.txt
```

```
echo 02 > demoCA/serial
```

Directives de configuration dans le fichier de configuration

```
GNU nano 7.2 /usr/lib/ssl/openssl.cnf
#####
[ CA_default ]

dir               = ./demoCA               # Where everything is kept
certs             = $dir/certs             # Where the issued certs are kept
crl_dir           = $dir/crl               # Where the issued crl are kept
database          = $dir/index.txt         # database index file.
#unique_subject   = no                    # Set to 'no' to allow creation of
                                           # several certs with same subject.
new_certs_dir     = $dir/newcerts          # default place for new certs.

certificate       = $dir/cacert.pem        # The CA certificate
serial            = $dir/serial            # The current serial number
crlnumber         = $dir/crlnumber         # the current crl number
                                           # must be commented out to leave a V1 CRL
crl               = $dir/crl.pem           # The current CRL
private_key       = $dir/private/cakey.pem # The private key
#the quieter you become, the more you are able to hear"
x509_extensions   = usr_cert              # The extensions to add to the cert
```



Directives de configuration dans le fichier de configuration

Création de certificat:

// modification de correspondance de politique

```
mv cacert.pem demoCA/
```

```
mv cakey.pem demoCA/private/
```

```
sudo nano /usr/lib/ssl/openssl.cnf
```

```
# For the CA policy
[ policy_match ]
countryName          = match
stateOrProvinceName  = optional
organizationName     = optional
organizationalUnitName = optional
commonName           = supplied
emailAddress          = optional
```



Directives de configuration dans le fichier de configuration

Création de certificat:

// clé du serveur.

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -pkeyopt  
rsa_keygen_pubexp:65537 -out privkey-www.masterstri.ma.pem
```

// signature.

```
openssl req -new -key privkey-www.masterstri.ma.pem -out certreq-  
www.masterstri.ma.csr
```

NB: les informations que nous avons renseignées lors de la signature de la clé du client doivent être répétées ici pour la signature de la clé du serveur.

Directives de configuration dans le fichier de configuration

Création de certificat:

//signature de l'autorité de certification

```
openssl ca -in certreq-www.masterstri.ma.csr -out cert-  
www.masterstri.ma.pem
```

```
(kali㉿kali)-[~]  
$ openssl ca -in certreq-www.masterstri.ma.csr -out cert-www.masterstri.ma.pem  
Using configuration from /usr/lib/ssl/openssl.cnf  
Check that the request matches the signature  
Signature ok  
Certificate Details:  
  Serial Number: 2 (0x2)  
  Validity  
    Not Before: Apr 27 22:49:03 2023 GMT  
    Not After : Apr 26 22:49:03 2024 GMT  
  Subject:  
    countryName           = Ma  
    stateOrProvinceName   = Beni mellal  
    organizationName      = fpbm  
    commonName            = www.masterstri.ma  
    emailAddress          = admin@masterstri.ma  
  X509v3 extensions:  
    X509v3 Basic Constraints:  
      CA:FALSE  
    X509v3 Subject Key Identifier:  
      A0:FC:47:20:FD:B6:26:EB:42:F3:38:04:9D:52:F3:3E:E7:74:BF:CB  
    X509v3 Authority Key Identifier:  
      B0:DF:07:5B:33:D4:67:CB:78:6C:30:0F:40:63:6F:35:25:FC:2B:2B  
Certificate is to be certified until Apr 26 22:49:03 2024 GMT (365 days)  
Sign the certificate? [y/n]:y
```

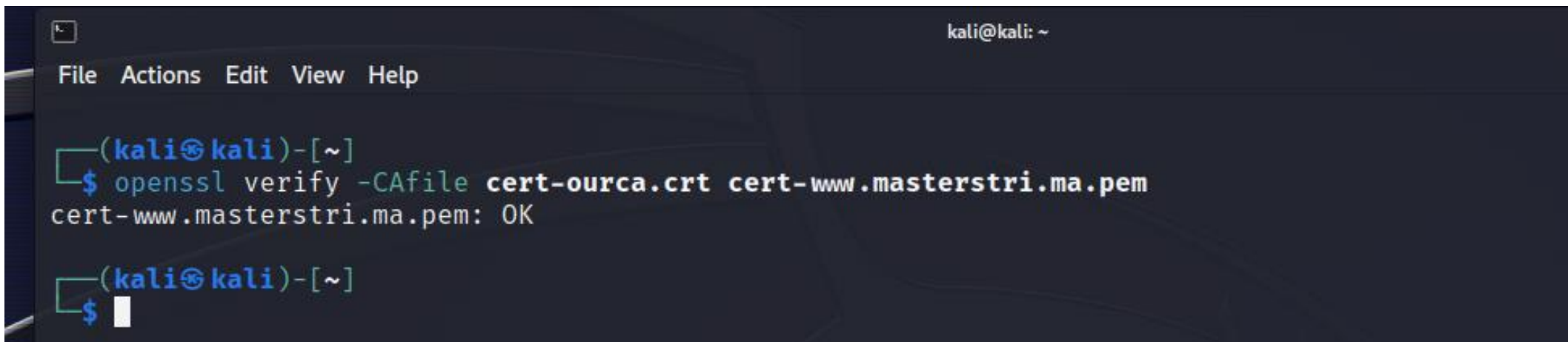

Directives de configuration dans le fichier de configuration

Création de certificat:

pour assurer la validation du certificat, nous pouvons faire la vérification

```
cp demoCA/cacert.pem cert-ourca.crt
```

```
openssl verify -CAfile cert-ourca.crt cert-www.masterstri.ma.pem
```



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ openssl verify -CAfile cert-ourca.crt cert-www.masterstri.ma.pem  
cert-www.masterstri.ma.pem: OK  
(kali@kali)-[~]  
$
```

Directives de configuration dans le fichier de configuration

Création de certificat:

- nous devons maintenant copier ces certificats là où apache (notre serveur Web) peut les utiliser

```
sudo cp cert-www.masterstri.ma.pem /etc/ssl/certs
```

```
sudo cp cert-ourca.crt /etc/ssl/certs
```

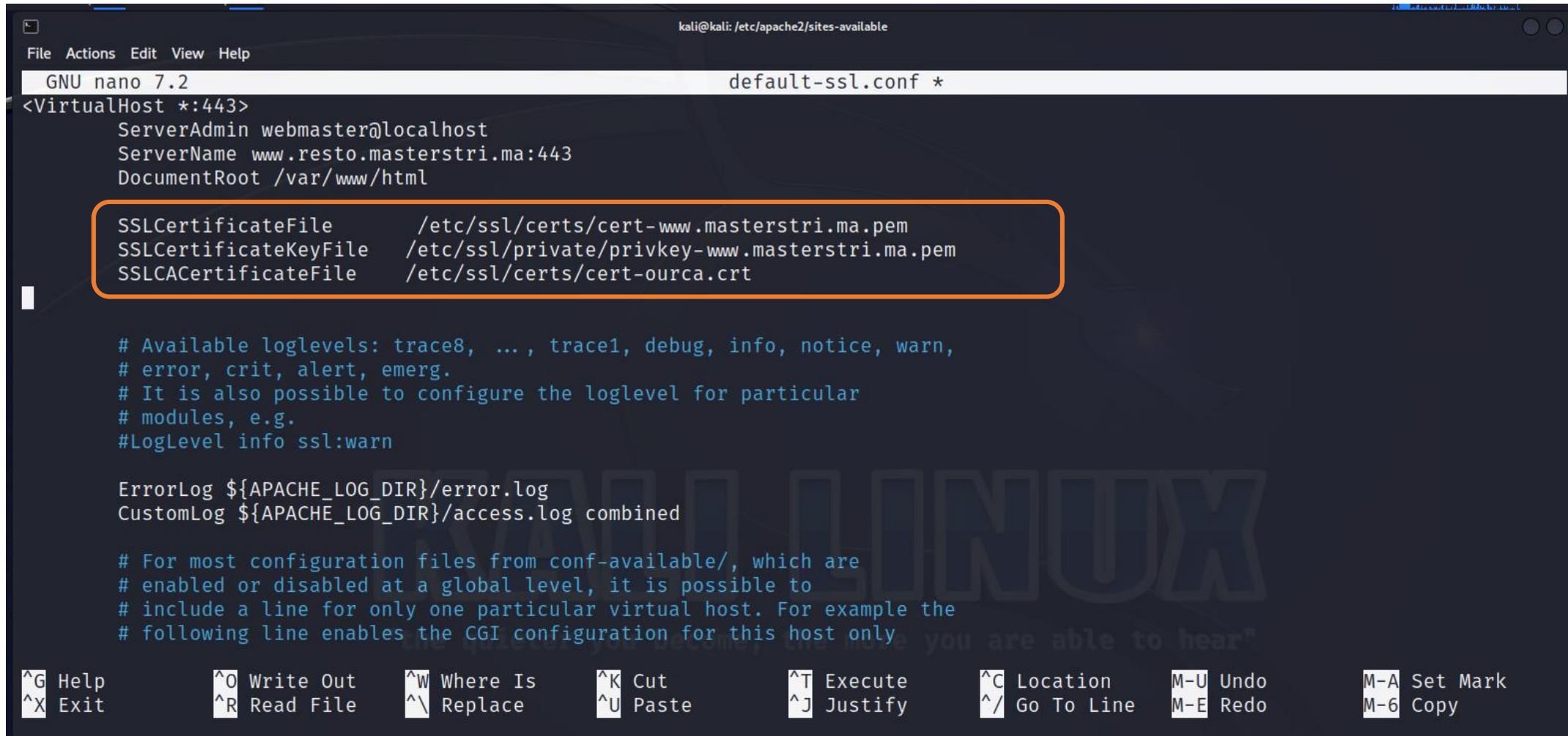
```
sudo cp privkey-www.masterstri.ma.pem /etc/ssl/private/
```

- revenons maintenant à la partie où nous devons activer le fichier **default-ssl.conf**

```
(kali㉿kali)-[/etc/apache2/sites-available]
$ ls
000-default.conf  default-ssl.conf

(kali㉿kali)-[/etc/apache2/sites-available]
$ sudo nano default-ssl.conf
```

Directives de configuration dans le fichier de configuration



```
kali@kali: /etc/apache2/sites-available
GNU nano 7.2 default-ssl.conf *
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    ServerName www.resto.masterstri.ma:443
    DocumentRoot /var/www/html

    SSLCertificateFile      /etc/ssl/certs/cert-www.masterstri.ma.pem
    SSLCertificateKeyFile    /etc/ssl/private/privkey-www.masterstri.ma.pem
    SSLCACertificateFile    /etc/ssl/certs/cert-ourca.crt

    # 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
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    #
```

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo Set Mark Copy

Directives de configuration dans le fichier de configuration

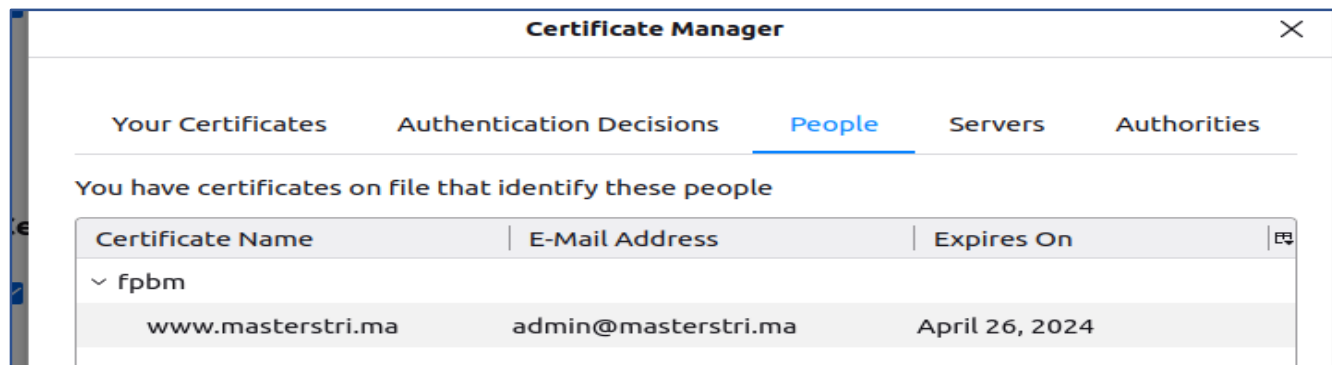
// activation de site default-ssl pour la connexion HTTPs

```
sudo a2ensite default-ssl
```

// pour charger les modifications dans le serveur

```
sudo systemctl reload apache2
```

- l'étape suivant nous devons informer le navigateur du client que notre certificat peut être approuvé en installant le certificat cert-ourca.crt.
- pour fair ça on cherche dans le navigateur **paramètres** -> **Confidentialité et sécurité** -> **Certificats** -> **importer** et on choisit le fichier **cert-ourca.crt**



Configurtion d'hôtes virtuels

- ❑ Un hôte virtuel est une directive de configuration Apache qui nous permet d'exécuter plusieurs sites Web sur un seul serveur.
- ❑ il y a deux type d'hôtes virtuels:
 - basé sur le nom
 - basé sur l' IP
- ❑ nous allons créer trois sites:
 - `www.othman.ma`
 - `www.omar.ma`
 - `www.ayok.ma`

dans le dossier **sites-available** nous allons créer trois fichiers des noms:

othman-ssl.conf

omar-ssl.conf

ayok-ssl.conf

Configuration d'hôtes virtuels

command : **sudo nano othman-ssl.conf**

```
File Actions Edit View Help
GNU nano 7.2 othman-ssl.conf *
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    ServerName www.othman.ma:443
    DocumentRoot /var/www/othman

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

    SSLEngine on

    SSLCertificateFile      /etc/ssl/certs/cert-www.othman.ma.pem
    SSLCertificateKeyFile    /etc/ssl/private/privkey-www.othman.ma.pem
    SSLCACertificateFile    /etc/ssl/certs/cert-othman.crt

    <FilesMatch "\.(?:cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
</VirtualHost>
```

Après, dans chaque fichier (omar-ssl.conf et ayok-ssl.conf), on fait ces configurations

Configuration d'hôtes virtuels

Activations des hôtes virtuels:

```
sudo a2ensite othman-ssl.conf
```

```
sudo a2ensite omar-ssl.conf
```

```
sudo a2ensite ayok-ssl.conf
```



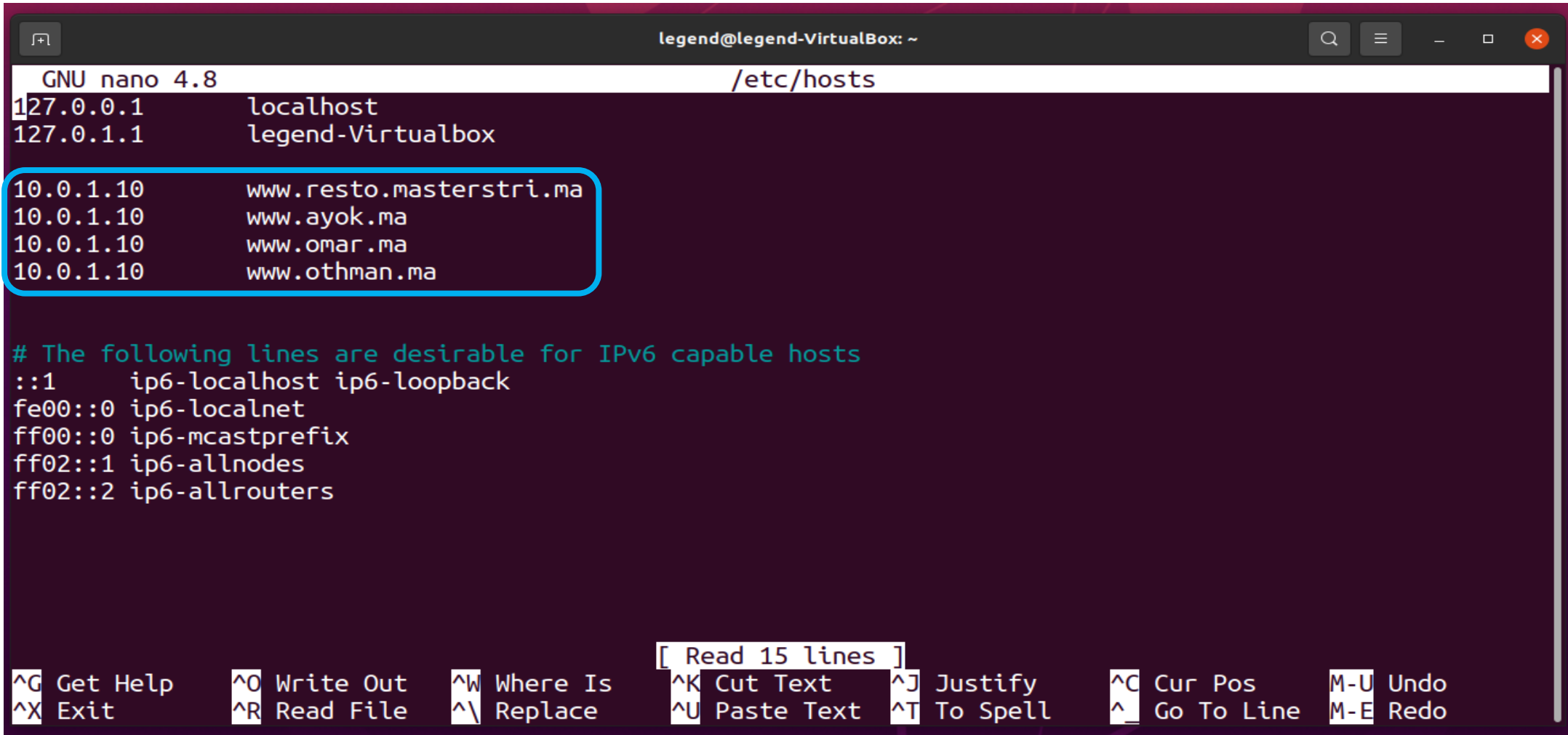
```
kali@kali: /etc/apache2
File Actions Edit View Help

(kali@kali)-[/etc/apache2]
$ ls sites-enabled
ayok-ssl.conf  default-ssl.conf  omar-ssl.conf  othman-ssl.conf

(kali@kali)-[/etc/apache2]
$
```

Le fichier hosts est un moyen de mapper les noms d'hôtes aux adresses IP.

Dans un sens, le fichier hosts agit comme un serveur DNS local. **sudo nano /etc/hosts**



```
GNU nano 4.8 /etc/hosts
1 127.0.0.1    localhost
2 127.0.1.1    legend-Virtualbox
3
4 10.0.1.10    www.resto.masterstri.ma
5 10.0.1.10    www.ayok.ma
6 10.0.1.10    www.omar.ma
7 10.0.1.10    www.othman.ma
8
9 # The following lines are desirable for IPv6 capable hosts
10 ::1         ip6-localhost ip6-loopback
11 fe00::0     ip6-localnet
12 ff00::0     ip6-mcastprefix
13 ff02::1     ip6-allnodes
14 ff02::2     ip6-allrouters
```

[Read 15 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo

**Présentation et
fonctions de HTTPs**



**Configuration de
l'environnement et du
serveur principal**



**Modules, Démarrage
et arrêt de serveur
HTTPs**



**Directives de
configuration et hôtes
virtuels**



Conclusion

Demo



Conclusion

- ❑ HTTPS est plus sécurisé que HTTP grâce à sa notion des certificats, dans ce mini projet nous avons utilisé des certificats valable uniquement au sein d'une LAN.
- ❑ D'avoir un certificat pour les réseaux publique il faut l'acheter chez l'autorité des certificat (**Comodo, GeoTrust, Symantec etc...**).
- ❑ Cependant, il est possible d'obtenir un certificat gratuitement chez **Let's Encrypt**.

Merci pour votre attention !