

## DDWS

1. Introduction	2
2. VM	3
3. Apache2	3
4. Différents serveurs Web	5
5. Jobs 4 et 6	6
6. Nom de domaine public	12
7. DHCP	13
8. Passerelle	14
9. Pare-feu	14

## 1. Introduction

Dans ce sujet, nous cherchons à mettre en place un serveur principal qui contiendra plusieurs services réseau. Pour cela, il faut déjà savoir ce que sont DNS et les serveurs web.

Le système DNS (pour Domain Name System) permet de traduire un nom de domaine en adresse IP et vice versa. Il fonctionne comme un annuaire téléphonique et sert de passerelle entre les deux. Sa fonction principale est de simplifier la recherche d'un site sur Internet. Plutôt que de chercher l'adresse IP d'un site, on cherchera son nom (aussi parfois appelé adresse logique). On note que ces deux types d'adresse sont uniques. Pour cela, le DNS passe par un processus de conversion nommé "résolution DNS". C'est le résolveur DNS qui enclenche le processus de recherche DNS.

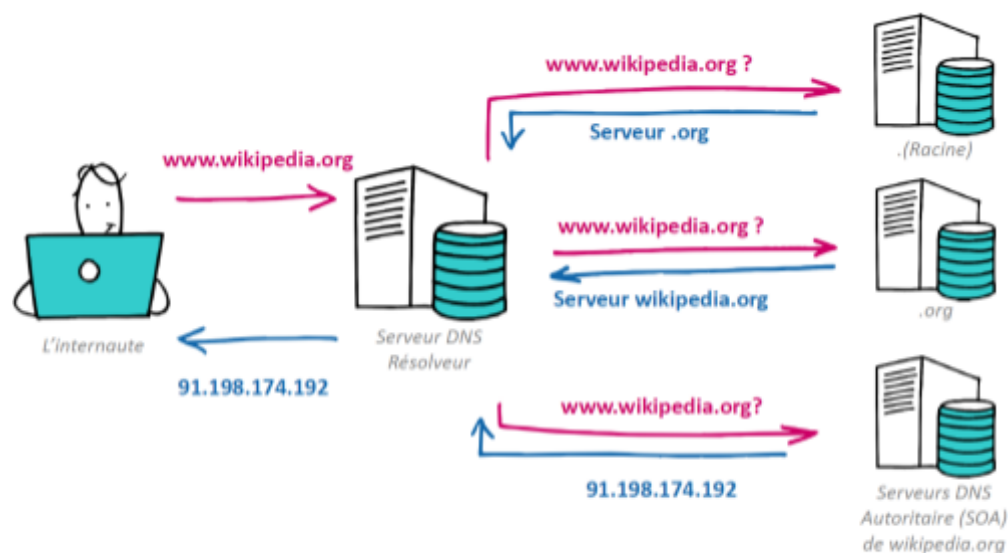


Figure 1 : Schéma de la résolution DNS

Le temps de résolution dépend des capacités matérielles du serveur (CPU, RAM).

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. Ce sont les requêtes itératives qui le font. Celles-ci renvoient la meilleure réponse possible, en envoyant une recommandation vers un serveur DNS de référence. La dernière requête est la non récursive : le client DNS contacte les serveurs un par un jusqu'à trouver celui contenant les informations nécessaires.

Un serveur web peut aussi bien désigner une machine physique qu'un logiciel. En tant que machine physique, il s'agit simplement d'un ordinateur relié à Internet et

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. C'est aussi lui qui va rendre public le site.

## 2. VM

Pour réaliser ce sujet, nous installons un VM Debian avec interface graphique. Il faut dans un premier temps télécharger l'OS Debian, puis lancer l'installation via VMWare. Après avoir attribué la configuration nécessaire à l'hyperviseur, l'installation se lance. On pense alors à cocher serveur web et SSH.

## 3. Apache2

Pour continuer, il nous faut installer un serveur web. Il nous est conseillé d'utiliser Apache2, ce que nous faisons donc. On exécute toutes les commandes en su pour plus de facilité et moins de perte de temps. Les commandes à inscrire dans le terminal de Debian sont :

```
su
apt-get update
apt-get install apache2
```

On remarque cependant qu'Apache est installé par défaut dans Debian. Il faut maintenant le configurer. On se dirige (en su toujours) dans `/etc/apache2/sites-available`. On copie alors le fichier `000-default.conf` que l'on renomme `dnsproject.prepa.com.conf`. Il faut alors configurer ce nouveau fichier pour donner un nom au serveur. Enfin, il faut lancer la commande `a2ensite` pour mettre le tout en ligne. Les commandes et configurations sont les suivantes :

```
su
cd /etc/apache2/sites-available
cp 000-default.conf dnsproject.prepa.com.conf
nano dnsproject.prepa.com.conf
```

On modifie alors le script comme ceci :

```
GNU nano 5.4                                dnsproject.prepa.com.conf
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 dnsproject.prepa.com
    ServerAlias dnsproject.prepa.com

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

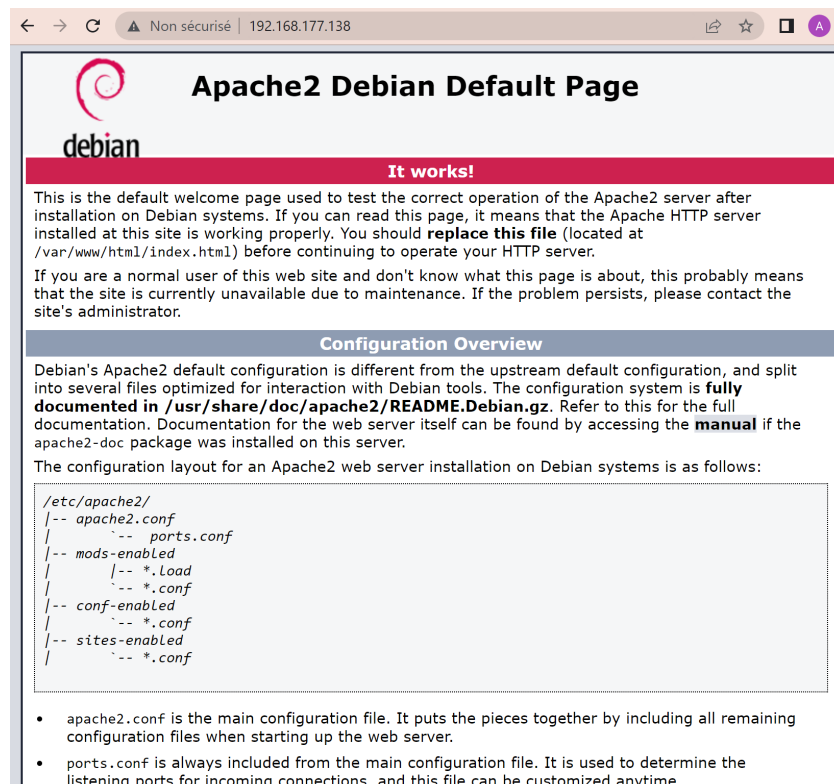
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

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

Puis on lance le site :

a2ensite

Apache est donc configuré. Il est donc possible depuis l'hôte d'ouvrir la page Apache sur un navigateur via l'IP de notre VM :



#### 4. Différents serveurs Web

Il existe plusieurs serveurs Web (au sens logiciel) dont voici quelques exemples :

- ❖ Apache ou Apache HTTP Server : 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 (les statistiques varient selon les sites web).

Il s'agit d'un des serveurs les plus anciens puisque sa première version date de 1995. Il est disponible sur Linux, Mac et Windows et est open source et gratuit, même pour un usage commercial. Il est régulièrement mis à jour et dispose de correctifs de sécurité réguliers. De nombreux langages informatiques y sont pris en charge (Python, PHP, Perl...). Sa structure étant basée sur des modules, il est donc personnalisable et facilement configurable. Cependant, un trop grand nombre de configurations peut aller à l'encontre de la sécurité. De plus, sur les sites web ayant un trafic énorme, on retrouve d'importants problèmes de performances. Cette problématique peut toutefois être résolue avec d'autres serveurs, dont le suivant.

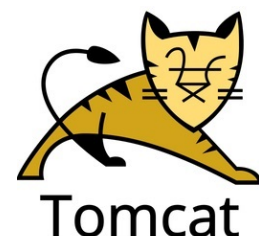


- ❖ Nginx : il est le deuxième serveur le plus utilisé au monde. Sa première version datant de 2004, il permet notamment de multiples connexions sur un serveur en simultané. En effet, il a résolu le problème c10k : les serveurs n'étaient pas capables de gérer plus de 10 000 connexions simultanées.

Un des autres avantages de Nginx est qu'il utilise très peu de ressources.

A titre d'exemple, Netflix, Pinterest ou encore Airbnb sont hébergés sur Nginx.

- ❖ Tomcat ou Apache Tomcat : Ce serveur web est développé par la même société qu'Apache (Apache Software Foundation). Bien que ce soit un serveur HTTP, il est principalement utilisé pour les applications Java (il a d'ailleurs été créé pour celles-ci). Cependant, il est moins efficace qu'Apache et est largement moins configurable que ce dernier ou que les autres serveurs.



- ❖ **Lighttpd** : L'objectif principal de ce serveur est d'avoir un rapport fonctionnalités/performances optimal. Il fait lui aussi partie des serveurs Web les plus utilisés au monde. A titre d'exemple, on y retrouve Youtube et MySpace. Il supporte lui aussi la plupart des langages de programmation.
- ❖ **Node.js** : Ce serveur est open source et multiplateforme. Il fonctionne avec Javascript, C et C++. Twitter, Spotify ou encore eBay y sont hébergés. Cependant, il est très gourmand en CPU.
- ❖ **IIS** : Provenant de Microsoft, ce dernier ne s'exécute que sur Windows. Il est en général fourni avec et a besoin d'une licence de Microsoft pour un fonctionnement commercial. Contrairement à Apache, il dispose d'une équipe employée pour la résolution de problèmes. (Apache vit grâce à sa communauté).
- ❖ **LiteSpeed** : Celui-ci est plus rapide qu'Apache ou Nginx, tout en consommant moins de ressources. Il est en général plus capable de gérer les pics de connexions. Il est possible de passer d'Apache à LiteSpeed sans aucun souci puisqu'aucun fichier de configuration ne nécessite de modification. De plus, il permet aux sites qu'il héberge un meilleur référencement sur les moteurs de recherche. Cependant, il s'agit d'un serveur payant.
- ❖ Et bien d'autres tels que Rails, Netware ou Django



## 5. Jobs 4 et 6

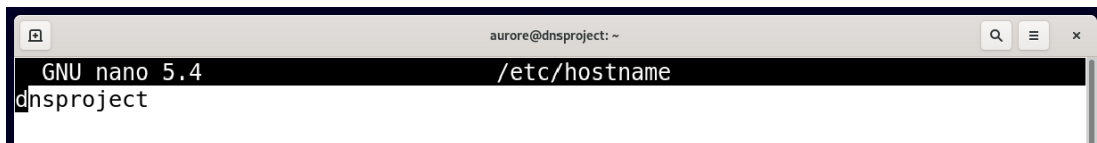
Il faut maintenant mettre en place un DNS. Pour cela nous allons utiliser bind9 (Berkeley Internet Name Daemon). Encore une fois, nous réalisons toutes les commandes en su.

```
su
apt-get update
apt-get install bind9
```

La première étape est de configurer le nom de notre serveur. Il faut donc modifier le fichier **/etc/hostname** puis redémarrer la VM. Le commande **hostname** permet d'afficher le nom de la machine.

**nano /etc/hostname**

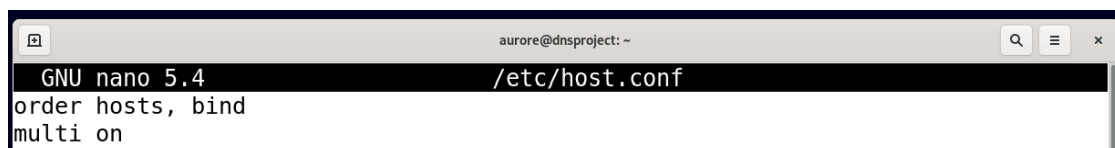
On inscrit dans le script :



```
GNU nano 5.4 /etc/hostname
dnsproject
```

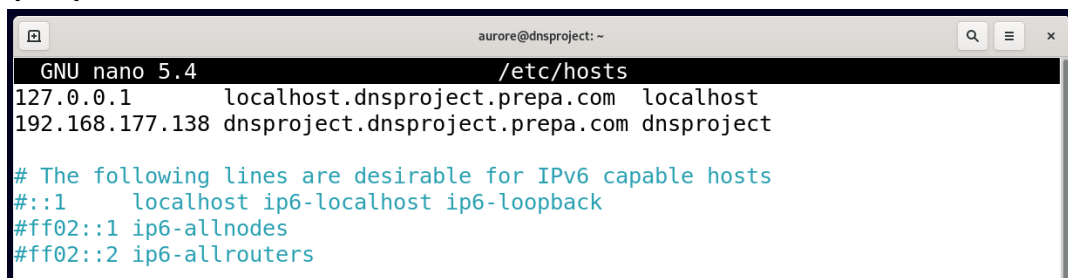
Il faut ensuite configurer le fichier **/etc/host.conf** et **/etc/hosts** pour permettre au serveur d'accueillir des clients.

**nano /etc/host.conf**



```
GNU nano 5.4 /etc/host.conf
order hosts, bind
multi on
```

**nano /etc/hosts**



```
GNU nano 5.4 /etc/hosts
127.0.0.1 localhost.dnsproject.prepa.com localhost
192.168.177.138 dnsproject.dnsproject.prepa.com dnsproject

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

Nous pouvons alors déclarer un nom de domaine dans **/etc/resolv.conf** :



```
GNU nano 5.4 /etc/resolv.conf *
# Generated by NetworkManager
domain dnsproject.prepa.com
search dnsproject.prepa.com
nameserver 192.168.177.138
```

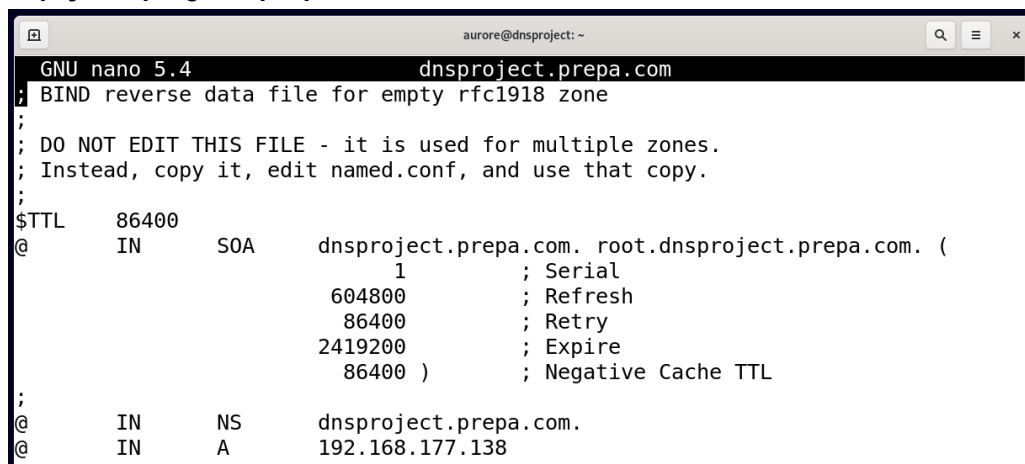
Puis on redémarre le réseau :

**/etc/init.d/networking start**

On remarque que ce fichier se réinitialise à chaque fois que l'on redémarre notre VM. Il faut donc penser à la configurer à chaque démarrage.


Il ne reste donc plus qu'à configurer bind9.

```
cd /etc/bind
cp db.empty dnsproject.prepa.com
```



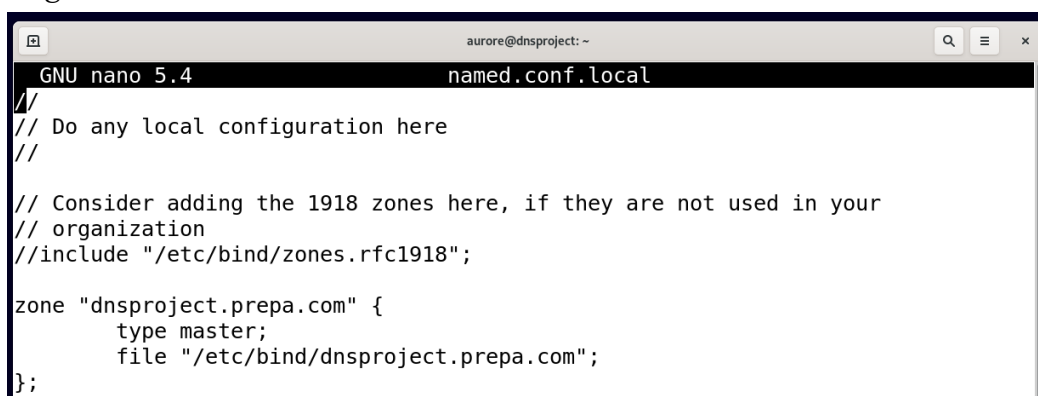
```
GNU nano 5.4 dnsproject.prepa.com
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL      86400
@         IN      SOA      dnsproject.prepa.com. root.dnsproject.prepa.com. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                86400 )    ; Negative Cache TTL
;
@         IN      NS       dnsproject.prepa.com.
@         IN      A        192.168.177.138
```

On crée le fichier inverse, qui permet un retour d'information de la part de la page Web :  
**cp db.127 db.192**



```
GNU nano 5.4 db.192
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      dnsproject.dnsproject.prepa.com root.dnsproject.prepa.c
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )    ; Negative Cache TTL
;
@         IN      NS       dnsproject.
138       IN      PTR      dnsproject.dnsproject.prepa.com.
```

On configure le fichier **named.conf.local** :



```
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" {
    type master;
    file "/etc/bind/dnsproject.prepa.com";
};
```



On s'occupe aussi du fichier **named.conf.options** :

```
GNU nano 5.4 named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        192.168.177.138;
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    auth-nxdomain no;
    version none;
    forward only;
    // listen-on-v6 { any; };
};
```

Puis on redémarre bind9 :

**systemctl bind9 restart**

Notre DNS est mis en place. Il faut toutefois encore faire quelques configurations. Nous devons configurer une IP fixe à notre Debian (celle que nous avons mis dans les configurations) :

Annuler Filaire Appliquer

Détails Identité IPv4 IPv6 Sécurité

**Méthode IPv4**

- ☐ Automatique (DHCP)
- ☒ Manuel
- ☐ Partagée avec d'autres ordinateurs
- ☐ Réseau local seulement
- ☐ Désactiver

**Adresses**

Adresse	Masque de réseau	Passerelle
192.168.177.138	255.255.255.0	

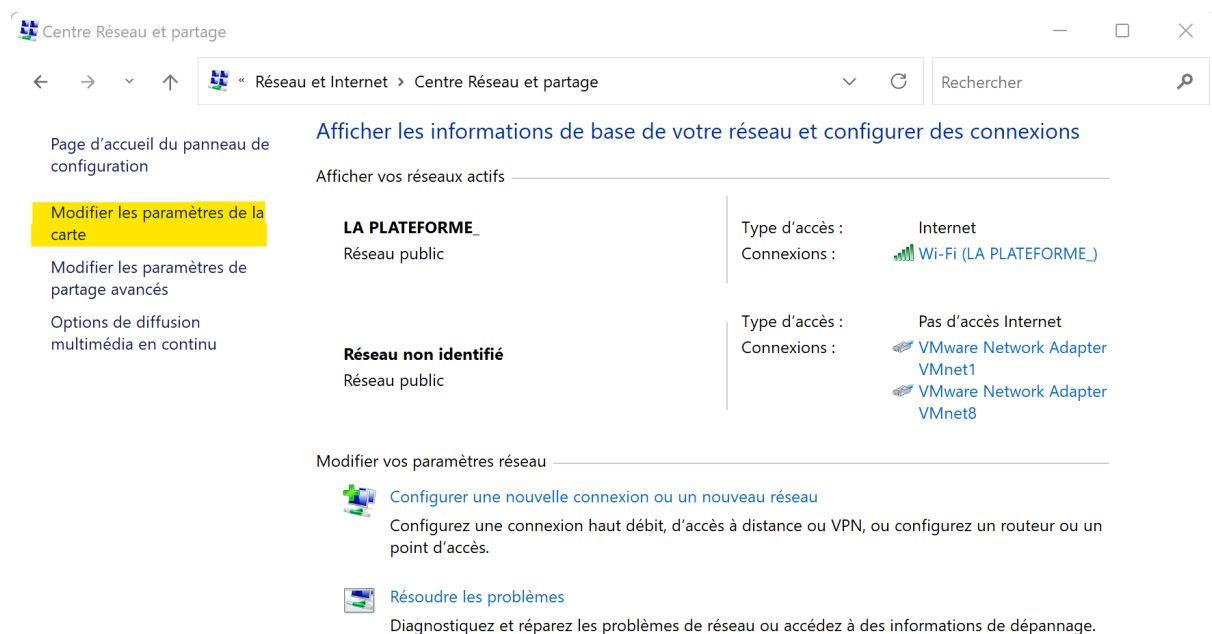
On remarque qu'il est aussi possible de configurer une IP fixe en passant par le terminal. Pour cela, il faut modifier le fichier **etc/network/interfaces** tel quel :

```
auto lo
iface lo inet loopback
```

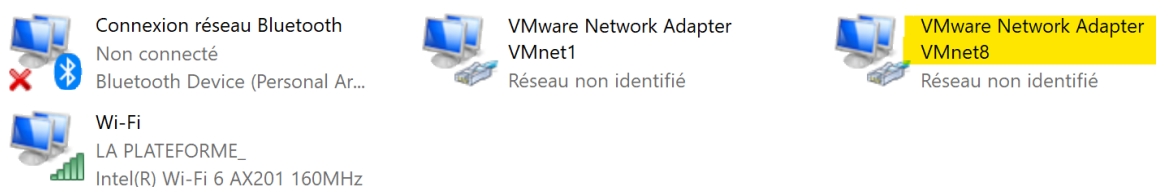
```
auto eth0
iface eth0 inet static
address 192.168.177.138
netmask 255.255.255.0
```

Puis il faudra redémarrer le système.

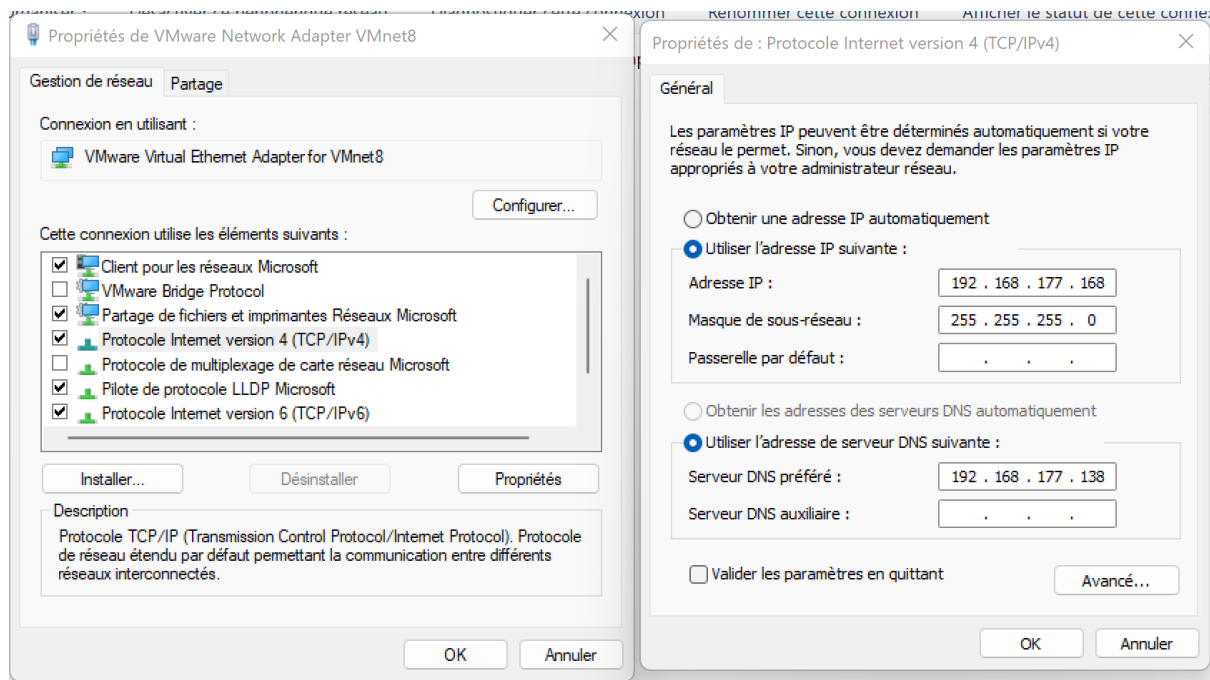
Dans notre hôte (Windows dans mon cas), on passe dans le Panneau de configuration > Centre Réseau et Partage. Puis on modifie les paramètres de la carte :



On choisit VMWare Network Adapter VMnet8 (Clique droit puis Propriétés)



On modifie alors comme ci-dessous :



Tout est alors configuré. Il est possible de ping de notre hôte via l'IP ou le nom de domaine. Il en va de même pour accéder à la page Apache sur un navigateur.

```
C:\Users\auror>ping dnsproject.prepa.com

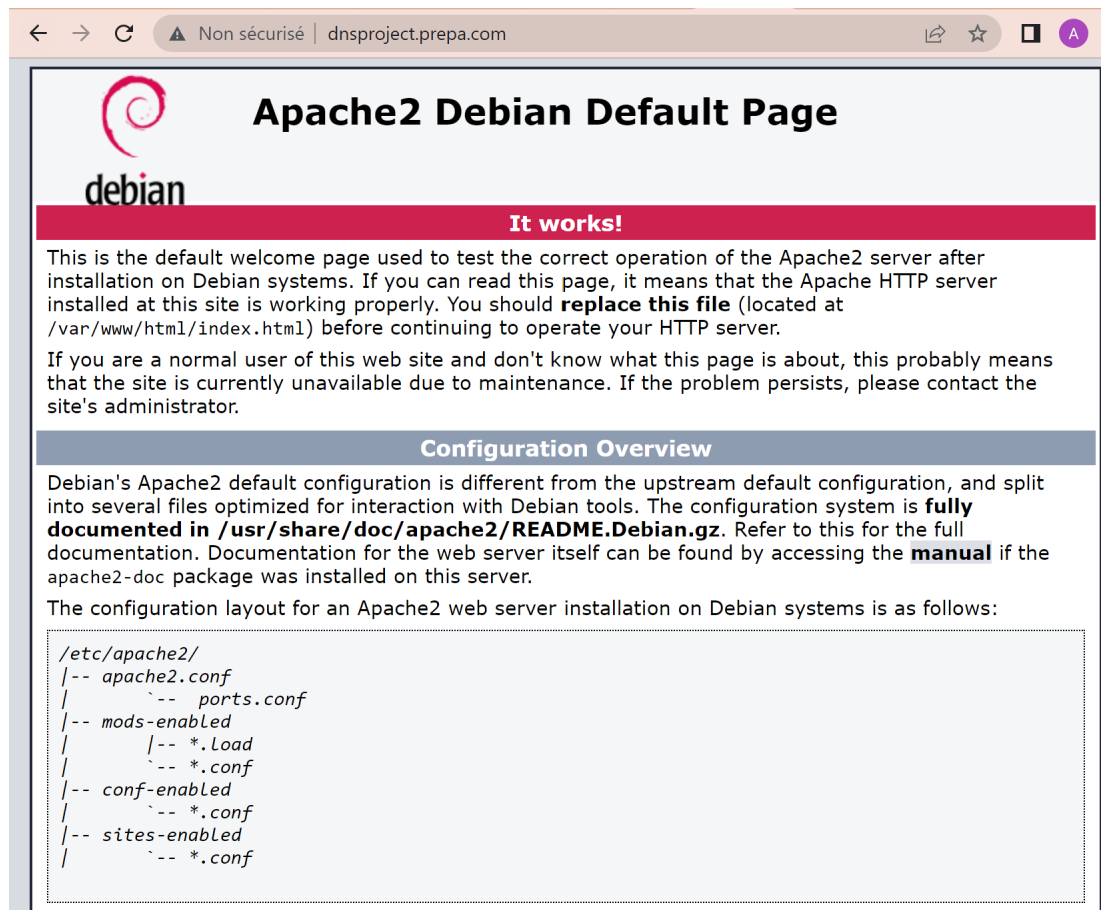
Envoi d'une requête 'ping' sur dnsproject.prepa.com [192.168.177.138] avec 32 octets de données :
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.177.138:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms

C:\Users\auror>ping 192.168.177.138

Envoi d'une requête 'Ping' 192.168.177.138 avec 32 octets de données :
Réponse de 192.168.177.138 : octets=32 temps<1ms TTL=64
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.177.138 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.177.138:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```



## 6. Nom de domaine public

Un nom de domaine permet de donner plusieurs indications. Il informe non seulement sur le nom du site mais aussi sur sa localisation géographique ou sur son type d'activité. En effet, ce sont les lettres situées après le dernier point qui renseignent sur ces informations : les extensions. Par exemple, un site venant de France aura ".fr", pour l'Italie ".it" ou pour l'Union européenne ".eu". Ce sont des extensions dites nationales. Pour ce qui est à vocation internationale, on retrouve ".com" pour les activités commerciales, ".net" pour les entreprises ou ".org" pour les associations ou les ONG). Celles-ci sont dites extensions génériques. Les extensions permettent donc de classer les noms de domaine selon leur fonction.

Il existe trois catégories d'extensions. C'est L'IANA (Internet Assigned Numbers Authority) qui permet la reconnaissance officielle des ces trois types de Top Level Domain.

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

Les extensions géographiques sont ccTLD. Celles-ci regroupent toutes celles provenant du nom d'un pays. Il en existe actuellement 312 différentes.

Les nouvelles extensions génériques (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. Cependant, si un nom de marque a déjà été déposé, il est possible de se voir refuser le nom si l'on ne possède pas la marque. De plus, il faut payer, ces services ne sont pas gratuits. Il faudra aussi passer par un hébergeur de domaine, dont nous avons discuté précédemment.

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. Il est toutefois possible, comme pour les annuaires téléphoniques, de mettre son nom de domaine en liste rouge.

## 7. DHCP

Pour rappel, un DHCP permet l'attribution automatique d'adresses IP. Dans notre cas, il devra donc distribuer des adresses aux machines présentes sur le réseau local.

**su**

**apt install isc-dhcp-server**

**nano /etc/default/isc-dhcp-server**



```
GNU nano 5.4 /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

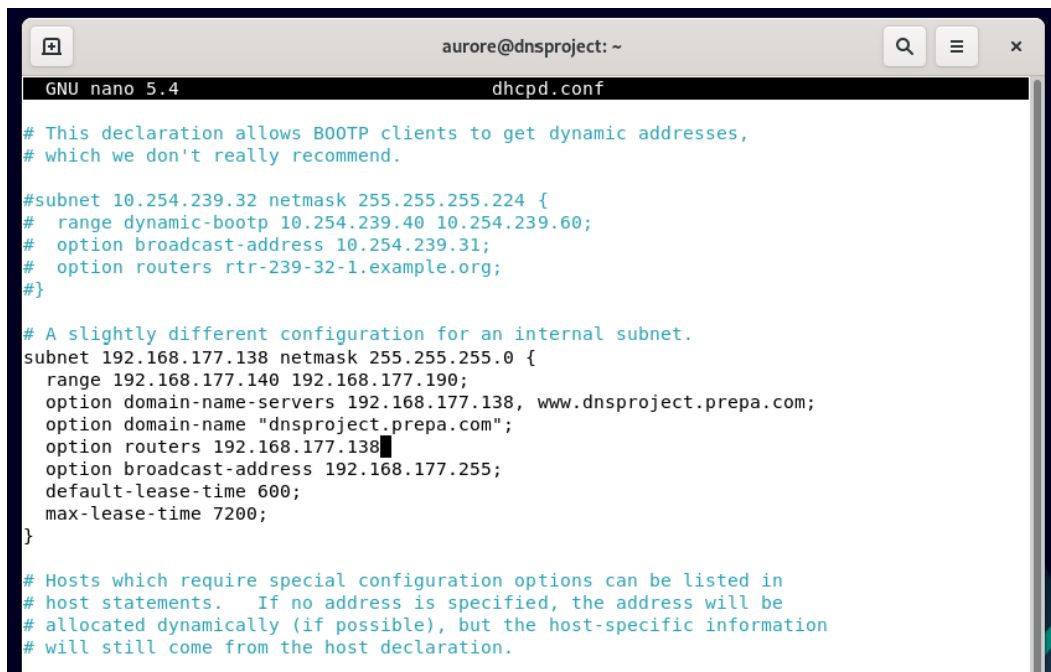
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# 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=enp23s0
INTERFACESv6=enp23s0
```

Puis on configure le fichier **/etc/dhcp/dhcpd.conf** tel quel :



```
GNU nano 5.4 dhcpd.conf

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.177.138 netmask 255.255.255.0 {
  range 192.168.177.140 192.168.177.190;
  option domain-name-servers 192.168.177.138, www.dnsproject.prepa.com;
  option domain-name "dnsproject.prepa.com";
  option routers 192.168.177.138;
  option broadcast-address 192.168.177.255;
  default-lease-time 600;
  max-lease-time 7200;
}

# 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.
```

Enfin, on redémarre le système avec :  
**/etc/init.d/isc-dhcp-server restart**

Il est alors possible de voir le bon fonctionnement (ou non de notre serveur) avec la commande :

**systemctl status isc-dhcp-server.service**

## 8. Passerelle

On veut maintenant que notre serveur serve de passerelle aux autres machines. Celles-ci doivent avoir la possibilité d'accéder à Internet en passant par le Gateway.

## 9. Pare-feu

Un pare-feu (ou firewall) permet la surveillance du trafic entrant et sortant d'une machine. C'est lui qui décide d'autoriser ou de bloquer le trafic selon des règles de sécurité prédéfinies.