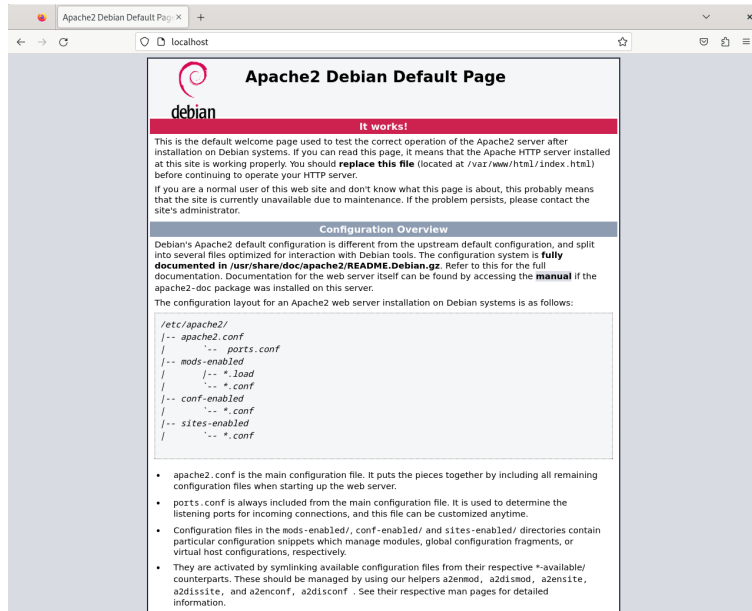


## Job 2

Pour installer apache j'ai utilisé la commande "sudo apt-get install apache2". Une fois installé, je suis allé sur mon navigateur où j'ai tapé le nom de domaine <https://localhost>. Je suis arrivé sur la page suivante :



Pour accéder à mon serveur depuis mon hôte j'ai rentré l'adresse ip de mon serveur dans mon navigateur web.

## Job 3

Les différents serveurs web :

### - Apache HTTP server

En open source et très utilisé, il est donc très bien documenté et compatible avec tous les systèmes d'exploitation. Il est aussi configurable en profondeur et on peut rajouter de nombreux modules. Il peut par contre être assez gourmand en ressources et pas forcément l'option la plus rapide. Il est aussi assez complexe pour les débutants.

### - Nginx

Nginx est le deuxième plus utilisé du monde et est connu pour avoir de bonnes performances et peu intensif au niveau des ressources. Il est aussi populaire par sa capacité à croître et à augmenter le trafic et est très rapide pour servir rapidement les fichiers statiques. Il est par contre plus complexe pour configurer les contenus dynamiques et les applications.

### - LiteSpeed

Ce serveur web est aussi axé sur la vitesse et l'efficacité, il supporte les fichiers de configurations apache ce qui rend la migration entre ces deux serveurs faciles. Son principal inconvénient est son prix, car la version gratuite a des limitations.

- Caddy

Caddy est très facile à prendre en main, configure automatiquement HTTPS en utilisant "let's encrypt". Il n'a par contre pas autant de fonctionnalités que ses concurrents qui pourra être gênant si l'on veut mettre en place des serveurs complexes.

- Microsoft Internet Information Service (IIS)

Directement intégré avec Windows Server et bien adapté pour les applications .NET. Il est par contre premièrement pensé pour Windows, pas aussi versatile que d'autres plateformes car pas recommandé pour les autres systèmes d'exploitation.

- Cherokee

Très recommandé pour les débutants, avec une interface web pour la configuration. Il n'est par contre pas autant utilisé que d'autres plateformes ce qui rends le dépannage grâce à la communauté plus difficile.

- OpenLiteSpeed

La version open source de LiteSpeed, il est similaire à celui ci mais ne possède pas autant de fonctionnalités que son aîné.

- Lighttpd

Performant et peu gourmand, il est très performant pour les contenus statiques mais pas autant pour les applications web complexes.

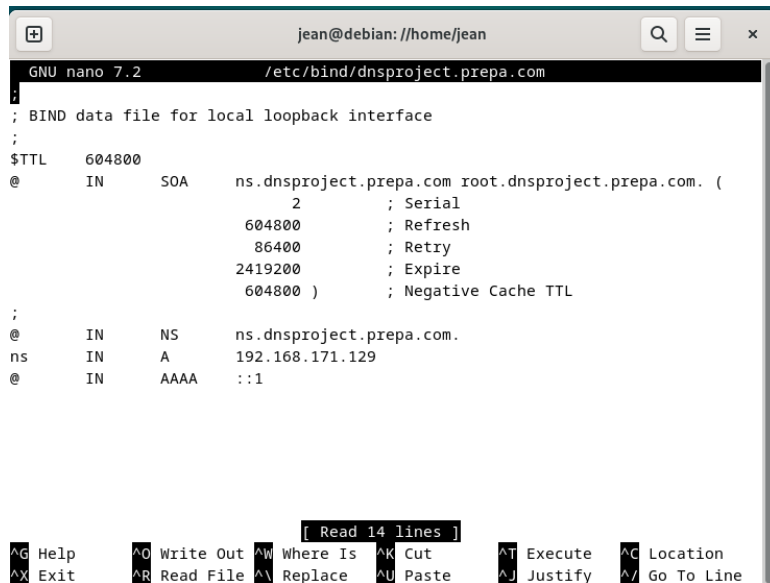
## **Job 4**

Pour mettre en place un DNS j'ai installé Bind.

Je suis ensuite allé dans "named.conf.options" pour indiquer l'ip de mon serveur ainsi que le DNS de google en forwarder. Cela permet de passer sur le DNS de Google au cas où on ne trouve pas le nom de domaine sur le premier DNS. Cela n'est pas forcément pertinent pour l'exercice mais j'imagine que ca l'est généralement.

Je suis aussi allé dans les fichiers de configurations, notamment "named.conf.local" pour déclarer une zone nommée "dnsproject.prepa.com" avec un fichier appelé "dnsproject.prepa.com" situé dans "/etc/bind/". J'ai ensuite créé le fichier, j'ai copié collé le

fichier db.local en l'appelant "dnsproject.prepa.com". Je l'ai ensuite modifié comme ceci :



```
GNU nano 7.2 /etc/bind/dnsproject.prepa.com
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.dnsproject.prepa.com root.dnsproject.prepa.com. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.dnsproject.prepa.com.
ns        IN      A        192.168.171.129
@         IN      AAAA     ::1

[ Read 14 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

On peut voir que j'ai modifié le SOA (Start of Authority) ainsi qu'un enregistrement de type A qui permet de faire correspondre un nom DNS avec une adresse ip. J'ai ensuite redémarré bind avec la commande "sudo service bind9 restart".



```
[sudo] password for jean:
jean@debian:~/home/jean$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (192.168.171.129) 56(84) bytes of data:
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=5 ttl=64 time=0.041 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=6 ttl=64 time=0.055 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=7 ttl=64 time=0.046 ms
64 bytes from dnsproject.prepa.com (192.168.171.129): icmp_seq=8 ttl=64 time=0.054 ms
^C
--- dnsproject.prepa.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7146ms
rtt min/avg/max/mdev = 0.031/0.048/0.055/0.008 ms
jean@debian:~/home/jean$
```

Comme montré ci-dessus je peux ping mon serveur.

## Job 5

Pour obtenir un nom de domaine public il faut passer par un hébergeur de domaine ou à un bureau d'enregistrement de noms de domaines. On choisira ensuite un nom de domaine non utilisé et le payer.

Les extensions de domaines ont été créées dans une volonté de classification, il en existe énormément. Un autre terme serait "Top Level Domain" souvent abrégé par TLD.

Quelques exemples :

### .com

La plus courante. Près de 50 % des sites web l'utilisent.  
Elle désigne à la base un site commercial.

### .net

Comme le .com, le .net est l'une des premières extensions de domaine. Elle a été créée pour être utilisée par les organisations impliquées dans les technologies de réseau.

### .org

Le .org a été créé pour être utilisé par les organisations à but non lucratif et reste l'extension de domaine de choix pour ces organisations.

L'extension .org est ouverte à tous.

### .gov

L'extension .gov ne peut être utilisée que par les organisations gouvernementales américaines.

### .io

Réservé à des entreprises technologiques. Il fait allusion à I/O (Input/Output).

Les entreprises choisissent le .io car il indique que leur site web est lié à la technologie. Il est également populaire parce que les domaines .io sont souvent disponibles lorsque la version .com est prise.

### .int

.int signifie international, et c'est l'une des extensions de domaine dont les exigences sont les plus strictes. Elle ne peut être utilisée que par des organisations et des programmes approuvés par un traité international. Par exemple, le site web de l'Agence spatiale européenne est esa.int.

Cependant, certains anciens domaines .int qui ne répondent pas aux exigences, comme ymca.int, ont été conservés.

### .mobi

Le TLD .mobi indique qu'un site est optimisé pour les appareils mobiles. Un site utilisant .mobi doit se conformer aux exigences fixées par le registre.

De nos jours, presque tous les sites sont adaptés aux appareils mobiles, de sorte que la désignation d'un site .mobi distinct a moins de valeur qu'auparavant.

## **Job 6**

Pour connecter mon hôte au nom de domaine local j'ai renseigné l'adresse ip de mon serveur dans le fichier /etc/hosts sur debian.

```
jean@debian: //home/jean
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 debian
192.168.171.129 dnsproject.prepa.com
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

[ Read 7 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^P Paste     ^J Justify   ^_ Go To Line
```

Je l'ai aussi fais dans windows pour pouvoir y accéder de là bas dans  
`/C:/Windows/System32/drivers/etc/hosts`.

```
hosts
Fichier  Modifier  Affichage

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97      rhino.acme.com      # source server
#      38.25.63.10      x.acme.com        # x client host

# localhost name resolution is handled within DNS itself.
#
#      127.0.0.1        localhost
#      ::1              localhost
#      192.168.171.129  dnsproject.prepa.com
```

## Job 7

J'ai commencé par installer ufw puis j'ai modifié le fichier de config "before.rules" dans  
`/etc/ufw/`.

A la ligne "`#ok icmp codes for INPUT`" j'ai remplacé tous les ACCEPT par des DROP. Cela  
permettra de bloquer les pings entrants. J'ai aussi ouvert le port 80 avec la commande "`sudo  
ufw allow 80/TCP`" utilisé pour le trafic web.



```
jean@debian: /etc/ufw
GNU nano 7.2 before.rules
# quickly process packets for which we already have a connection
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j DROP
-A ufw-before-input -p icmp --icmp-type time-exceeded -j DROP
-A ufw-before-input -p icmp --icmp-type parameter-problem -j DROP
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP

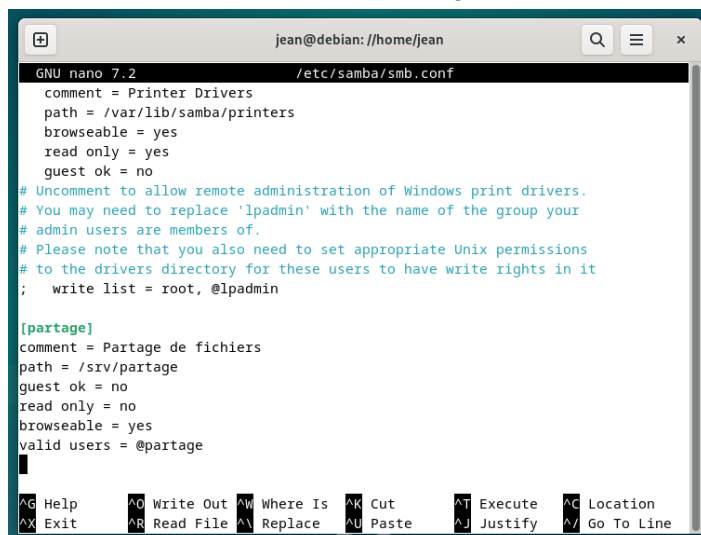
# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^V Replace   ^P Paste     ^J Justify   ^_ Go To Line
```

## Job 8

J'ai commencé par installer samba avec la commande "apt-get install -y samba".

J'ai ensuite édité le fichier de configuration "/etc/samba/smb.conf".



```
jean@debian: //home/jean
GNU nano 7.2 /etc/samba/smb.conf
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[partage]
comment = Partage de fichiers
path = /srv/partage
guest ok = no
read only = no
browseable = yes
valid users = @partage
```

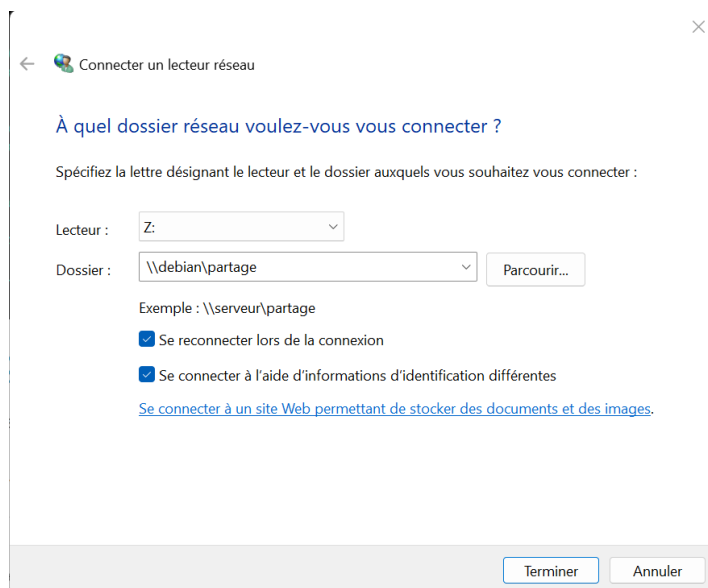
Dans ce fichier j'ai ajouté ces lignes pour déclarer mon partage. La ligne "path" est importante car elle définit le dossier à partager. La ligne valid users définit aussi le groupe qui aura droit d'accéder à ces fichiers.

J'ai ensuite redémarré samba avec la commande "systemctl restart smbd".

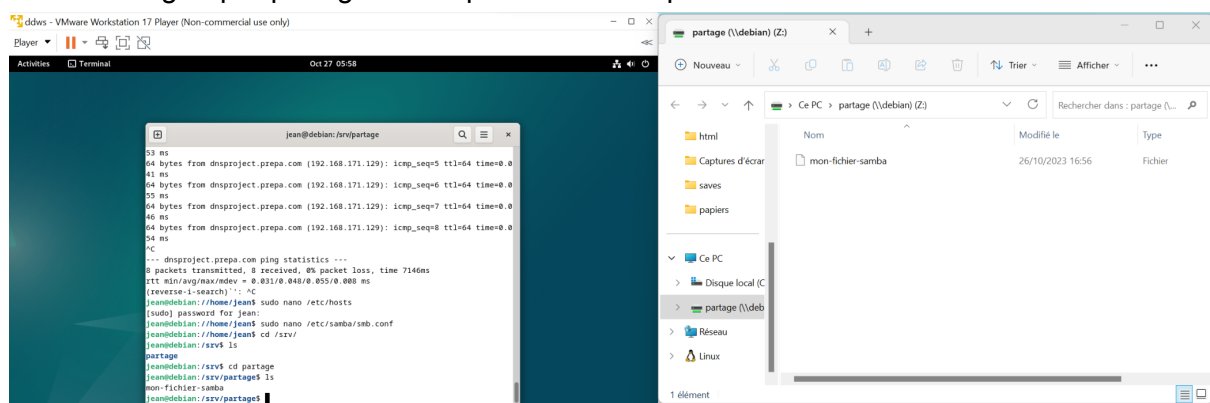
J'ai aussi ajouté un mot de passe samba à mon utilisateur avec la commande "smbpasswd -a jean". Une fois mon mot de passe samba établi, j'ai créé le groupe "partage" que j'avais cité dans mon fichier de configuration avec la commande "groupadd" et j'ai rajouté mon utilisateur au groupe avec la commande "gpasswd -a".

J'ai fini par créer mon dossier "partage" avec la commande "mkdir", attribué le dossier au groupe "partage" avec "chgrp" (le groupe partage sera donc propriétaire du dossier partage) et ajouté les droits de lecture et d'écriture au dossier avec "chmod".

Mon samba est maintenant configuré, on va maintenant essayer de se connecter à partir de mon hôte sur windows, dans le gestionnaire de fichier je vais faire un clic droit sur "mon pc" et cliquer sur "connecter un lecteur réseau".



Dans cette fenêtre on va renseigner le nom de la machine du serveur, dans mon cas “debian” suivi du dossier partagé, ici “partage”. On va ensuite renseigner les informations de connexions que l’on avait créées auparavant, c'est-à-dire le nom de l'utilisateur que l’on a mis dans le groupe “partage” ainsi que son mot de passe samba.



A noter que pour accéder au dossier depuis windows j'ai dû ouvrir les ports 139 et 445 sur mon firewall debian.

## Pour aller plus loin

Pour créer un certificat j'ai utilisé openssl. J'ai commencé par créer une clé privée en utilisant la commande “openssl genrsa -des3 -out dnsproject.prepa.com 2048”. J'ai ensuite entré un mot de passe quand demandé.

J'ai par la suite créé un CSR (certificate signing request) à partir de ma clé privée. J'ai utilisé pour ça la commande “openssl req -key dnsproject.prepa.com.key -new -out dnsproject.prepa.com.csr”. Cela va nous demander un mot de passe, où l'on rentrera celui que l'on a créé à l'étape d'avant, ainsi que diverses informations sur notre domaine comme le pays, la région, une adresse email etc. Le plus important est “Common Name” où il faudra rentrer le nom de notre domaine.

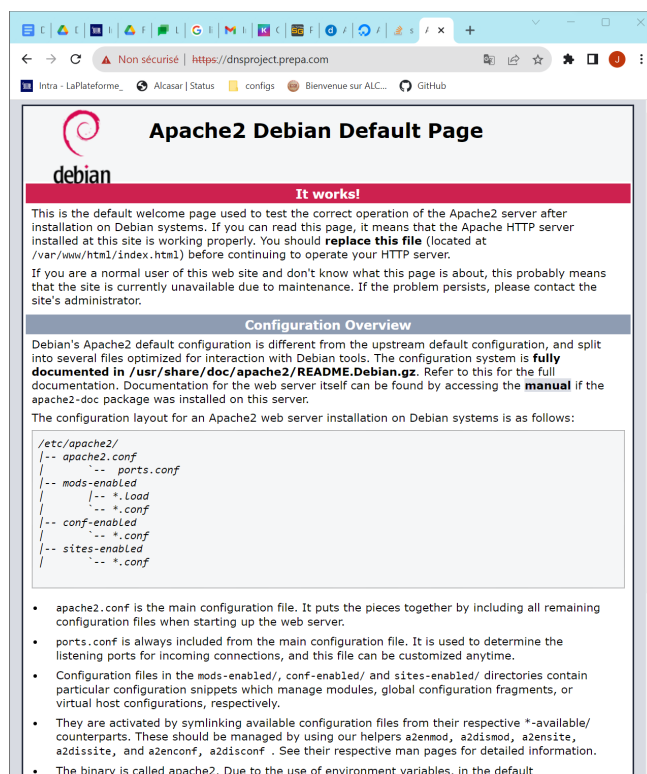
La prochaine étape est de créer un certificat que l'on va signer nous même, avec la commande suivante : "openssl x509 -signkey dnsproject.prepa.com.key -in dnsproject.prepa.com.csr -req -days 365 -out dnsproject.prepa.com.crt. Une fois mon certificat obtenu je vais devoir l'installer sur mon serveur apache. Je vais donc créer un fichier de configuration dans /etc/apache2/sites-enabled/ qui s'appellera dnsproject.prepa.com.conf et qui aura la configuration suivante :

```
GNU nano 7.2 dnsproject.prepa.com.conf
<VirtualHost 192.168.171.129:443>
DocumentRoot /var/www/html
ServerName dnsproject.prepa.com

SSLEngine on
SSLCertificateFile /etc/ssl/dnsproject.prepa.com.crt
SSLCertificateKeyFile /etc/ssl/dnsproject.prepa.com.key
</VirtualHost>
```

On peut voir que j'ai mis mes fichiers dans le dossier /etc/ssl/.

Une fois que cela est fait j'ai dû recharger le module ssl avec la commande "sudo a2enmod ssl" car même si il était censé être activé j'obtenais une erreur qui me disait que la commande SSLEngine était invalide. Recharger le module a bien marché car une fois le port 443 ouvert sur le firewall je peux bien accéder à mon site en https comme montré ci dessous.



J'ai par contre un message d'avertissement de mon navigateur m'informant que mon certificat n'est pas valide et que le site possède donc un risque de sécurité. C'est parce que j'ai créé et signé moi même mon certificat SSL. La différence entre mon certificat signé moi même et un certificat signé par un CA (certificate authority) est que le CA est un tiers de



confiance vérifiant l'identité des personnes déposant un certificat. Les web browsers n'accordent que leur confiance aux certificats signés par un CA, même si le chiffrement des données peut être exactement le même.