

Dans le but de la mise en place un serveur principal qui contiendra plusieurs services réseau, Nous allons aborder deux principaux concepts qui sont DNS et les serveurs web.

Le système DNS (Domain Name System) permet de traduire un nom de domaine en adresse IP et vice versa. Il fonctionne comme un annuaire téléphonique (pages jaunes).

DNS sert de passerelle entre les deux, sa fonction principale est de simplifier la recherche d'un site sur Internet, au lieu de chercher l'adresse IP d'un site, on cherchera son nom de domaine.

Il existe plusieurs types de requêtes DNS :

Les requêtes récursives : sont utilisées par le client résolveur, soit par la machine qui cherche à connaître la conversion nom/IP, afin de résoudre un nom. Cette requête exige soit la bonne réponse, soit une erreur. Elle ne renvoie pas vers un autre serveur.

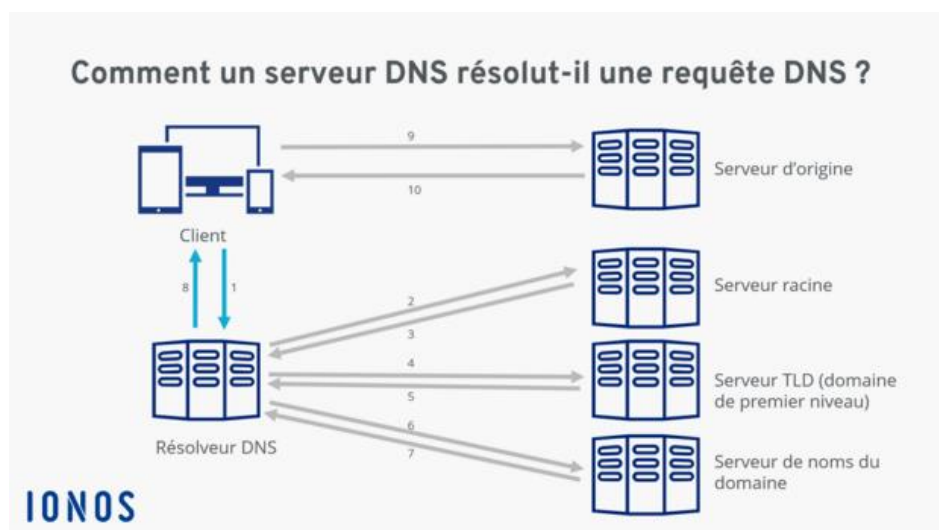
Les requêtes itératives : qui le font. Celles-ci renvoient la meilleure réponse possible, en envoyant une recommandation vers un serveur DNS de référence.

Les requêtes non récursives : le client DNS contacte les serveurs un par un jusqu'à trouver celui contenant les informations nécessaires.

Un serveur web il peut désigner une machine physique qu'un logiciel.

En tant que machine physique, il s'agit d'un ordinateur relié à Internet et en hébergeant des ressources qu'ils partagent. La plupart du temps, ils ne disposent d'aucun périphérique et n'ont pas d'interface graphique. Cependant on y retrouve une grande capacité de stockage.

En tant que logiciel, un serveur est un ensemble de programmes qui permet de faire fonctionner un site web et le rendre public.



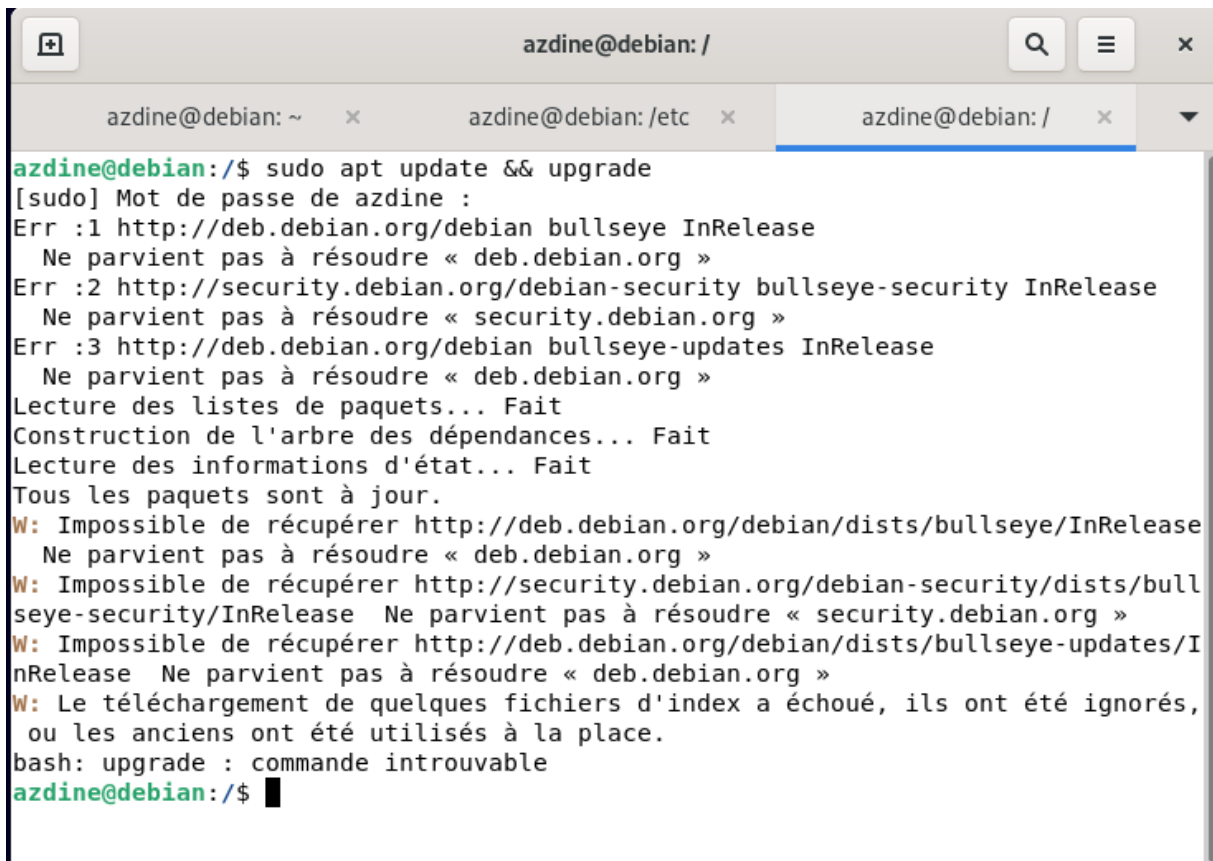
JOB 01 :

Pour l'installation d'une VM avec une interface graphique, il faut dans le premier temps télécharger l'ISO de Debian, puis lancer l'installation avec l'aide de la VMWare, après avoir choisi la configuration nécessaire pour notre VM (espace de stockage, RAM, etc.)

JOB 02 :

Pour ce job, il nous est demandé d'installer un serveur web et nous avons opté pour Apache 2.

Avant l'installation il est conseillé de mettre à jour les paquets avec la commande suivante :



```
azdine@debian: /  
azdine@debian: ~ x azdine@debian: /etc x azdine@debian: / x  
azdine@debian:/$ sudo apt update && upgrade  
[sudo] Mot de passe de azdine :  
Err :1 http://deb.debian.org/debian bullseye InRelease  
  Ne parvient pas à résoudre « deb.debian.org »  
Err :2 http://security.debian.org/debian-security bullseye-security InRelease  
  Ne parvient pas à résoudre « security.debian.org »  
Err :3 http://deb.debian.org/debian bullseye-updates InRelease  
  Ne parvient pas à résoudre « deb.debian.org »  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Tous les paquets sont à jour.  
W: Impossible de récupérer http://deb.debian.org/debian/dists/bullseye/InRelease  
  Ne parvient pas à résoudre « deb.debian.org »  
W: Impossible de récupérer http://security.debian.org/debian-security/dists/bullseye-security/InRelease  Ne parvient pas à résoudre « security.debian.org »  
W: Impossible de récupérer http://deb.debian.org/debian/dists/bullseye-updates/InRelease  Ne parvient pas à résoudre « deb.debian.org »  
W: Le téléchargement de quelques fichiers d'index a échoué, ils ont été ignorés, ou les anciens ont été utilisés à la place.  
bash: upgrade : commande introuvable  
azdine@debian:/$
```

On procède par la suite à l'installation de Apache2 avec la commande suivante sur le terminale :

```
azdine@debian: /
azdine@debian: ~ x azdine@debian: /etc x azdine@debian: / x
azdine@debian:/$ sudo apt install apache2
```

Ensuite, il faut afficher la liste des pare-feux et autoriser le trafic sur le port 80 avec les commandes suivantes sur le terminale

```
azdine@debian: /
azdine@debian: ~ x azdine@debian: /etc x azdine@debian: / x
azdine@debian:/$ sudo ufw app list
Available applications:
AIM
Bind9
Bonjour
CIFS
CUPS
DNS
Deluge
IMAP
IMAPS
IPP
KTorrent
Kerberos Admin
Kerberos Full
Kerberos KDC
Kerberos Password
LDAP
LDAPS
LPD
MSN
MSN SSL
Mail submission
NFS
OpenSSH
POP3
POP3S
PeopleNearby
SMTP
SSH
Socks
Telnet
Transmission
Transparent Proxy
VNC
WWW
WWW Cache
WWW Full
WWW Secure
XMPP
Yahoo
qBittorrent
svnserve
azdine@debian:/$
```

```
azdine@debian: ~
azdine@debian:~$ sudo ufw allow 'www'
[sudo] Mot de passe de azdine :
Rules updated
Rules updated (v6)
```

Ensuite, on affiche l'adresse de notre hostname puis on l'insert dans le fichier /etc/hosts avec le nom de domaine « dnsproject.prepa.com »

```
azdine@debian: ~  
azdine@debian:~$ hostname -i  
127.0.1.1  
azdine@debian:~$ sudo nano /etc/hosts
```

```
GNU nano 5.4 /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    debian.azdine  debian  
127.0.0.1    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  
  
[ Lecture de 7 lignes ]  
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement  
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne
```

Sur notre moteur de recherche de notre VM on saisit l'adresse 127.0.0.1 ou bien le nom de domaine dnsproject.prepa.com , nous aurons la même page d'accueil.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Nous pouvons vérifier par la suite que le serveur est bien actif

```
azdine@debian: ~  
azdine@debian:~$ sudo systemctl status apache2  
[sudo] Mot de passe de azdine :  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>  
   Active: active (running) since Sat 2022-11-12 18:40:21 CET; 46min ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Process: 585 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC>  
 Main PID: 623 (apache2)  
    Tasks: 55 (limit: 4622)  
   Memory: 12.0M  
      CPU: 230ms  
   CGroup: /system.slice/apache2.service  
           └─623 /usr/sbin/apache2 -k start  
             └─625 /usr/sbin/apache2 -k start  
               └─626 /usr/sbin/apache2 -k start  
  
nov. 12 18:40:21 debian systemd[1]: Starting The Apache HTTP Server...  
nov. 12 18:40:21 debian systemd[1]: Started The Apache HTTP Server.  
lines 1-16/16 (END)
```

JOB 03 :

Il existe différents types de serveurs, ils permettent l'affichage des sites web, parmi les serveurs les plus utilisés sont Apache et Nginx.

Apache : il s'agit du serveur qui héberge le plus de sites web au monde. On estime qu'environ la moitié des sites y sont hébergés, c'est logiciel open source, gratuit et disponible sur Linux, Mac et Windows, c'est un serveur modulable et souvent mis à jour et dispose de nombreux correctif de sécurité.

Cependant Apache est plus adapté aux serveurs ayant un trafic plutôt modéré afin que ces performances ne soient pas altérées, ainsi qu'il ne faut pas trop ajouter de configurations pour ne pas l'exposer à des failles de sécurité

Nginx : il est le deuxième serveur le plus utilisé au monde, il permet des multiples connexions simultanées au serveur, il nécessite peu de ressources donc moins de coûts d'hébergement, il peut générer plus de 10 000 requêtes sans saturer la RAM du serveur, à titre d'exemple Netflix et Airbnb sont hébergés sur Nginx.

En revanche, l'assistante est moins rapide en cas de problème et le contenu dynamique est généralement géré par des logiciels tiers, d'où son utilisation est moins répondu qu'Apache.

Tomcat : il est développé par la même société qu'Apache, il est principalement utilisé pour les applications Java, mais il est moins efficace qu'Apache car il est moins configurable.

IIS : développé par Microsoft, il est utilisable uniquement sur Windows, et besoin d'une licence Microsoft pour une utilisation commerciale.

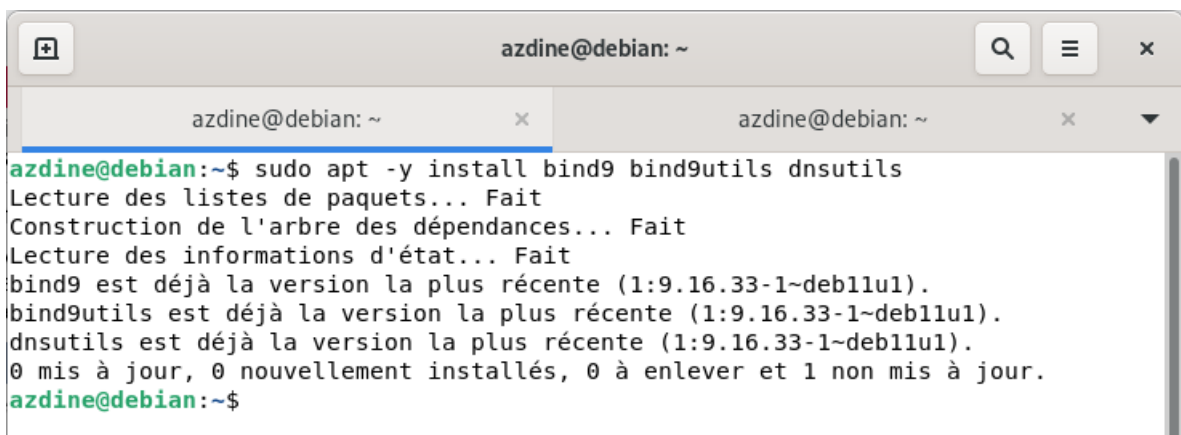
Node.js : ce serveur est multiplateforme et open source, il fonctionne avec Javascript, C++, il est très gourmand en CPU, il héberge notamment Twitter, eBay, Spotify...etc.

Lighttpd : il fait partie des serveurs les plus utilisés également, il conjugue à la fois le rapport fonctionnalités/performances optimales, il supporte la plus part de langages de programmation, il est utilisé par exemple par Youtube, Myspace.

[LiteSpeed](#): il est moins gourmand en terme de ressources et plus rapide qu'Apache et Nginx, il est capable de gérer les pics de connexions. Il permet aux sites qu'il héberge un meilleur référencement sur les différents moteurs de recherche, cependant il est payant.

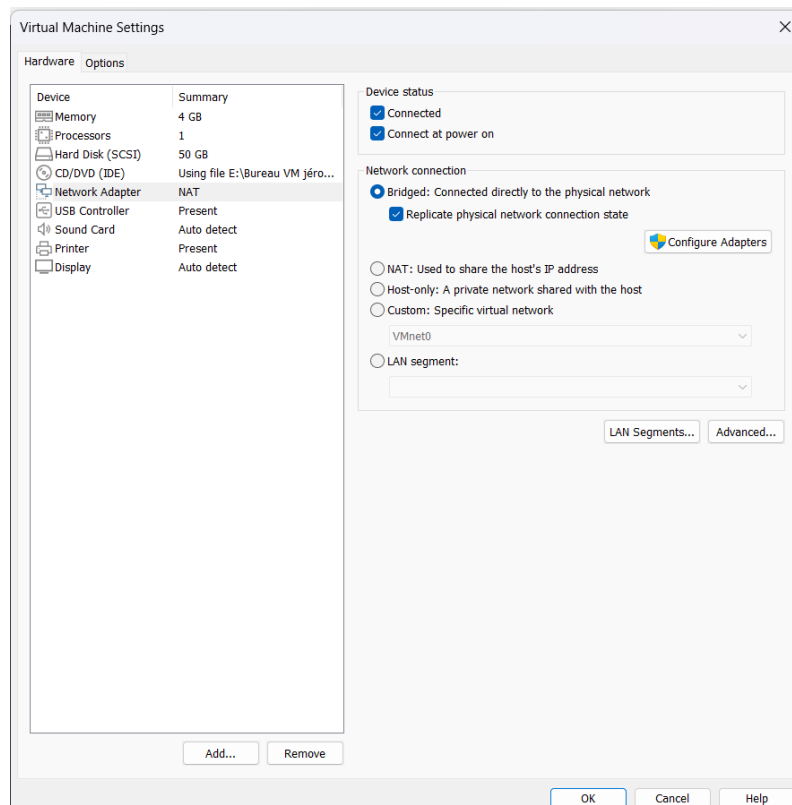
JOB 04 /06:

Il faut mettre en place, pour cela nous allons utiliser bind9 (Berkeley Internet Name Daemon).



```
azdine@debian: ~  
azdine@debian: ~  
azdine@debian:~$ sudo apt -y install bind9 bind9utils dnsutils  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
bind9 est déjà la version la plus récente (1:9.16.33-1~deb11u1).  
bind9utils est déjà la version la plus récente (1:9.16.33-1~deb11u1).  
dnsutils est déjà la version la plus récente (1:9.16.33-1~deb11u1).  
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.  
azdine@debian:~$
```

Il faut mettre sa connexion réseau en bridge




```
azdine@debian: /etc/bind
azdine@debian: ~$ cd /etc/bind
azdine@debian:/etc/bind$ sudo cp db.local direct
```

```
GNU nano 5.4 direct
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      dnsproject.prepa.com. debian.dnsproject.prepa.com. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       localhost.
Debian    IN      A        10.10.30.183
www       IN      CNAME    debian.dnsproject.prepa.com.

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire   ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne
```

```
azdine@debian: /etc/bind
azdine@debian:/etc/bind$ sudo nano direct
[sudo] Mot de passe de azdine :
Désolé, essayez de nouveau.
[sudo] Mot de passe de azdine :
azdine@debian:/etc/bind$ hostname
debian
azdine@debian:/etc/bind$ sudo cp direct inverse
azdine@debian:/etc/bind$ sudo nano inverse
```

```
azdine@debian: /etc/bind
GNU nano 5.4 inverse
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      dnsproject.prepa.com. debian.dnsproject.prepa.com. (
                                2           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200     ; Expire
                                604800 )    ; Negative Cache TTL
;
@         IN      NS       localhost.
Debian    IN      A        10.10.30.183
www       IN      CNAME    debian.dnsproject.prepa.com.
```

[Lecture de 14 lignes]

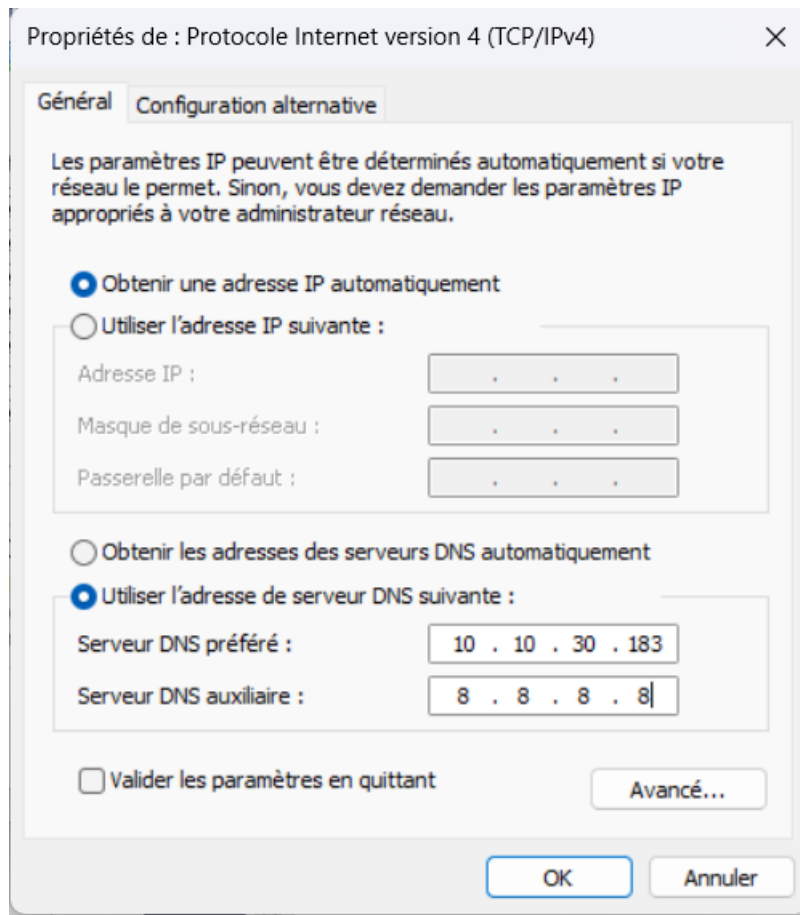
^G Aide	^O Écrire	^W Chercher	^K Couper	^T Exécuter	^C Emplacement
^X Quitter	^R Lire fich.	^N Remplacer	^U Coller	^J Justifier	^_ Aller ligne

```
azdine@debian: /etc/bind
GNU nano 5.4 named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "dnsproject.prepa.com" IN {
    type master;
    file "/etc/bind/direct";
};
zone "10.10.in-addr.arpa" IN {
    type master;
    file "/etc/bind/inverse";
};

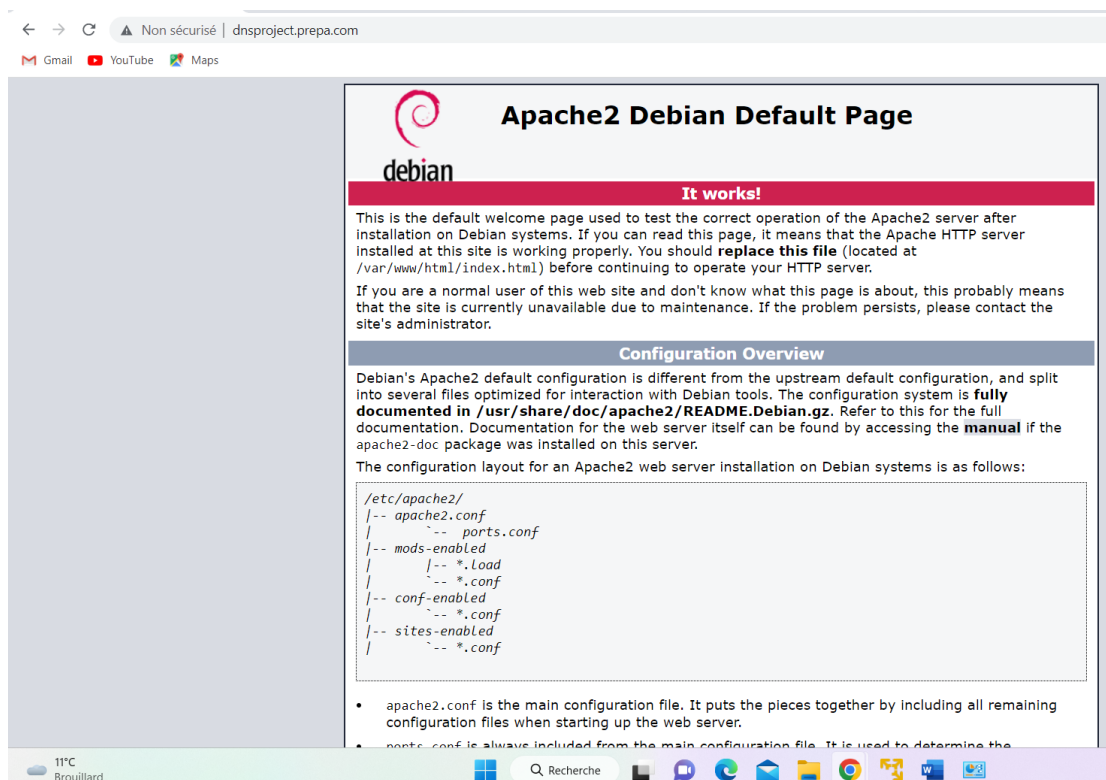
[ Lecture de 16 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^N Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

```
azdine@debian: /etc
GNU nano 5.4 resolv.conf
# Generated by NetworkManager
search dnsproject.prepa.com
nameserver 10.10.30.183

[ Lecture de 3 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^N Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```



Comme on peut le constater que cela fonctionne, car la page s'ouvre sur le moteur de recherche de Windows et non celui de la VM



JOB 05:

Le nom de domaine identifie un site internet et constitue un moyen de localisation géographique ou de son type d'activité, par exemple un site avec une extension « .fr » est un site venant de France ; « .eu » pour l'union européenne ...etc ce sont des extensions dites nationales.

Ils existent des noms de domaines génériques (à vocation internationale) selon le nom de domaine souhaité, il faut s'adresser à différents organismes gestionnaires responsables.

C'est L'IANA (Internet Assigned Numbers Authority) qui permet la reconnaissance officielle de ces trois types de Top Level Domain :

gTLD (domaines génériques de premier niveau), Il regroupe toutes les extensions indépendantes du pays d'origine, telles que .com, .org ou .net.

ccTLD celles-ci regroupent toutes celles provenant du nom d'un pays.

nTLD comportent au moins trois caractères et peuvent être libres. Pour obtenir un nom de domaine, il faut le demander à un organisme qui le gère. L'organisme dépend de l'extension que l'on veut demander.

L'attribution des noms de domaine suit la règle du premier arrivé premier servi. De plus, il faut payer, ces services ne sont pas gratuits. Il faudra aussi passer par un hébergeur de domaine, On peut noter qu'il existe un annuaire répertoriant tous les noms de domaine, ainsi que les personnes ou les organisations qui y sont associées. Il s'agit de WHOIS.

JOB 07:

Le serveur DHCP permet l'attribution automatique des adresses IP, il devra donc distribuer des adresses à l'ensemble des machines présentes dans notre réseau local.

Pour commencer il faudrait installer la paquet du serveur DHCP

```
azdine@debian: ~  
azdine@debian:~$ sudo apt install isc-dhcp-server
```

Passant ensuite à la configuration du serveur dhcp :

```
azdine@debian:~$ sudo nano /etc/default/isc-dhcp-server  
azdine@debian:~$
```

```
GNU nano 5.4 /etc/default/isc-dhcp-server *  
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)  
  
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).  
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf  
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf  
  
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).  
#DHCPDv4_PID=/var/run/dhcpd.pid  
#DHCPDv6_PID=/var/run/dhcpd6.pid  
  
# Additional options to start dhcpd with.  
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead  
#OPTIONS=""  
  
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?  
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".  
INTERFACESv4="ens33"  
INTERFACESv6="ens33"  
█  
  
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement  
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

```
azdine@debian:~$ sudo nano /etc/dhcp/dhcpd.conf  
azdine@debian:~$ █
```

```
azdine@debian: ~
GNU nano 5.4 /etc/dhcp/dhcpd.conf
# option broadcast-address 10.254.239.31;
# option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 10.1.1.0 netmask 255.255.255.0 {
    range 10.5.5.26 10.10.1.100;
    option domain-name-servers 10.10.1.1;
    option domain-name "dnsproject.prepa.com";
    option routers 10.10.1.1;
    option broadcast-address 10.10.1.255;
    default-lease-time 86600;
    max-lease-time 72600;
}

# Hosts which require special configuration options can be listed in
# host statements.  If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

#host passacaglia {
# hardware ethernet 0:0:c0:5d:bd:95;
[ 107 lignes écrites ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

```
azdine@debian: ~
azdine@debian:~$ sudo nano /etc/resolv.conf
```

```
azdine@debian: ~
GNU nano 5.4 /etc/resolv.conf
# Generated by NetworkManager
search dnsproject.prepa.com
nameserver 10.10.1.1
```

```
azdine@debian: /etc/network
azdine@debian:/etc/network$ sudo nano interfaces
```

```
azdine@debian: /etc/network
GNU nano 5.4 interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface ens33 inet static
address 10.10.1.1
netmask 255.255.255.0
```

JOB 08:

Après la configuration du serveur DHCP, et puis le client (debian) a une adresse ip correspondant à la range défini sur le serveur et qui a accès à la page apache via le dns.

Le client doit être configuré en réseau interne sur le même réseau.

Pour le Gateway, il s'agira de configurer un routage entre le sous-réseau et le vrai réseau.

JOB 09:

L'utilisation de pare-feu (firewall) permet la surveillance de tout les trafics entrants et sortants dans notre machine, il décide selon la configuration qu'on lui attribue d'autoriser ou de bloquer le trafic.

On peut installer ufw si ce n'est pas déjà installé par défaut, ensuite l'activer avec la commande :

sudo ufw enable.

Nous allons maintenant configurer ufw (/etc/ufw/before.rules)

A terminal window titled 'azdine@debian: /' with search, menu, and close buttons. The terminal shows the command 'sudo nano /etc/ufw/before.rules' being executed. The prompt changes to '[sudo] Mot de passe de azdine :', and then back to 'azdine@debian:/\$' after the password is entered.

```
azdine@debian:/$ sudo nano /etc/ufw/before.rules
[sudo] Mot de passe de azdine :
azdine@debian:/$
```

Désormais il faut changer les 4 lignes après « ok icmp codes input » qui étaient sur « ACCEPT » il faut les remplacer par « DROP ».


```
azdine@debian: /
GNU nano 5.4 /etc/ufw/before.rules *
-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

# allow dhcp client to work

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier
```

Par la suite on peut ouvrir ou fermer des port avec la commande :

ufw allow <port> par exemple `ufw allow port80`

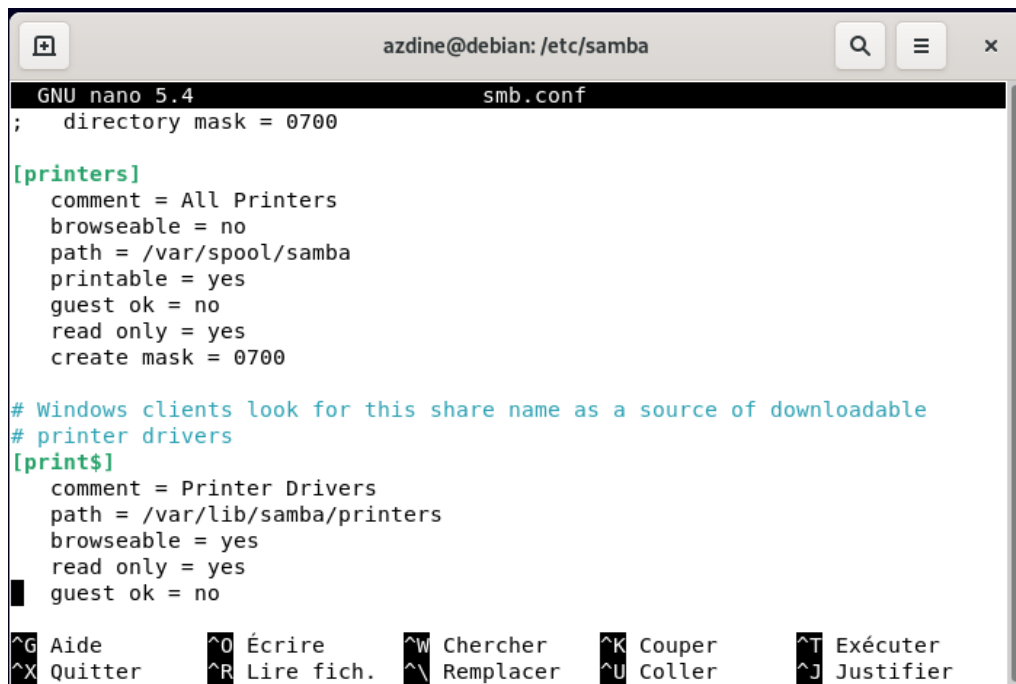
sudo ufw reload, le pare-feu devrait être actif avec la prise en charge de toutes les modifications que nous avons opérés.

JOB 10:

Pour ce job, nous allons utiliser le paquet samba pour le partage des dossiers

```
azdine@debian: /
azdine@debian:/$ sudo apt -y install samba smbclient cifs-utils
```

Puis dans le fichier `smb.conf` on peut créer des dossiers partagés auxquels on peut définir leurs spécificités.



```
GNU nano 5.4          smb.conf
;  directory mask = 0700

[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
printable = yes
guest ok = no
read only = yes
create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier
```

On peut rajouter par la suite pour chaque dossier partagé les valeurs suivantes/

Comment : pour ajouter un commentaire

Path : pour ajouter le chemin du dossier partagé

Writable : si le dossier peut être modifié , rajouter ou supprimer du contenu

Valid users : définir les utilisateurs qui auront accès au dossier.

Pour créer un groupe qui auront accès aux dossiers partagés il faut passer les commandes suivantes :

sudo groupadd smbshare

sudo chgrp -R smbshare 'chemin du dossier'

sudo chmod 2775 'chemin du dossier'

afin de restreindre l'accès au groupe créé il faut passer les commandes suivantes

sudo useradd -M -s /sbin/nologin sambuser

sudo useradd -aG smbshare sambuser

on peut créer un utilisateur local qui n'aura pas forcément besoin d'un mot de passe pour les dossiers partagés en force create/directory mode 770 puis on le rajoute dans le groupe.

sudo smbpasswd -a sambuser

sudo smbpasswd -e sambuser

Par la suite, il faudra autoriser le pare-feu si ce n'est pas déjà fait pour autoriser l'accès à distance pour les dossiers partagés avec la commande /

sudo ufw allow from <adress ip> to any app samba.

il faut redémarrer samba:

sudo testparm (pour vérifier les paramètres samba)

sudo systemctl restart nmbd

Mes clients en sous-réseau devront également installer le paquet samba, après l'installation les dossiers créer et partagés devraient être accessibles via le gestionnaire de fichiers en se connectant avec l'utilisateur déjà créer avant avec son passe word .

Pour aller plus loin:

Dans le but de sécuriser notre page web apache, nous allons configurer une connexion en SSL /TLS, pour cela nous devons d'abord générer une clé/certificat en utilisant openssl puis configurer apache2 pour qu'il utilise ce certificat afin de permettre une connexion sécurisée entre le serveur et le client.

D'abord, on installe openssl si ce n'est pas fait avec la commande :

sudo apt install openssl-server.

On se connecte au serveur via SSH, puis on active le module ssl d'apache

sudo ssh 'utilisateur@<ip> -p22'

sudo a2enmod ssl

on génère la clé et le certificat via openssl

sudo openssl req req -x509 -nodes -days 365 -newkey rsa:2048 |

-keyout /etc/ssl/private/dnsproject.prepa.com.key |

-out /etc/ssl/certs/dnsproject.prepa.com.crt

On configure apache2 dans le dossier /etc/apache2/sites-available afin que l'accès à la page soit sécurisé avec le certificat qu'on vient de générer.