Configuration DNS TP Debian 11

Table des matières

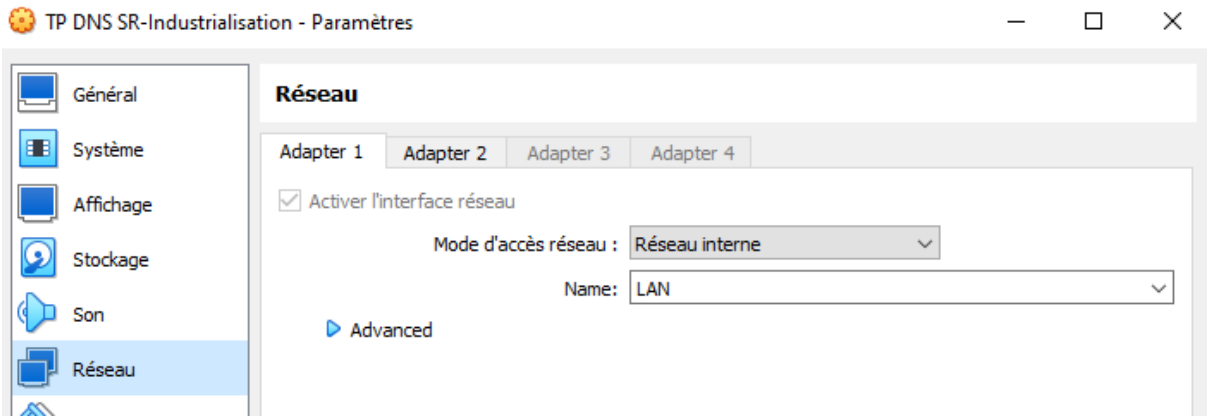
Configuration DNS TP Debian 11 1
Mise en place du Service DNS 2
1. Configuration de l'adressage IP 2
2. Configuration du service DNS : 4
Configuration de la zone DNS : 6
3. Capture d'écran de l'état du service Bind9 7
4. Tests : 8
Tests sur le serveur DNS : 8
Tests sur machine cliente Windows 10 : 9
.....10
5. Tests sur Wireshark, schéma d'échanges des trames : 10
Mise en place Service DHCP : 12
1. Installation des packages pour le service DHCP : 12
2. Configuration du service DHCP : 13
3. Tests Hors accès à internet 15
Tests sur machine Windows 10 : 15
4. Client avec accès internet 17
1. Activation du mode routeur : 17
2. Activation du NAT 17
.....18
3. Automatisation du montage des règles iptables 18
4. Tests sur machine avec accès internet 19

Mise en place du Service DNS

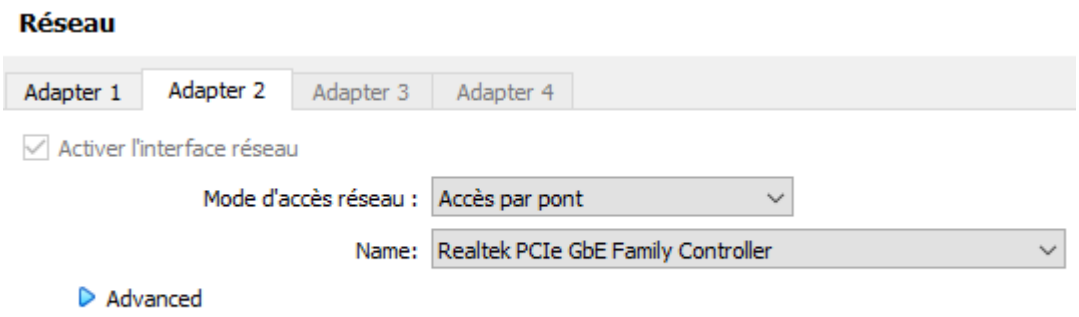
1. Configuration de l'adressage IP

Tout d'abord commençons par configurer nos interfaces réseaux :

- enp0s3 en Interne en fixe 10.0.1.24/24



- enp0s8 en DHCP sur le réseau du lycée



Ensuite sur la machine, connectée en SSH, nous voyons que nous avons notre Adresse IP fournie par le DHCP du réseau du lycée et notre ip fixe :

```
root@dns:/etc/bind# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:ef:a4:84 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.24/24 brd 10.0.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feef:a484/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:da:2f:5c brd ff:ff:ff:ff:ff:ff
    inet 192.168.51.55/24 brd 192.168.51.255 scope global dynamic enp0s8
       valid_lft 10411sec preferred_lft 10411sec
    inet6 fe80::a00:27ff:feda:2f5c/64 scope link
       valid_lft forever preferred_lft forever
```

Et nous installons les packages nécessaires pour la configuration du DNS, dans notre cas nous allons utiliser bind9 :

```
root@dns:/etc/bind# apt install bind9
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 (1:9.18.28-1~deb12u2).
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  linux-image-6.1.0-18-amd64
Veuillez utiliser « apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

Notre serveur DNS va posséder l'adresse ip 10.0.1.24 sur l'interface enp0s3 :

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:ef:a4:84 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.24/24 brd 10.0.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feef:a484/64 scope link
       valid_lft forever preferred_lft forever
```

2. Configuration du service DNS :

Mise en place des prérequis

Pour cela, nous nous rendons dans */etc/bind/named.conf.local* afin de configurer les zones DNS :

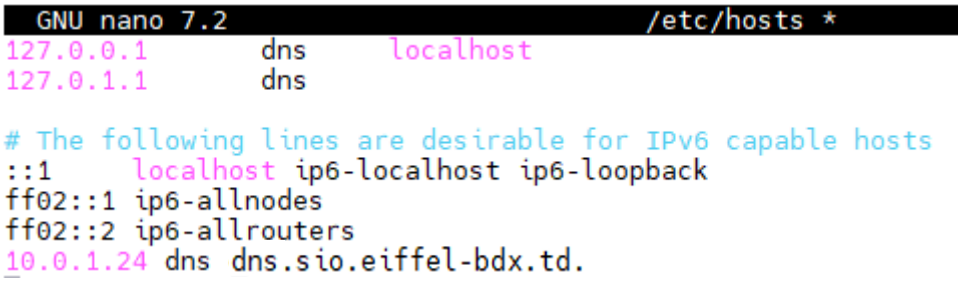
Ma zone dns s'appelle eiffel-bdx.td. , Je dis qu'elle se déclare dans le fichier /etc/bind/db.eiffel-bdx.td . Et que c'est le serveur maitre.

```
//
// 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 "eiffel-bdx.td" IN {
        type master;
        file "/etc/bind/db.eiffel-bdx.td" ;
};
```

On enregistre et on vérifie bien que le nom de notre machine est bien « DNS » pour la configuration plus tard :

```
  GNU nano 7.2                                /etc/hostname
dns
```

Ensuite nous nous rendons dans /etc/hosts afin de dire à notre machine que le serveur DNS est sur les adresses de Loopback (nous-mêmes) mais aussi que l'adresse ip 10.0.1.24 est le Serveur DNS



Configuration de la zone DNS :

Nous nous rendons dans /etc/bind/db.eiffel-bdx.td que nous avons déclaré dans ce fichier :*etc/bind/named.conf.local*.

```
  GNU nano 7.2                                                    db.eiffel-bdx.td *
; Definition de la zone eiffel-bdx.fl
; Le serveur d'autorité est dns.sio.eiffel-bdx.fl (serveur primaire)
; Il est administré par une personne qu'on peut joindre à l'adresse adm@eiffel-bdx.fl

$TTL 21600
@   IN  SOA dns.sio.eiffel-bdx.td. adm@eiffel-bdx.td. (
            3         ; serial - numéro de série (de la 1 à n) modifié à chaque modification de la zone
            3600      ; refresh - intervalle de réplication des serveurs secondaires (en secondes)
            600       ; retry - délai d'attente après une tentative de réplication infructueuse
            3600000   ; expire - délai de validité des répliques de la configuration
            21600 )   ; TTL (Time to Live) - durée de vie en cache des réponses

; avec deux serveurs de noms dans cette zone
@   IN  NS  dns.sio.eiffel-bdx.td.

; déclaration des adresses faisant autorité
localhost.eiffel-bdx.td.    IN  A     127.0.0.1
srv.eiffel-bdx.td.          IN  A     192.168.51.255
proxy.eiffel-bdx.td.        IN  A     10.0.2.2
routeur.eiffel-bdx.td.      IN  A     10.0.2.200
routeur.eiffel-bdx.td.      IN  A     10.0.1.100
client1.eiffel-bdx.td.      IN  A     10.0.1.50
client2.eiffel-bdx.td.      IN  A     10.0.1.51

; glue data
dns.sio.eiffel-bdx.td.      IN  A     10.0.1.24
```

La ligne :

```
@   IN  SOA dns.sio.eiffel-bdx.td. adm@eiffel-bdx.td. (
```

@ veut dire que nous sommes dans le domaine actuel (eiffel-bdx.td)

IN type de classe de l'enregistrement dns « Internet »

SOA veut dire que c'est le serveur DNS qui fait autorité

Et dns.sio.eiffel-bdx.td est le nom du serveur DNS Primaire pour cette zone

admgb.eiffel-bdx.fl. est l'adresse email de l'administrateur et l'@ est remplacé par le .

Tous les paramètres suivant cette ligne sont les paramètres de fonctionnement du serveur, tels que le numéro de série pour la zone DNS, les temps d'essais avant qu'un serveur secondaires tente de se connecter etc....

```
$TTL 21600
eiffel-bdx.td. IN SOA dns.sio.eiffel-bdx.td. admgb.eiffel-bdx.fl. (
    3         ; serial → numéro de série (de 1 à n) modifié à chaque changement ou numéro de version au format yyyymmddhhmm
    36000     ; refresh → intervalle de réplication imposé aux serveurs secondaires (en secondes)
    3600      ; retry → délai d'attente après une tentative de réplication infructueuse
    360000    ; expire → délai de validité des répliques de la configuration
    21600     ; TTL (Time To Live) → durée de vie en cache des réponses
)
```

Ensuite nous partons sur la partie la plus intéressante :

```
; déclaration des adresses faisant autorité
localhost.eiffel-bdx.td.    IN  A     127.0.0.1
srv.eiffel-bdx.td.          IN  A     192.168.51.255
proxy.eiffel-bdx.td.        IN  A     10.0.2.2
routeur.eiffel-bdx.td.      IN  A     10.0.2.200
routeur.eiffel-bdx.td.      IN  A     10.0.1.100
client1.eiffel-bdx.td.      IN  A     10.0.1.50
client2.eiffel-bdx.td.      IN  A     10.0.1.51

; glue data
dns.sio.eiffel-bdx.td.      IN  A     10.0.1.24
```

localhost.eiffel-bdx.td. IN A 127.0.0.1 - Associe l'adresse IP de loopback 127.0.0.1 au nom localhost.

pve.eiffel-bdx.td. IN A 192.168.51.255 - Associe l'adresse IP 192.168.51.255 au nom pve.

proxy.eiffel-bdx.td. IN A 10.0.2.2 - Associe l'adresse IP 10.0.2.2 au nom proxy

srv.eiffel-bdx.td. IN A 10.0.2.1 - Associe l'adresse IP 10.0.2.1 au nom srv.

routeur.eiffel-bdx.td. IN A 10.0.2.200 - Associe l'adresse IP 10.0.2.200 au nom routeur.

routeur.eiffel-bdx.td. IN A 10.0.1.200 - Associe aussi l'adresse IP 10.0.1.200 au nom routeur.

client.eiffel-bdx.td. IN A 10.0.1.216 - Associe l'adresse IP 10.0.1.216 au nom client.

dns.sio.eiffel-bdx.td. IN A 10.0.1.24 - Associe l'adresse IP 10.0.1.24 au nom dns.sio

Il faut redémarrer le service Bind9 afin d'enregistrer les modifications

3. Capture d'écran de l'état du service Bind9

```
user@dns:~$ systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-09-13 19:20:30 CEST; 3min 23s ago
       Docs: man:named(8)
   Main PID: 3156 (named)
     Status: "running"
      Tasks: 8 (limit: 2307)
     Memory: 36.6M
        CPU: 28ms
     CGroup: /system.slice/named.service
             └─3156 /usr/sbin/named -f -u bind
```

4. Tests :

Tests sur le serveur DNS :

Zone DNS :

Sur le serveur DNS :

```
root@dns:/etc/bind# nslookup dns.sio.eiffel-bdx.td
Server:         10.0.1.24
Address:        10.0.1.24#53

Name:   dns.sio.eiffel-bdx.td
Address: 10.0.1.24
```

Sur le routeur (il retournera deux adresses)

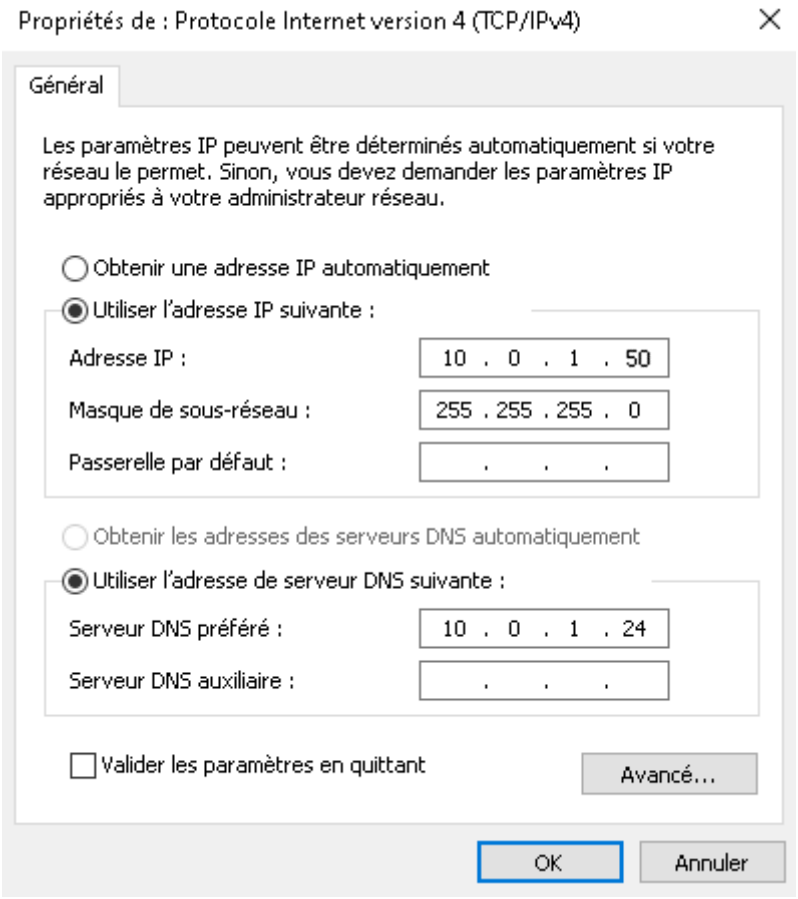
```
root@dns:/etc/bind# nslookup routeur.eiffel-bdx.td
Server:         10.0.1.24
Address:        10.0.1.24#53

Name:   routeur.eiffel-bdx.td
Address: 10.0.1.200
Name:   routeur.eiffel-bdx.td
Address: 10.0.2.200
```

Sur le client Windows 10

```
root@dns:/etc/bind# nslookup client.eiffel-bdx.td
Server:         10.0.1.24
Address:        10.0.1.24#53

Name:   client.eiffel-bdx.td
Address: 10.0.1.50
```

Tests sur machine cliente Windows 10 :

On change ses parametres réseaux :



On teste le serveur DNS :

Interroger le serveur DNS :

```
C:\Users\user>nslookup dns.sio.eiffel-bdx.td
Serveur  : dns.sio.eiffel-bdx.td
Address:  10.0.1.24

Nom :    dns.sio.eiffel-bdx.td
Address:  10.0.1.24
```

Interroger le client et le routeur :

```
C:\Users\sio>nslookup routeur.eiffel-bdx.td
Serveur :  dns.sio.eiffel-bdx.td
Address:  10.0.1.24

Nom :    routeur.eiffel-bdx.td
Addresses:  10.0.1.200
          10.0.2.200
```

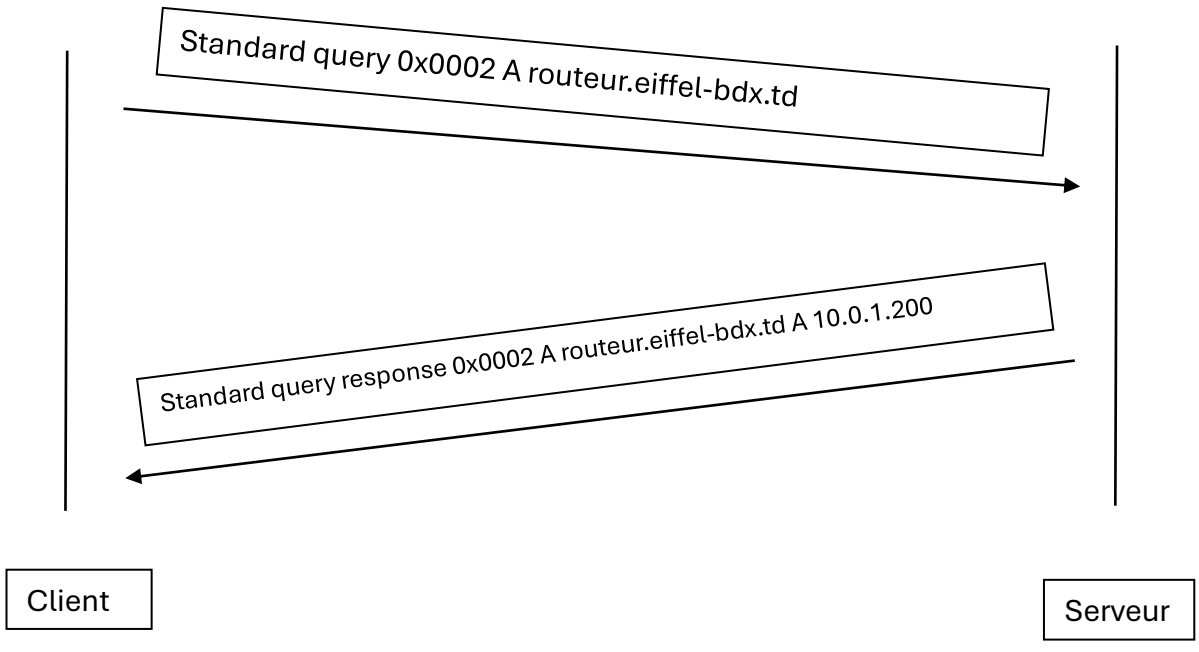
```
C:\Users\sio>nslookup client.eiffel-bdx.td
Serveur :  dns.sio.eiffel-bdx.td
Address:  10.0.1.24

Nom :    client.eiffel-bdx.td
Address:  10.0.1.50
```

5. Tests sur Wireshark, schéma d'échanges des trames :

On peut voir les échanges de trames notamment entre la résolution de l'adresse IP en nom de domaine avec :

Routeur :

5	5.521162	10.0.1.216	10.0.1.24	DNS	81 Standard query 0x0002 A routeur.eiffel-bdx.td
6	5.521488	10.0.1.24	10.0.1.216	DNS	113 Standard query response 0x0002 A routeur.eiffel-bdx.td A 10.0.1.200 A 10.0.2.200

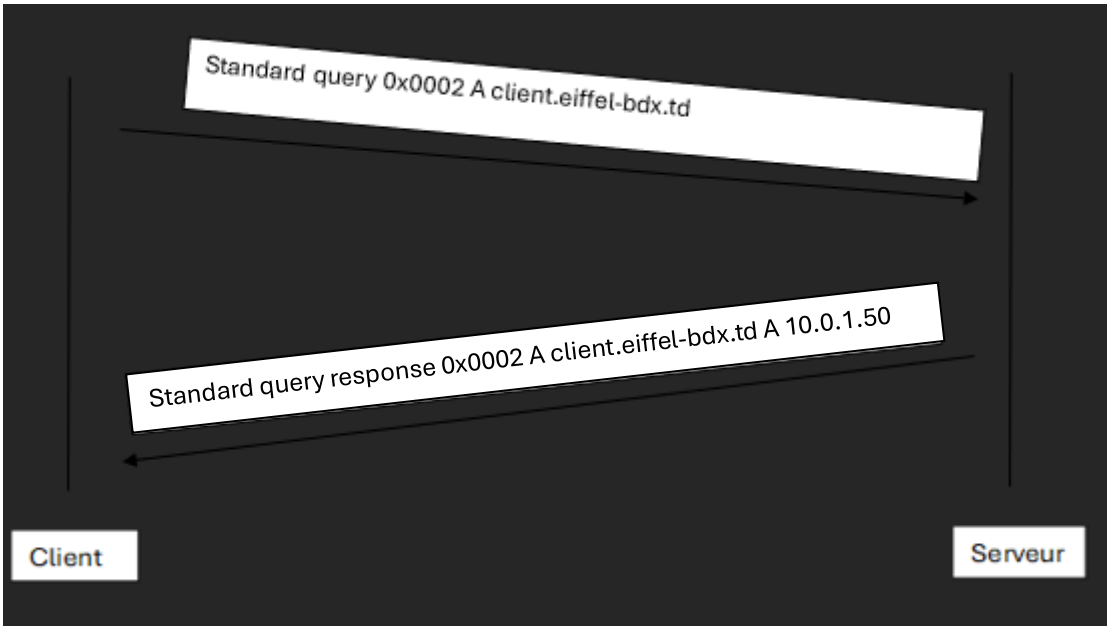


On peut voir l'échange de trames entre Le DNS et le demandeur et que ça résout le nom Routeur.eiffel-bdx.td en 10.0.1.200 et 10.0.2.200

Client :

On peut voir l'échange de trames entre Le DNS et le demandeur et que ça résout le nom client.eiffel-bdx.td en 10.0.1.50

3	0.005284	10.0.1.50	10.0.1.24	DNS	80 Standard query 0x0002 A client.eiffel-bdx.td
4	0.005759	10.0.1.24	10.0.1.50	DNS	96 Standard query response 0x0002 A client.eiffel-bdx.td A 10.0.1.50



Mise en place Service DHCP :

1. Installation des packages pour le service DHCP :

```
root@dns:/home/user# apt install isc-dhcp-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  policycoreutils selinux-utils
Paquets suggérés :
  policykit-1 isc-dhcp-server-ldap ieee-data
Les NOUVEAUX paquets suivants seront installés :
  isc-dhcp-server policycoreutils selinux-utils
0 mis à jour, 3 nouvellement installés, 0 à enlever et 1 non mis à jour.
Il est nécessaire de prendre 1 766 ko dans les archives.
Après cette opération, 7 818 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n]
```

2. Configuration du service DHCP :

Tout d'abord nous allons dans ce fichier : /etc/default/isc-dhcp-server pour dire au service DHCP que nous écoutons sur cette interface et uniquement elle (un service DHCP écoute sur toutes les interfaces connectées) :

```
  GNU nano 7.2                  /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="enp0s3"
#INTERFACESv6=""
```

Ici nous disons que la configuration du service se passe dans le fichier /etc/dhcp/dhcpd.conf :

Et que nous écoutons sur notre interface enp0s3

Nous nous rendons sur notre fichier de configuration /etc/dhcp/dhcpd.conf :

Et nous allons mettre ces paramètres :

```
  GNU nano 7.2
subnet 10.0.1.0 netmask 255.255.255.0 {
authoritative;
range 10.0.1.50 10.0.1.60;
default-lease-time 3600;
max-lease-time 3600;
option subnet-mask 255.255.255.0;
option broadcast-address 10.0.1.255;
option routers 10.0.1.24;
option domain-name-servers 10.0.1.24;
option domain-name "eiffel-bdx.td";
}
```

La ligne subnet signifie que nous mettons une plage d'adresse ip pour ce sous réseau :

La ligne authoritative signifie que ce serveur est le serveur qui va fournir les adresses IP

La ligne range est la ligne qui définit quelles adresses vont être attribuées aux clients

La ligne default-lease-time 3600 signifie que le temps du bail est d'une heure

La ligne max-lease-time définit la durée maximale du bail en secondes, soit une heure

La ligne option subnet mask permet d'indiquer aux clients quel masque ils vont avoir

La ligne option broadcast permet d'indiquer aux clients quelle adresses de broadcast ils vont avoir

La ligne option routers définit la passerelle par défaut :

La ligne option domaine-name-server indique vers quel DNS les clients doivent pointer

La ligne option domaine-name indique vers quel DNS les clients doivent pointer

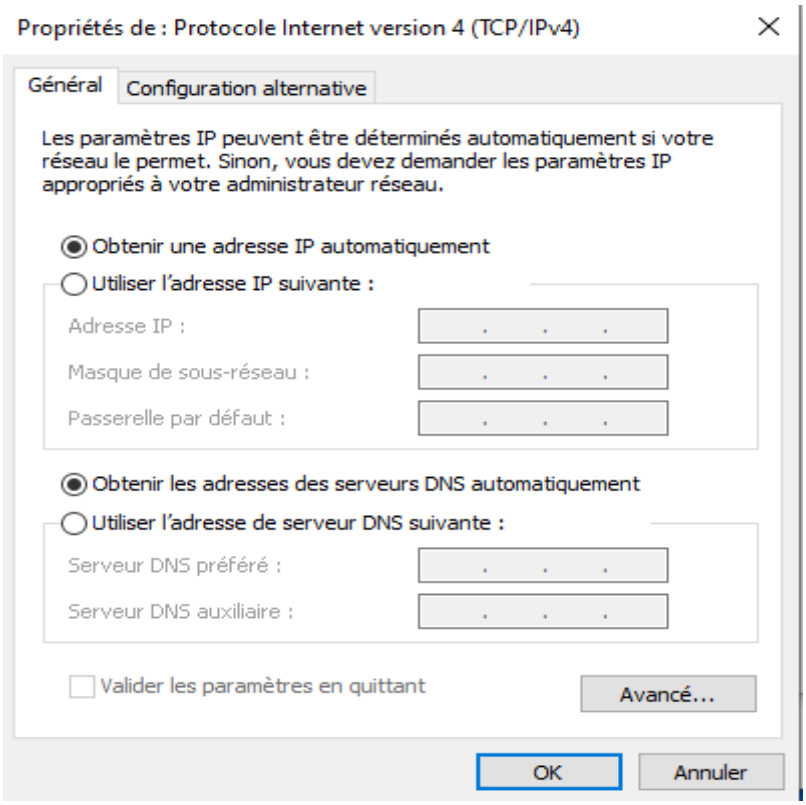
On n'oublie pas de redémarrer notre service isc-dhcp-server avec ;

Systemctl restart isc-dhcp-server

3. Tests Hors accès à internet

Tests sur machine Windows 10 :

On doit configurer la machine pour qu'elle puisse recevoir les indications du serveur DHCP :



On peut voir que nous avons reçu les informations de notre serveur DHCP :

```
C:\Users\user>ipconfig

Configuration IP de Windows


Carte Ethernet Ethernet :

   Suffixe DNS propre à la connexion. . . : eiffel-bdx.td
   Adresse IPv6 de liaison locale. . . . .: fe80::b886:253:d65f:9fd4%8
   Adresse IPv4. . . . . . . . . . . . . .: 10.0.1.50
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . : 10.0.1.24
```

On peut voir aussi que le DNS fonctionne encore :

```
C:\Users\user>nslookup dns.sio.eiffel-bdx.td
Serveur : dns.sio.eiffel-bdx.td
Address:  10.0.1.24

Nom :    dns.sio.eiffel-bdx.td
Address:  10.0.1.24
```

Avec routeur.eiffel-bdx.td :

```
C:\Users\user>nslookup routeur.eiffel-bdx.td
Serveur : dns.sio.eiffel-bdx.td
Address:  10.0.1.24

Nom :    routeur.eiffel-bdx.td
Addresses:  10.0.1.100
          10.0.2.200
```

4. Client avec accès internet

Pour cela nous devons avoir une interface reliée à un accès internet et une avec une IP Fixe, comme nous l'avons configuré :

```
root@dns:/etc/bind# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:74:bf:fe brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.24/24 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe74:bffe/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0f:2a:c5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.24/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 81099sec preferred_lft 81099sec
    inet6 fe80::a00:27ff:fe0f:2ac5/64 scope link
       valid_lft forever preferred_lft forever
```

1. Activation du mode routeur :

Nous devons aller dans le fichier /etc/sysctl.conf et on décommente cette ligne pour laisser passer les paquets vers d'autres interfaces et réseaux

```
 GNU nano 7.2                                                    /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

2. Activation du NAT

Pour faire passer les trames à travers notre routeur nous devons faire ceci sur notre interface qui pointe vers Internet :

```
sudo iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE
```

Nous pouvons voir notre iptables en faisant la commande suivante :

```
sudo iptables -L -t nat
```

```
root@dns:/etc/bind# sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
```

3. Automatisation du montage des règles iptables

Nous allons automatiser nos règles sur l'interface enp0s8, pour cela nous devons enregistrer notre configuration actuelle dans un fichier :

```
sudo iptables-save >> /etc/iptables_rules.save
```

Et dans notre configuration réseau on rajoute cette ligne au niveau de l'interface sur laquelle les trames vont sortir, en l'occurrence notre interface enp0s8 :

post-up iptables-restore < /etc/iptables_rules.save

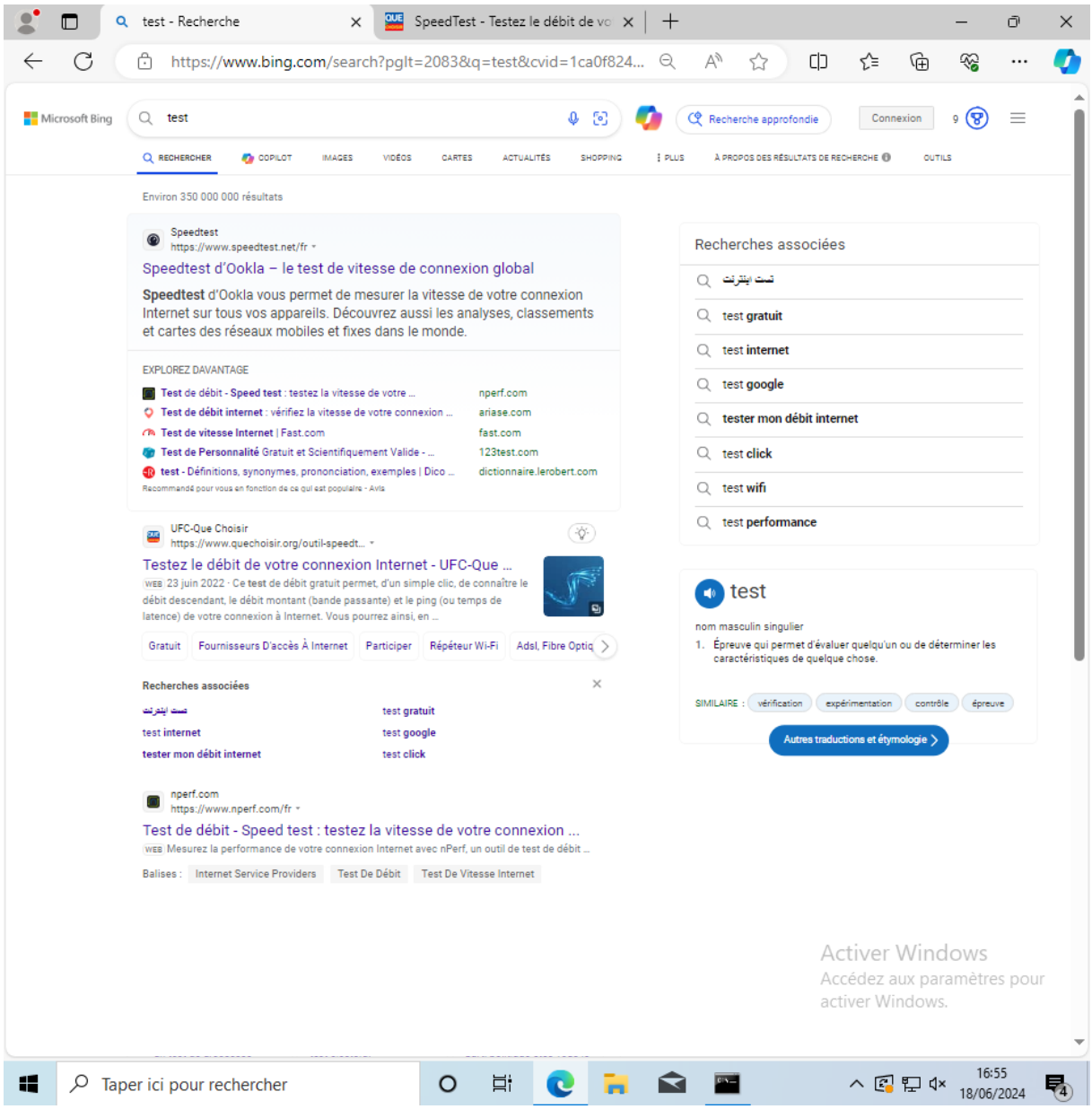
```
allow-hotplug enp0s8
iface enp0s8 inet dhcp
post-up iptables-restore < /etc/iptables_rules.save
```

Avec cette commande nous voyons que notre règle est présente :

```
root@dns:/etc/bind# sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
```

4. Tests sur machine avec accès internet

Nous pouvons voir que nous prenons bien la configuration par le DHCP et que nous avons bien accès à internet :



Et que nous pouvons ping 1.1.1.1

```
C:\Users\user>ping 1.1.1.1

Envoi d'une requête 'Ping'  1.1.1.1 avec 32 octets de données :
Réponse de 1.1.1.1 : octets=32 temps=11 ms TTL=50
Réponse de 1.1.1.1 : octets=32 temps=40 ms TTL=50
Réponse de 1.1.1.1 : octets=32 temps=12 ms TTL=50
Réponse de 1.1.1.1 : octets=32 temps=11 ms TTL=50

Statistiques Ping pour 1.1.1.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 11ms, Maximum = 40ms, Moyenne = 18ms
```