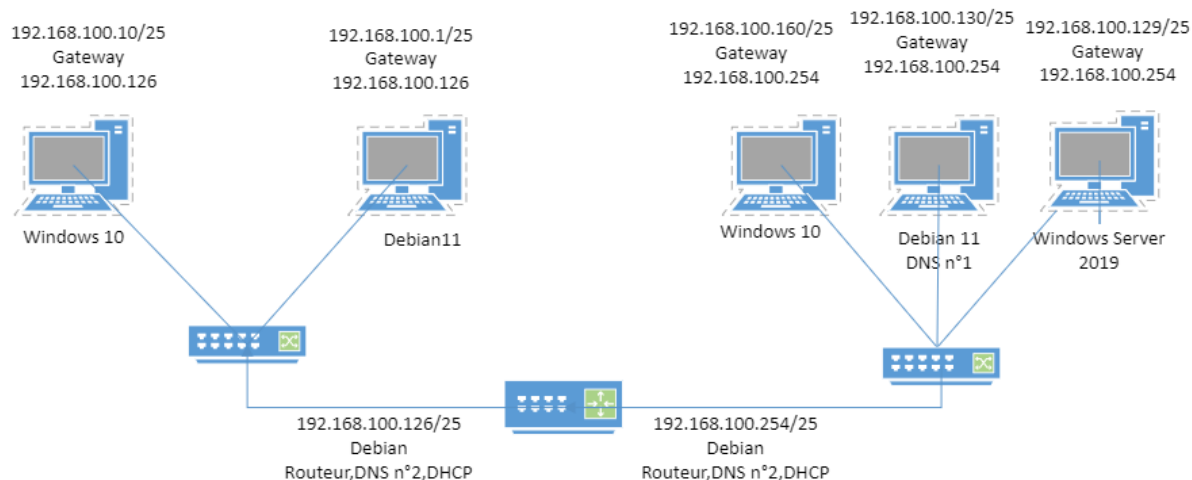


Compte rendu DHCP&DNS

Table des matières

Compte rendu DHCP&DNS	1
Test architecture:	2
Serveur DHCP Linux :	7
Routeur Linux et relais DHCP :	8
Relais DHCP :	11
Routeur :	12
Test routeur, relais DHCP et DHCP sur Debian :	12
Tolérance de panne DHCP :	13
Test tolérance de panne DHCP :	15
Serveur DNS Linux :	17
Installation et configuration :	17
Test DNS :	19
Tolérance de panne serveur DNS :	20
Test tolérance de panne DNS :	21



Test architecture:

Avant de commencer il faut tester la communication entre les différents périphériques du même réseau, on va tout d'abord tester le réseau qui contient les trois machines à partir de la machine Debian 11 :

```
--- 192.168.100.129 ping statistics ---
27 packets transmitted, 27 received, 0% packet loss, time 26042ms
rtt min/avg/max/mdev = 0.777/1.575/4.602/0.848 ms
root@IBsrvli12:/etc/network# ping 192.168.100.160
PING 192.168.100.160 (192.168.100.160) 56(84) bytes of data.
64 bytes from 192.168.100.160: icmp_seq=1 ttl=128 time=9.56 ms
64 bytes from 192.168.100.160: icmp_seq=2 ttl=128 time=0.714 ms
64 bytes from 192.168.100.160: icmp_seq=3 ttl=128 time=0.815 ms
64 bytes from 192.168.100.160: icmp_seq=4 ttl=128 time=0.662 ms
^C
--- 192.168.100.160 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 0.662/2.938/9.564/3.825 ms
root@IBsrvli12:/etc/network# ping 192.168.100.129
PING 192.168.100.129 (192.168.100.129) 56(84) bytes of data.
64 bytes from 192.168.100.129: icmp_seq=1 ttl=128 time=1.50 ms
64 bytes from 192.168.100.129: icmp_seq=2 ttl=128 time=1.06 ms
64 bytes from 192.168.100.129: icmp_seq=3 ttl=128 time=1.25 ms
64 bytes from 192.168.100.129: icmp_seq=4 ttl=128 time=1.50 ms
^C
```

Ensuite on test le réseau sur la machine contenant une image de windows server:

```
PS C:\Users\Administrateur> ping 192.168.100.160

Envoi d'une requête 'Ping' 192.168.100.160 avec 32 octets de données :
Réponse de 192.168.100.160 : octets=32 temps<1ms TTL=128
Réponse de 192.168.100.160 : octets=32 temps=1 ms TTL=128
Réponse de 192.168.100.160 : octets=32 temps<1ms TTL=128
Réponse de 192.168.100.160 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 192.168.100.160:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
PS C:\Users\Administrateur> ping 192.168.100.130

Envoi d'une requête 'Ping' 192.168.100.130 avec 32 octets de données :
Réponse de 192.168.100.130 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.100.130 : octets=32 temps<1ms TTL=64
Réponse de 192.168.100.130 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.100.130 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.100.130:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```

Ensuite on test avec le client windows 10 :

```
PS C:\WINDOWS\system32> ping 192.168.100.129

Envoi d'une requête 'Ping' 192.168.100.129 avec 32 octets de données :
Réponse de 192.168.100.129 : octets=32 temps<1ms TTL=128
Réponse de 192.168.100.129 : octets=32 temps<1ms TTL=128
Réponse de 192.168.100.129 : octets=32 temps<1ms TTL=128
Réponse de 192.168.100.129 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 192.168.100.129:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
PS C:\WINDOWS\system32> ping 192.168.100.130

Envoi d'une requête 'Ping' 192.168.100.130 avec 32 octets de données :
Réponse de 192.168.100.130 : octets=32 temps<1ms TTL=64
Réponse de 192.168.100.130 : octets=32 temps<1ms TTL=64
Réponse de 192.168.100.130 : octets=32 temps<1ms TTL=64
Réponse de 192.168.100.130 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.100.130:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
PS C:\WINDOWS\system32>
```

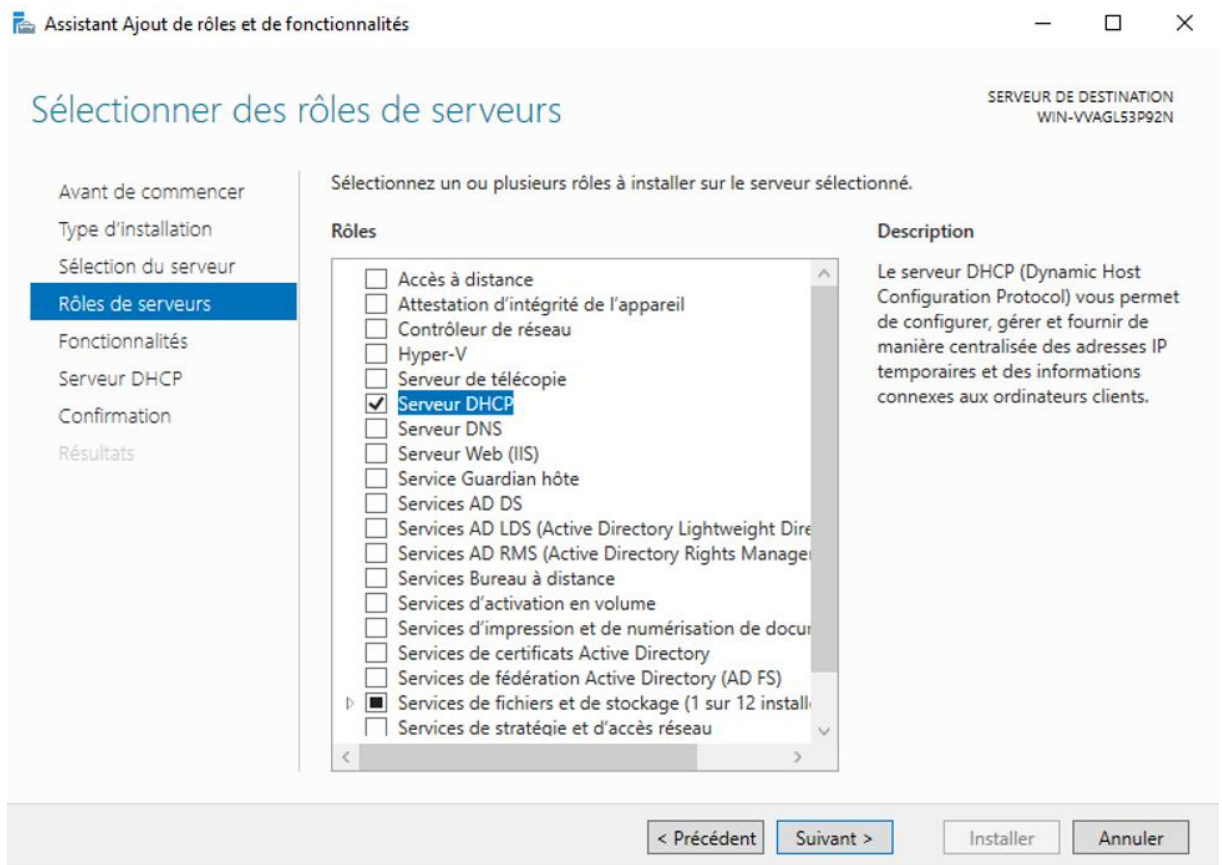
Le réseau est donc complètement opérationnel, a noté que les deux réseaux sont sur deux machines physiques différentes sur Virtual box toute configuré en réseau interne. (A noter que pour ping la machine Windows Server il faut autoriser les pings venant de l'extérieur dans le pare-feu)

Serveur DHCP Windows :

Voici les étapes pour installer le service DHCP sur un serveur Windows :

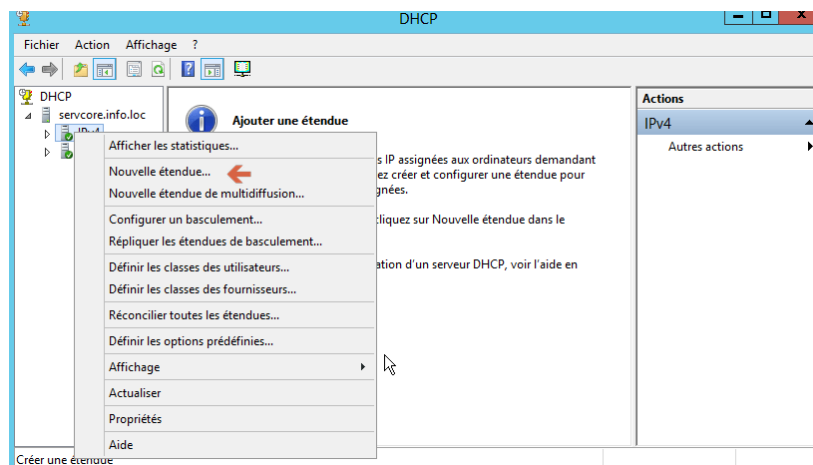
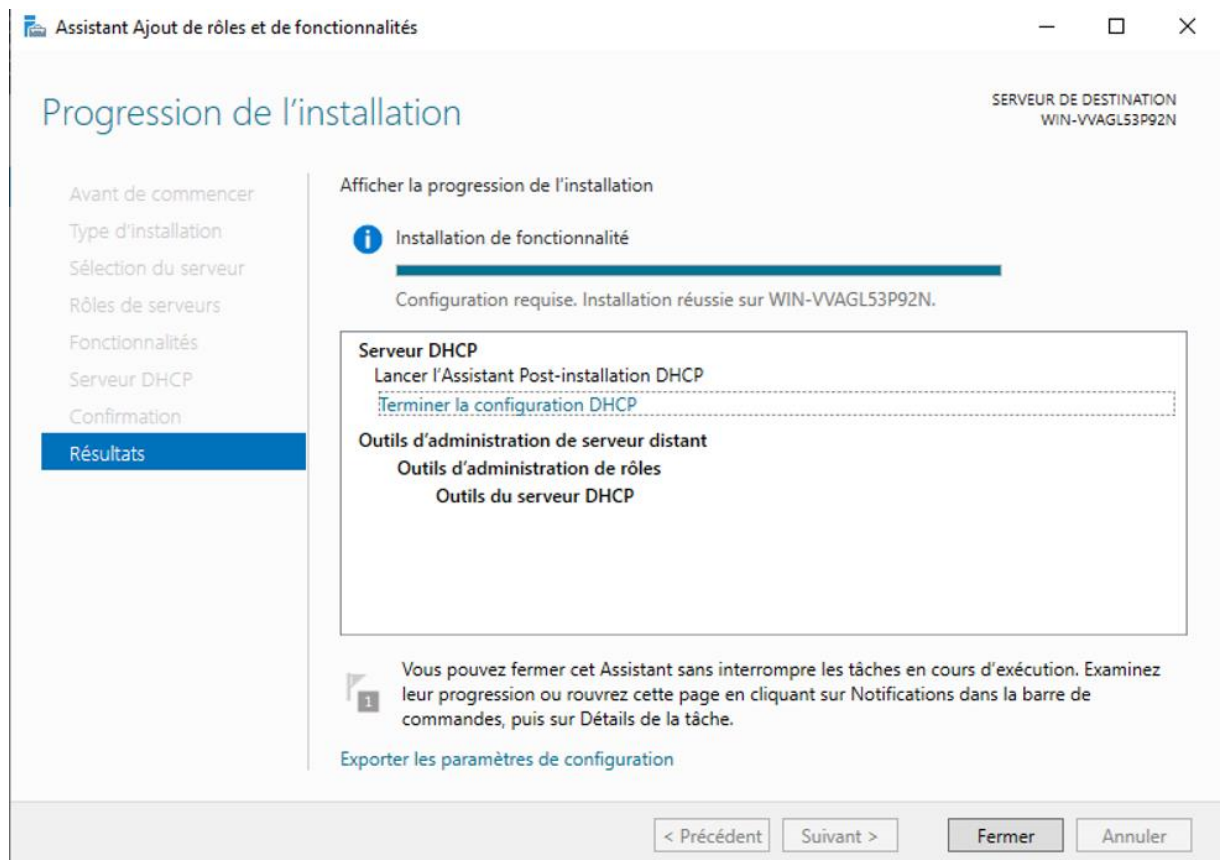
1. **Ouvrir le gestionnaire de serveur** : Vous pouvez le trouver dans le menu "Outils d'administration" ou en cliquant sur l'icône du gestionnaire de serveur dans la barre des tâches.

2. **Sélectionner "Ajouter des rôles et fonctionnalités"** : Dans le gestionnaire de serveur, cliquez sur "Ajouter des rôles et fonctionnalités" dans le volet de droite.
3. **Assistant "Ajouter des rôles et fonctionnalités"** : Cliquez sur "Suivant" pour commencer l'assistant.
4. **Sélectionner le type d'installation** : Choisissez "Installation basée sur un rôle ou une fonctionnalité" et cliquez sur "Suivant".
5. **Sélectionner le serveur** : Choisissez le serveur sur lequel vous souhaitez installer le rôle DHCP et cliquez sur "Suivant".



6. **Sélectionner le rôle DHCP** : Dans la liste des rôles, cochez la case "Serveur DHCP" et cliquez sur "Suivant".
7. **Confirmer l'installation des fonctionnalités supplémentaires** : Si des fonctionnalités supplémentaires sont nécessaires pour DHCP, cliquez sur "Ajouter des fonctionnalités".
8. **Résumé de l'installation** : Examinez les informations récapitulatives et cliquez sur "Installer" pour démarrer le processus d'installation.
9. **Installation en cours** : Attendez que l'installation soit terminée. Vous pouvez suivre la progression de l'installation sur la page.

10. Installation réussie : Une fois l'installation terminée, cliquez sur "Fermer" pour quitter l'assistant.



Maintenant il faut définir les étendues, on ne va pas inclure dans l'étendue les adresses de début pour les réserver à d'autre potentielle serveur, pour ne pas inclure l'adresse de réseau et pour ne pas inclure l'adresse du serveur DHCP ! Ne pas les inclure dans l'étendue vas nous éviter de devoir faire des exclusions et donc nous faire gagner du temps. Dans l'étendue on ne prend pas aussi les adresses fin pour les réserver à des

potentielles routeurs et pour ne pas inclure l'adresse de broadcast.

Assistant Nouvelle étendue

Plage d'adresses IP

Vous définissez la plage d'adresses en identifiant un jeu d'adresses IP consécutives.



Paramètres de configuration pour serveur DHCP

Entrez la plage d'adresses que l'étendue peut distribuer.

Adresse IP de début :

Adresse IP de fin :

Paramètres de configuration qui se propagent au client DHCP.

Longueur :

Masque de sous-réseau :

< Précédent **Suivant >** Annuler

Ensuite on définit la durée du bail :

Assistant Nouvelle étendue

Durée du bail

La durée du bail spécifie la durée pendant laquelle un client peut utiliser une adresse IP de cette étendue.



La durée du bail doit théoriquement être égale au temps moyen durant lequel l'ordinateur est connecté au même réseau physique. Pour les réseaux mobiles constitués essentiellement par des ordinateurs portables ou des clients d'accès à distance, des durées de bail plus courtes peuvent être utiles.

De la même manière, pour les réseaux stables qui sont constitués principalement d'ordinateurs de bureau ayant des emplacements fixes, des durées de bail plus longues sont plus appropriées.

Définissez la durée des baux d'étendue lorsqu'ils sont distribués par ce serveur.

Limitée à :

Jours : Heures : Minutes :

< Précédent **Suivant >** Annuler

C'est bon le serveur DHCP sur windows Server est fonctionnel , on vas le tester sur la machine Debian pour voir toutes les étapes :

```
root@IBsrvli12:/etc/network# ifup enp0s3
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s3/08:00:27:ae:72:48
Sending on   LPF/enp0s3/08:00:27:ae:72:48
Sending on   Socket/fallback
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 11
DHCPOFFER of 192.168.100.141 from 192.168.100.129
DHCPREQUEST for 192.168.100.141 on enp0s3 to 255.255.255.255 port 67
DHCPACK of 192.168.100.141 from 192.168.100.129
bound to 192.168.100.141 -- renewal in 34023 seconds.
root@IBsrvli12:/etc/network#
```

On voit ici que le serveur DHCP a rempli sa fonction et a attribuer une adresse IP et le masque de sous réseau a la machine Debian. On peut aussi regarder directement dans les enregistrement du serveur DHCP :

Adresse IP du client	Nom	Expiration du bail	Type	ID unique
 192.168.100.140	Win10A	07/03/2024 11:59:33	DHCP	080027a09...
 192.168.100.141	IBsrvli12	07/03/2024 11:49:30	DHCP	27ae72480...

Ici on voit bien que les deux machines dans le réseau du serveur DHCP on bien reçu une adresse IP et un masque de la part du serveur DHCP.

Et dans regardant les trames sur WireShark on voit tout le processus DORA = Discover Offer Request Acknowledge :

54	47.257970	192.168.100.129	255.255.255.255	DHCP	342 DHCP Offer	- Transaction ID 0xcce742a	
55	47.258250	0.0.0.0	255.255.255.255	DHCP	351 DHCP Request	- Transaction ID 0xcce742a	
56	47.260622	192.168.100.129	255.255.255.255	DHCP	342 DHCP ACK	- Transaction ID 0xcce742a	
57	47.295652	192.168.100.140	224.0.0.22	IGMPv3	54 Membership Report / Join group	224.0.0.251 for any	
58	47.309982	192.168.100.140	224.0.0.22	IGMPv3	54 Membership Report / Join group	224.0.0.252 for any	

> Frame 54: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{45B8F8B4-FD09-48F1-8F96-7C6A29AC}

> Ethernet II, Src: PcsCompu_88:21:5d (08:00:27:88:21:5d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4, Src: 192.168.100.129, Dst: 255.255.255.255

> User Datagram Protocol, Src Port: 67, Dst Port: 68

> Dynamic Host Configuration Protocol (Offer)

Serveur DHCP Linux :

Installation de DHCP :

D'abord, pour avoir un serveur DHCP, il faut installer le service !

“apt **install** isc-dhcp-server”

Sur Debian, il y a une petite spécificité, il faut indiquer dans **/etc/default/isc-dhcp-server** sur quelles interfaces va écouter le service DHCP.

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

On décommente la troisième ligne et on indique l'interface.

```
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="enp0s3"
INTERFACESv6=""
```

Ensuite on modifie le fichier de configuration pour déclarer le réseau et y mettre l'étendue, on peut aussi définir le bail.

```
root@IBsrv112:/etc/dhcp# nano dhcpd.conf_
```

Dans la capture ci-dessous on définit dans la première ligne le réseau ou le serveur DHCP va attribuer les adresses IP et la plage d'adresse est défini dans range on met exactement la même étendue que dans le serveur DHCP sur Windows server qui a bien évidemment été désactiver.

```
# This is a very basic subnet declaration.

subnet 192.168.100.128 netmask 255.255.255.128 {
    range 192.168.100.140 192.168.100.220;
    # option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}
```

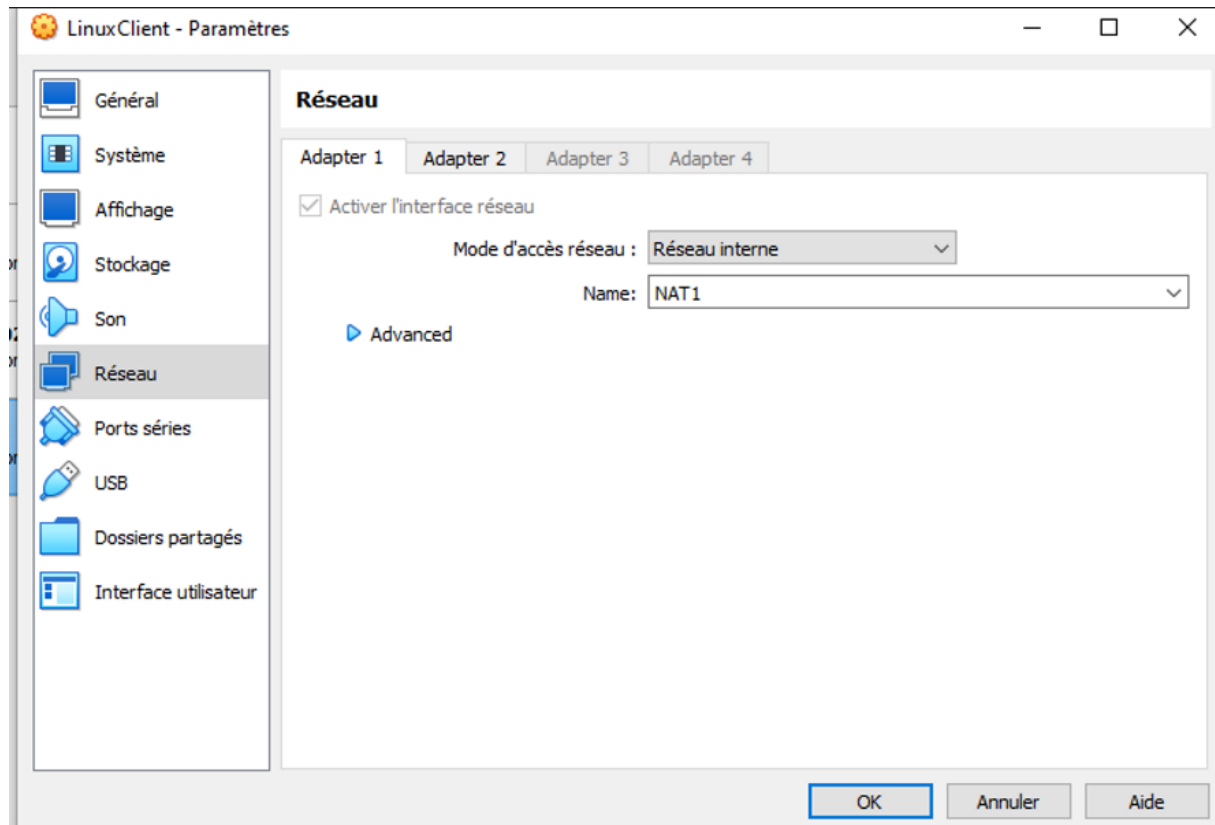
Et on définit une autre étendue pour le réseau distant .

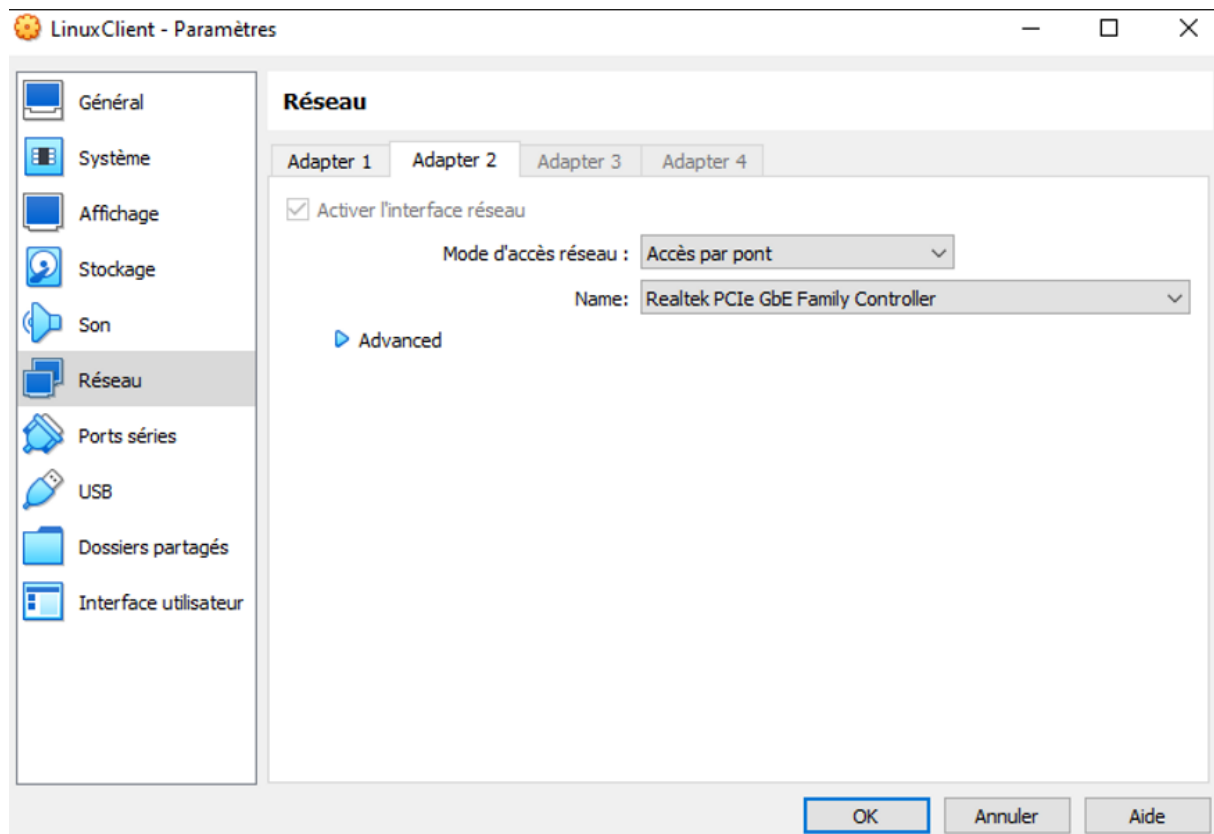
```
subnet 192.168.100.0 netmask 255.255.255.128 {
    range 192.168.100.10 192.168.100.115;_
}
```

Routeur Linux et relais DHCP :

Nous avons notre serveur DHCP qui peut attribuer des adresses IP sur son réseau mais nous voulons qu'il attribue des adresses IP sur un autre réseau , et pour faire cela il nous faut un relais DHCP et on vas utiliser le routeur.

Tout d'abord pour faire en sorte que notre linux Debian soit un routeur il faut sur virtual Box utilisés deux cartes réseaux physiques une en accès interne pour le réseau local , la deuxième carte en mode pont et directement branché a la machine physique contenant les deux machines virtuel debian et windows. C'est écrit LinuxClient mais c'est bien le routeur.





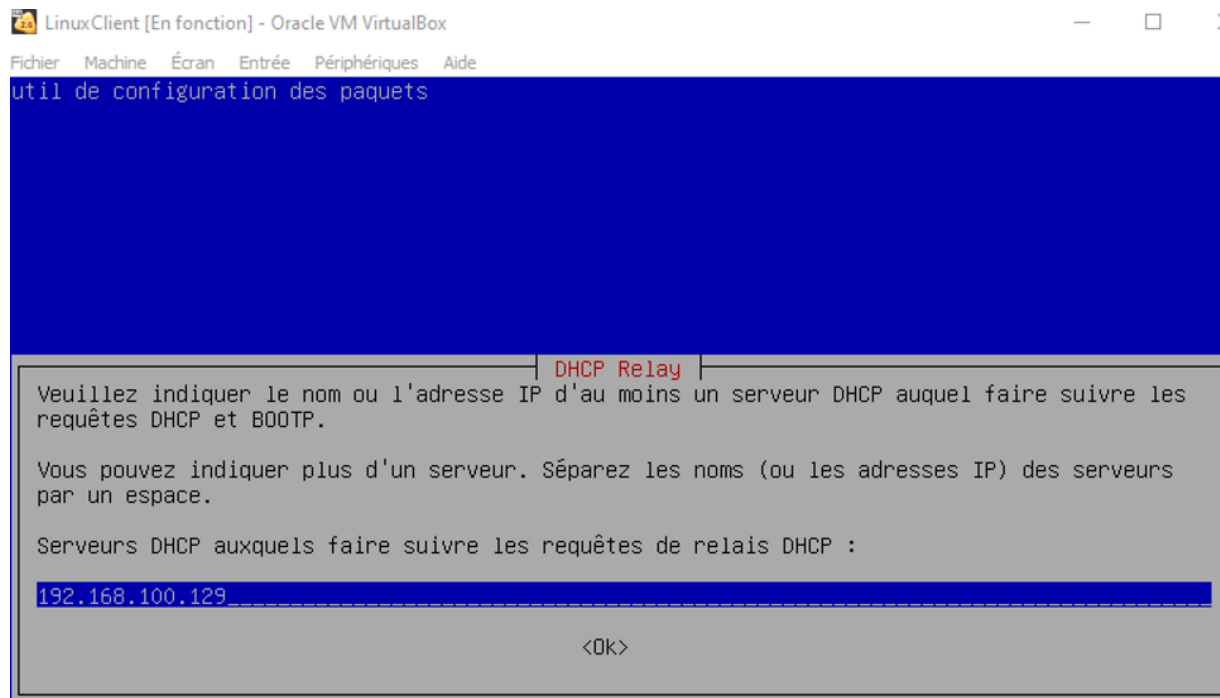
Le windows 10 qui sera le client est configuré sur Virtual box en mode accès par pont pour pouvoir communiquer avec le routeur et relais DHCP.

Relais DHCP :

Il faut tout d'abord installer le service qui permet de faire relais DHCP,

« **Apt install isc-dhcp-relay** »

Pendant l'installation une interface graphique va apparaître et il faut préciser l'adresse IP du serveur DHCP :



Ensuite il faut retourner sur le serveur dhcp pour rajouter le routeur et son adresse de diffusion dans l'étendue :

```
subnet 192.168.100.0 netmask 255.255.255.128 {  
    range 192.168.100.10 192.168.100.115;  
    option routers 192.168.100.254;  
    option broadcast-address 192.168.100.255;  
}
```

Routeur :

Pour configurer la machine debian en mode routeur il suffit de decommenter cette ligne dans le fichier de conf /etc/sysctl.conf :

« nano /etc/sysctl.conf »

```
GNU nano 5.4 /etc/sysctl.conf *
#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

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
```

Comme nous avons deux cartes réseaux il faut en configurer une pour chaque réseaux

```
GNU nano 5.4 /etc/network/interfaces *
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
#source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.100.126
netmask 255.255.255.128

# The secondary network interface
allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.100.254
netmask 255.255.255.128
```

Test routeur, relais DHCP et DHCP sur Debian :

Tout d'abord on test le serveur DHCP sur un client Debian sur le même réseau :

```
root@DebianClient:~# ifup ens18
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens18/e6:40:c4:48:57:f6
Sending on   LPF/ens18/e6:40:c4:48:57:f6
Sending on   Socket/fallback
DHCPDISCOVER on ens18 to 255.255.255.255 port 67 interval 8
DHCPOFFER of 192.168.100.140 from 192.168.100.130
DHCPREQUEST for 192.168.100.140 on ens18 to 255.255.255.255 port 67
DHCPACK of 192.168.100.140 from 192.168.100.130
bound to 192.168.100.140 -- renewal in 287 seconds.
root@DebianClient:~# _
```

On voit ici que le serveur DHCP sur l'adresse 192.168.100.130 donne une adresse au client en 192.168.100.140 qui est la première adresse de l'étendue. Ensuite nous allons tester si le routeur et le relais DHCP fonctionne en testant avec un client qui n'est pas dans le même réseau que le serveur DHCP :

```
root@DebianClient:~# ifup ens18
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens18/e6:40:c4:48:57:f6
Sending on   LPF/ens18/e6:40:c4:48:57:f6
Sending on   Socket/fallback
DHCPDISCOVER on ens18 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on ens18 to 255.255.255.255 port 67 interval 18
DHCPOFFER of 192.168.100.10 from 192.168.100.126
DHCPREQUEST for 192.168.100.10 on ens18 to 255.255.255.255 port 67
DHCPACK of 192.168.100.10 from 192.168.100.126
bound to 192.168.100.10 -- renewal in 275 seconds.
root@DebianClient:~# _
```

On voit ici que le client reçoit une réponse du routeur (192.168.100.126) donc du relais DHCP qui lui donne comme adresse 192.168.100.10 qui est bien l'adresse défini dans la deuxième étendue du serveur DHCP.

Tolérance de panne DHCP :

Pour effectuer de la tolérance de panne il suffit de décommenter cette ligne dans le fichier de configuration du serveur DHCP maître et rajouter les lignes si-dessous :

```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

Paramétrage du failover du DHCP Master

```
failoverpeer "test" {  
    primary ; # Déclare ce serveur comme master.  
    address 192.168.100.130 ; # Adresse du serveur master.  
    port 520 ; # Port d'écoute du serveur master.  
    peeraddress 192.168.100.131 ; # Adresse du serveur slave.  
    peer port 520 ; # Port d'écoute du serveur slave.  
    max-response-delay 60 ; # Temps de non réponse du slave.  
    max-unacked-updates 10 ;  
    mclt 3600 ;  
    split 128 ; # Répartition des plages d'adresses.  
    load balance max seconds 3 ;  
}
```

```
network, the authoritative direct  
authoritative;  
failover peer "test" {  
    primary;  
    address 192.168.100.130;  
    port 520;  
    peer address 192.168.100.131;  
    peer port 520;  
    max-response-delay 60;  
    max-unacked-updates 10;  
    mclt 3600;  
    split 128;  
    load balance max seconds 3;  
}
```

```
subnet 192.168.100.128 netmask 255.255.255.128 {  
    pool {  
        failover peer "test";  
        range 192.168.100.140 192.168.100.220;  
    }  
}
```

La directive « SPLIT » est également présente dans la configuration du service DHCP en mode « **failover** » tout comme la directive MCLT. Elle permet de « **splitter** » c'est-à-dire de diviser la plage d'adresses IP disponible en deux parties, afin de répartir la charge sur les deux serveurs.

Lorsque cette directive a la valeur « **128** », on part sur une répartition **50%/50%**, c'est-à-dire que chaque serveur gère 50% de la plage d'adresses disponible.

MCLT = Cette directive qui signifie « **Max Client Lead Time** » et qui est présente lors de la configuration du service DHCP en mode « **failover** » correspond au temps maximum, pendant lequel le serveur peer peut renouveler des requêtes après avoir perdu contact avec son partenaire.

Paramétrage du failover du DHCP Slave

```
failoverpeer "test" {
```

```

secondary      ; # Déclare ce serveur comme slave.
address 192.168.100.131 ; # Adresse du serveur slave.
port 520       ; # Port d'écoute du serveur slave.
peeraddress 192.168.100.130 ; # Adresse du serveur master.
peer port 520   ; # Port d'écoute du serveur master.
max-response-delay 60 ; # Temps de non réponse en secondes.
max-unacked-updates 10 ; # Nombre de mises à jour avant de déclarer le pair en
échec
load balance max seconds 3; # Durée max avant de décharger la requête vers le pair
}

```

```

authoritative;

failover peer "test"{
secondary;
address 192.168.100.131;
port 520;
peer address 192.168.100.130;
peer port 520;
max-response-delay 60;
max-unacked-updates 10;
load balance max seconds 3;
}

subnet 192.168.100.128 netmask 255.255.255.128 {
    pool{
        failover peer "test";
        range 192.168.100.140 192.168.100.220;
    }
}

# option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;

```

A noter qu'il est important que les deux serveurs est exactement la même date, heure et seconde , pour configurer cela on peut faire la commande date ou utiliser le protocole NTP .

Test tolérance de panne DHCP :

Nous allons tester notre tolérance de panne et surtout regarder si les deux serveurs sont bien lié pour regarder ceci il suffit de consulter les derniers logs et pour faire cela on

peut utiliser la commande « **systemctl status isc-dhcp-server** » :

```
root@ServeurDhcp:~# systemctl status isc-dhcp-server
• isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Wed 2024-04-17 21:09:32 CEST; 7min left
     Docs: man:systemd-sysv-generator(8)
  Process: 1242 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 2307)
   Memory: 3.9M
      CPU: 36ms
  CGroup: /system.slice/isc-dhcp-server.service
          └─1254 /usr/sbin/dhcdp -4 -q -cf /etc/dhcp/dhcpd.conf ens18

avril 17 21:10:53 ServeurDhcp dhcpd[1254]: Failover CONNECT to test rejected: Connection rejected, time mismatch too great.
avril 17 21:10:58 ServeurDhcp dhcpd[1254]: Failover CONNECT to test rejected: Connection rejected, time mismatch too great.
avril 17 21:10:58 ServeurDhcp dhcpd[1254]: Failover CONNECT to test rejected: Connection rejected, time mismatch too great.
avril 17 21:02:00 ServeurDhcp dhcpd[1254]: Failover CONNECT to test rejected: Connection rejected, time mismatch too great.
avril 17 21:02:05 ServeurDhcp dhcpd[1254]: failover peer test: peer moves from normal to communications-interrupted
avril 17 21:02:05 ServeurDhcp dhcpd[1254]: failover peer test: I move from communications-interrupted to normal
avril 17 21:02:05 ServeurDhcp dhcpd[1254]: balancing pool 561463264dc0 192.168.100.128/25 total 81 free 40 backup 40 lts 0 max-own (+/-)8
avril 17 21:02:05 ServeurDhcp dhcpd[1254]: balanced pool 561463264dc0 192.168.100.128/25 total 81 free 40 backup 40 lts 0 max-misbal 12
avril 17 21:02:05 ServeurDhcp dhcpd[1254]: failover peer test: peer moves from communications-interrupted to normal
avril 17 21:02:05 ServeurDhcp dhcpd[1254]: failover peer test: Both servers normal
root@ServeurDhcp:~#
```

On voit au début que la liaison a échoué et qu'ensuite cela a marcher leur mode de communication est passer de normal a interrompu et ensuite il est repassé de interrompu a normal, le load balancing marche et a la fin on nous précise que les deux serveurs fonctionne normalement les logs du serveur esclave sont similaires :

```
root@ServeurDhcpSlave:~# systemctl status isc-dhcp-server
• isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Wed 2024-04-17 21:17:55 CEST; 15min left
     Docs: man:systemd-sysv-generator(8)
  Process: 377 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 2307)
   Memory: 7.3M
      CPU: 47ms
  CGroup: /system.slice/isc-dhcp-server.service
          └─392 /usr/sbin/dhcdp -4 -q -cf /etc/dhcp/dhcpd.conf ens18

avril 17 21:18:41 ServeurDhcpSlave dhcpd[392]: Failover CONNECT from test: time offset too large
avril 17 21:18:41 ServeurDhcpSlave dhcpd[392]: failover: disconnect: time offset too large
avril 17 21:18:46 ServeurDhcpSlave dhcpd[392]: Failover CONNECT from test: time offset too large
avril 17 21:18:46 ServeurDhcpSlave dhcpd[392]: failover: disconnect: time offset too large
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: failover peer test: peer moves from normal to communications-interrupted
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: failover peer test: I move from communications-interrupted to normal
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: balancing pool 560cd58e0cd0 192.168.100.128/25 total 81 free 40 backup 40 lts 0 max-own (+/-)8
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: balanced pool 560cd58e0cd0 192.168.100.128/25 total 81 free 40 backup 40 lts 0 max-misbal 12
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: failover peer test: peer moves from communications-interrupted to normal
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: failover peer test: Both servers normal
root@ServeurDhcpSlave:~#
```

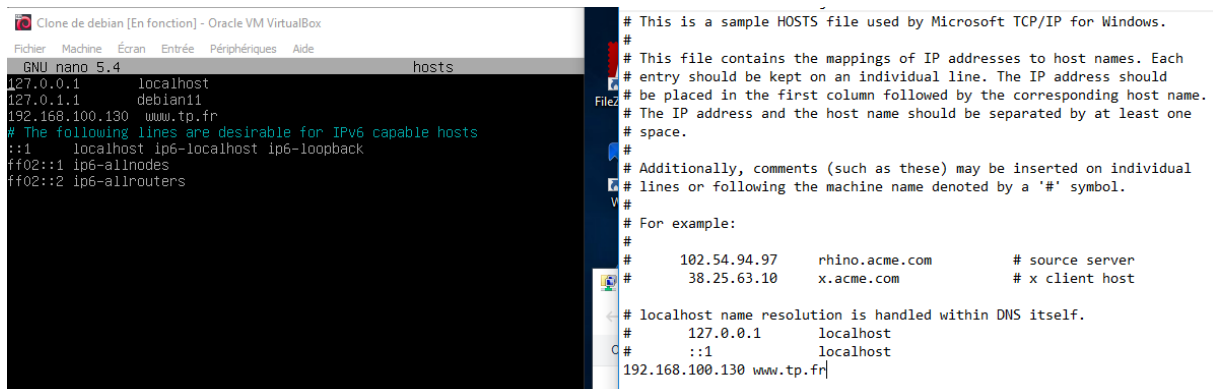
Maintenant arrêtons le serveur DHCP principale (« **systemctl stop isc-dhcp-server** ») et regardons ce qu'il se passe :

```
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: balanced pool 560cd58e0cd0 192.168.100.128/25 total 81 free 40 backup 40 lts 0 max-own (+/-)8
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: failover peer test: peer moves from communications-interrupted to normal
avril 17 21:02:04 ServeurDhcpSlave dhcpd[392]: failover peer test: Both servers normal
avril 17 21:03:10 ServeurDhcpSlave dhcpd[392]: peer test: disconnected
avril 17 21:03:10 ServeurDhcpSlave dhcpd[392]: failover peer test: I move from normal to communications-interrupted
root@ServeurDhcpSlave:~#
```

On voit sur les logs du serveur esclave que les communications entre les serveurs est passé de normal a interrompu, le serveur esclave attribue donc maintenant les adresses IP aux machines sur le même réseau.

Serveur DNS Linux :

Nous allons mettre en place un serveur DNS linux pour faire de la résolution de nom, on peut d'ailleurs faire de la résolution de nom dans le fichier hosts comme ceci :



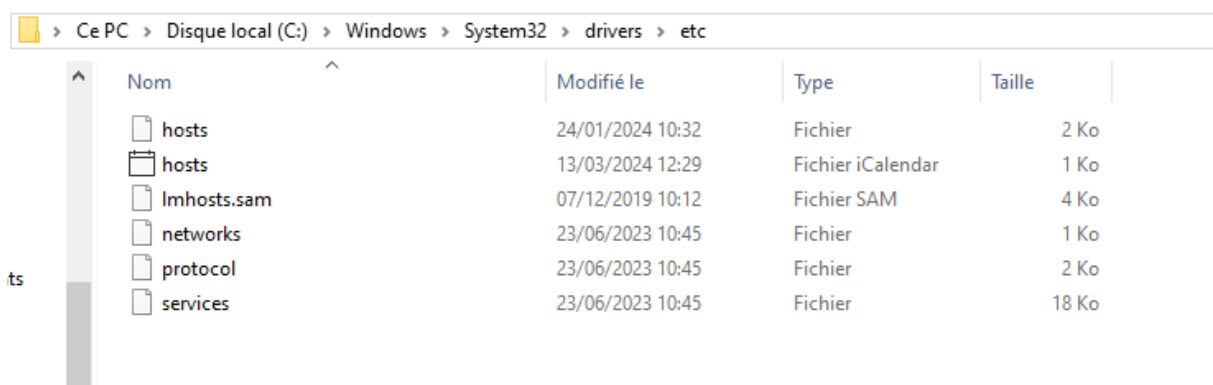
The image shows two side-by-side screenshots. The left screenshot is a terminal window titled 'Clone de debian [En fonction] - Oracle VM VirtualBox' showing the contents of the /etc/hosts file in nano 5.4. The right screenshot is a Windows File Explorer window showing the contents of the C:\Windows\System32\drivers\etc\hosts file.

```
127.0.0.1 localhost
127.0.1.1 debian11
192.168.100.130 www.tp.fr
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
# 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
192.168.100.130 www.tp.fr
```

Comme on peut le voir la syntaxe est la même on donne une adresse IP qui correspond à un nom.

A gauche le fichier est situé dans /etc/hosts et sur windows :



The image shows a Windows File Explorer window with the address bar set to 'Ce PC > Disque local (C:) > Windows > System32 > drivers > etc'. The main pane displays a list of files and folders.

Nom	Modifié le	Type	Taille
hosts	24/01/2024 10:32	Fichier	2 Ko
hosts	13/03/2024 12:29	Fichier iCalendar	1 Ko
lmhosts.sam	07/12/2019 10:12	Fichier SAM	4 Ko
networks	23/06/2023 10:45	Fichier	1 Ko
protocol	23/06/2023 10:45	Fichier	2 Ko
services	23/06/2023 10:45	Fichier	18 Ko

Et c'est aussi le fichier hosts.

Pour la prochaine etape nous n'avons pas besoin de faire cela nous allons utilisé le serveur DNS.

Installation et configuration :

Tout d'abord pour installer un serveur DNS sur Debian nous allons installer bind9 en faisant cette commande « **apt install bind9** ». Une fois installé nous devons indiqué a la machine ou est le serveur DNS et pour faire ceci nous devons modifier le fichier

resolv.conf en mettant notre serveur dns et son adresse IP :

```
GNU nano 5.4                                resolv.conf
domain test.fr
search test.fr
nameserver 192.168.100.130
```

Une fois ceci fait on peut commencer a configurer le serveur DNS , pour faire cela nous allons aller dans le fichier de configuration named.conf.default-zones situé dans « **/etc/bind/named.conf.default-zones** ». Dans ce fichier nous allons rajouter une zone de recherche direct et indirect :

Zone recherche direct = un nom égale une adresse IP

Zone de recherche indirect = une adresse IP égale un nom

```
zone "test.fr"{
    type master ;
    file "/etc/bind/test.zoned";
};
zone "100.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/test.zonei";
};
```

PS = ne pas oublier le point-virgule après l'accolade !

Nous allons maintenant créer les deux fichiers, pour simplifier la chose nous allons copier un fichier de configuration déjà existant :

```
root@IBsrvli12:/etc/bind# cp db.0 test.zoned
```

Et maintenant voici la configuration de la zone de recherche direct :

```
GNU nano 7.2                                test.zoned *
;
; BIND reverse data file for "this host on this network" zone
;
$TTL      604800
@         IN      SOA      IBsrvli12.tp.fr. root.tp.fr. (
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       IBsrvli12.tp.fr.
@ IN A 192.168.100.130
IBsrvli12 IN A 192.168.100.130
```

La deuxième ligne en résumé, spécifie que le serveur principal (maître) pour la zone "tp.fr" est "IBsrvli12.tp.fr." et que l'adresse e-mail du responsable de la zone est "root.tp.fr.". C'est un enregistrement SOA qui établit l'autorité sur la zone. IBsrvli12 correspond au nom de la machine ou est situé le serveur DNS.

Ensuite la troisième ligne en partant de la fin permet d'indiquer un enregistrement NS :

NS : C'est le type d'enregistrement, qui signifie "Name Server".

IBsrvli12.tp.fr. : C'est le nom du serveur de noms (Name Server) pour cette zone. Dans cet exemple, le serveur de noms pour la zone "tp.fr" est "IBsrvli12.tp.fr."

Les deux autres lignes permettent de faire un enregistrement A qui associe un nom de domaine a une adresse IPV4.

Configurons maintenant la zone indirecte nous allons encore copié un fichier de configuration déjà établi pour configurer notre zone :

```
root@IBsrvli12:/etc/bind# cp db.0 test.zonei
```

Maintenant voici la configuration de la zone indirect :

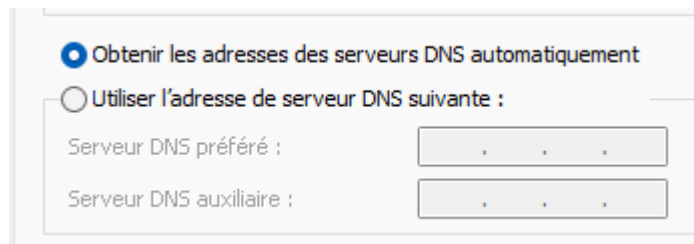
```
GNU nano 7.2 test.zonei *
;
; BIND reverse data file for "this host on this network" zone
;
$TTL      604800
@          IN      SOA      IBsrvli12.test.fr. root.test.fr. (
                        1      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@          IN      NS       IBsrvli12.test.fr.
130 IN PTR IBsrvli12.test.fr_
```

Ici on met uniquement l'enregistrement PTR (pointeur) qui vas permettre d'associer un nom de domaine a une adresse IP.

Test DNS :

Nous allons maintenant tester notre serveur DNS en utilisant nslookup :

Ne pas oublier que pour le faire sur windows il faut mettre l'adresse IP du serveur DNS créer

The image shows a Windows Network Settings window. It has two radio buttons: "Obtenir les adresses des serveurs DNS automatiquement" (selected) and "Utiliser l'adresse de serveur DNS suivante :". Below the second option, there are two input fields: "Serveur DNS préféré :" and "Serveur DNS auxiliaire :". Both fields contain three dots, indicating they are empty.

ici :

```
PS C:\WINDOWS\system32> nslookup 192.168.100.130
Serveur : IBsrvli12.tp.fr
Address: 192.168.100.130
```

```
Nom : IBsrvli12.tp.fr
Address: 192.168.100.130
```

```
PS C:\WINDOWS\system32> nslookup IBsrvli12.tp.fr
Serveur : IBsrvli12.tp.fr
Address: 192.168.100.130
```

```
Nom : IBsrvli12.tp.fr
Address: 192.168.100.130
```

```
root@IBsrvli12:/etc/bind# nslookup IBsrvli12
Server: 192.168.100.130
Address: 192.168.100.130#53

Name: IBsrvli12.tp.fr
Address: 192.168.100.130
```

```
root@IBsrvli12:/etc/bind# nslookup 192.168.100.130
130.100.168.192.in-addr.arpa name = IBsrvli12.tp.fr.
```

On voit sur ces captures que les deux recherche direct et inversé sont opérationnelles.

On peut aussi le voir grâce à WireShark :

1	0.000000	192.168.100.150	192.168.100.130	DNS	88 Standard query 0x0001 PTR 130.100.168.192.in-addr.arpa
2	0.001000	192.168.100.130	192.168.100.150	DNS	117 Standard query response 0x0001 PTR 130.100.168.192.in-addr.arpa PTR IBsrvli12.tp.fr
3	0.001367	192.168.100.150	192.168.100.130	DNS	88 Standard query 0x0002 PTR 130.100.168.192.in-addr.arpa
4	0.002245	192.168.100.130	192.168.100.150	DNS	117 Standard query response 0x0002 PTR 130.100.168.192.in-addr.arpa PTR IBsrvli12.tp.fr
9	26.050243	192.168.100.150	192.168.100.130	DNS	88 Standard query 0x0001 PTR 130.100.168.192.in-addr.arpa
10	26.051566	192.168.100.130	192.168.100.150	DNS	117 Standard query response 0x0001 PTR 130.100.168.192.in-addr.arpa PTR IBsrvli12.tp.fr
11	26.051988	192.168.100.150	192.168.100.130	DNS	75 Standard query 0x0002 A IBsrvli12.tp.fr
12	26.052673	192.168.100.130	192.168.100.150	DNS	91 Standard query response 0x0002 A IBsrvli12.tp.fr A 192.168.100.130
13	26.053162	192.168.100.150	192.168.100.130	DNS	75 Standard query 0x0003 AAAA IBsrvli12.tp.fr
14	26.053864	192.168.100.130	192.168.100.150	DNS	116 Standard query response 0x0003 AAAA IBsrvli12.tp.fr SOA IBsrvli12.tp.fr

```
> Internet Protocol Version 4, Src: 192.168.100.130, Dst: 192.168.100.150
> User Datagram Protocol, Src Port: 53, Dst Port: 50602
```

Ici on voit les requêtes DNS et les réponses DNS.

Tolérance de panne serveur DNS :

Nous allons mettre en place un deuxième serveur DNS au cas ou le premier soit hors service. Pour faire cela il faut tout d'abord sur le premier serveur DNS (Serveur maitre) aller dans le fichier `named.conf.default-zones` situé dans « `/etc/bind/named.conf.default-zones` » comme vu précédemment et mettre ceci :

```
zone "test.fr"{
    type master ;
    file "/etc/bind/test.zoned";
    allow-transfer { 192.168.100.131; };
    also-notify { 192.168.100.131; };
};
```

Ici on précise l'adresse du serveur esclave et on autorise le transfert d'enregistrement DNS et les logs du serveur maitre au serveur esclave.

Ensuite sur le serveur esclave il suffit d'aller dans `named.conf.default-zones` et de mettre ses paramètres :

```
zone "test.fr"{
    type slave ;
    file "test.zoned";
    masters { 192.168.100.130; };
};
```

On précise ici l'adresse IP du serveur maitre, on définit son type en slave et nous n'avons pas besoin de préciser de chemin de fichier il suffit juste de donner un nom au fichier qui contiendra les enregistrement DNS du serveur maitre, il créera automatiquement le fichier a cet emplacement :

```
root@DebianDNSSlave:/var/cache/bind# ls
managed-keys.bind managed-keys.bind.jnl test.zoned
```

Test tolérance de panne DNS :

Tout d'abord nous devons configurer le client comme ceci :

```
GNU nano 7.2 /etc/resolv.conf *
domain test.fr
search test.fr
nameserver 192.168.100.130
nameserver 192.168.100.131_
```

Il a le serveur DNS principale et l'esclave en DNS auxiliaire.

Nous allons tester si le serveur esclave est fonctionnel, pour faire ceci on désactive le serveur maitre et on regarde si le serveur esclave prend bien le relais :

```

root@IBsrvli12:~# nslookup test.fr
Server:          192.168.100.130
Address:         192.168.100.130#53

Name:   test.fr
Address: 192.168.100.130

root@IBsrvli12:~# nslookup test.fr
Server:          192.168.100.130
Address:         192.168.100.130#53

Name:   test.fr
Address: 192.168.100.130

root@IBsrvli12:~# systemctl stop bind9
root@IBsrvli12:~# nslookup test.fr
;; communications error to 192.168.100.130#53: connection refused
;; communications error to 192.168.100.130#53: connection refused
;; communications error to 192.168.100.130#53: connection refused
Server:          192.168.100.131
Address:         192.168.100.131#53

Name:   test.fr
Address: 192.168.100.130
;; communications error to 192.168.100.130#53: connection refused
;; communications error to 192.168.100.130#53: connection refused
;; communications error to 192.168.100.130#53: connection refused

```

Ici nous voyons que le serveur maitre est fonctionnel, ensuite on le désactive et on refait un nslookup, qui échoue au départ et ensuite marche avec l'adresse IP du serveur esclave.

On peut regarder dans le fichier `test.zoned` créer dans `var/cache/bind` :

```
GNU nano 7.2 test.zoned
^@^@^@B^@^@^@Af17/^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@P^@^@A^@^@F^@^@^@^@
:~^@^@^@A^@^@^@^@test^@Bfr^@^@01TBscwll12^@Btr^@Bfr^@^@Dnnot^@Btr^@Bfr^@^@^@^@A^@^@:~^@^@A^@^@
```

C'est illisible mais on voit que le fichier a bien été créer et a bien récupérer les enregistrements DNS du serveur maître.