

# Serveur Debian DS1 : routage et translation d'adresses

SAOU Rayan

## Table des matières

Table des matières .....	1
1. Configuration réseau DS1.....	1
2. Ajout de l'interface enp0s8.....	5
3. Transformation du serveur en routeur .....	8
4. Configuration du NAT sur le serveur DS1.....	12

Attention, certaines images ont été *corrompues* pendant leur traitement après sauvegarde du document, de ce fait, des images peuvent présenter quelques incohérences.

### 1. Configuration réseau DS1

Une fois connecté à notre DEB13Server, nous faisons un *apt-get update* pour récupérer les paquets à mettre à jour.

```
root@DEB13Server: ~#apt-get update
Atteint : 1 http://deb.debian.org/debian trixie InRelease
Réception de : 2 http://security.debian.org/debian-security trixie-security InRelease [43,4 kB]
Réception de : 3 http://deb.debian.org/debian trixie-updates InRelease [47,3 kB]
Réception de : 4 http://security.debian.org/debian-security trixie-security/main Sources [117 kB]
Réception de : 5 http://security.debian.org/debian-security trixie-security/main amd64 Packages [94,2 kB]
Réception de : 6 http://security.debian.org/debian-security trixie-security/main Translation-en [59,3 kB]
361 ko réceptionnés en 11s (34,2 ko/s)
Lecture des listes de paquets... Fait
```

Puis, nous vérifions que le prompt en couleur est bien actif dans `/root/.bashrc` à l'aide de l'éditeur *Nano*

```

GNU nano 8.4 /t
# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 is set in /etc/profile, and the default umask is defined
# in /etc/login.defs. You should not need this unless you want different
# defaults for root.
PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'
alias grep='grep --color=auto'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
PS1='\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\] \w\$ \[\033[00m\] '

```

Nous nous déconnectons et reconnectons avec la commande **logout**

Ensuite, nous renommons notre machine virtuelle en DS1 en modifiant les fichiers **/etc/hostname** et **/etc/hosts**

```

GNU nano 8.4 /etc/hostname *
DS1_

```

```

GNU nano 8.4
127.0.0.1    localhost
127.0.1.1    DS1_

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

```

Et nous redémarrons la machine à l'aide de la commande **nano**

Nous vérifions la configuration réseau grâce à la commande **ip -4 address** que notre carte est bien une IP en **172.17.XX.YY**, de ce fait nous sommes bien en accès pont sur notre VM.

```
root@DS1: ~#ip -4 address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   altnam enx08002791104d
   inet 172.17.1.19/16 brd 172.17.255.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85905sec preferred_lft 75105sec
root@DS1: ~#_
```

Donc, nous pouvons continuer en installant le paquet **resolvconf** avec la commande **apt-get install resolvconf**

```
root@DEB13Server: ~#apt-get install resolvconf
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
```

Ensuite, nous désactivons la carte réseau **enp0s3** avec la commande **ifdown** pour pouvoir changer notre adresse IP en statique.

```
root@DEB13Server: ~#ifdown enp0s3
sending signal ALRM to pid 650
waiting for pid 650 to exit
root@DEB13Server: ~#_
```

Pour pouvoir modifier l'adresse IP en statique nous modifions avec l'éditeur **Nano** le fichier **/etc/network/interfaces** et nous faisons les modifications :

Type d'IP : **Statique**

Adresse IP : **172.17.101.220**

Masque : **255.255.0.0**

Réseau : **172.17.0.0**

Broadcast : **172.17.255.255**

GW : **172.17.250.3**

DNS : **172.17.254.1**

Ensuite nous réactivons la carte réseau avec la commande **ifup**, et nous vérifions que l'adresse IP est correcte :

```

GNU nano 8.4 /etc/network/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 lo inet loopback

# The primary network interface
allow-hotplug enp0s3
auto enp0s3
iface enp0s3 inet static

address 172.17.101.120
netmask 255.255.0.0
network 172.17.0.0
gateway 172.17.250.3
broadcast 172.17.255.255
dns-nameservers 172.17.254.1_
# This is an autoconfigured IPv6 interface
iface enp0s3 inet6 auto

```

```

root@DS1: ~#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 08:00:27:91:10:4d brd ff:ff:ff:ff:ff:ff
    altname enx08002791104d
    inet 172.17.101.220/16 brd 172.17.255.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe91:104d/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever

```

Puis, nous consultons la table de routage avec la commande *ip route*

```


```

Et pour vérifier que tout fonctionne, nous effectuons un ping sur le serveur ROI : **172.17.254.1**

```

root@DS1: ~#ping 172.17.254.1
PING 172.17.254.1 (172.17.254.1) 56(84) bytes of data.
64 bytes from 172.17.254.1: icmp_seq=1 ttl=128 time=7.26 ms
64 bytes from 172.17.254.1: icmp_seq=2 ttl=128 time=4.10 ms
64 bytes from 172.17.254.1: icmp_seq=3 ttl=128 time=2.98 ms
64 bytes from 172.17.254.1: icmp_seq=4 ttl=128 time=3.46 ms
64 bytes from 172.17.254.1: icmp_seq=5 ttl=128 time=3.39 ms
^C
--- 172.17.254.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4014ms
rtt min/avg/max/mdev = 2.981/4.237/7.259/1.552 ms
root@DS1: ~#_

```

Tout fonctionne, nous pouvons donc essayer de *ping 8.8.8.8 (Google) et ac-nice.fr*

```

root@DS1: ~#ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=18.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=13.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=12.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=12.5 ms

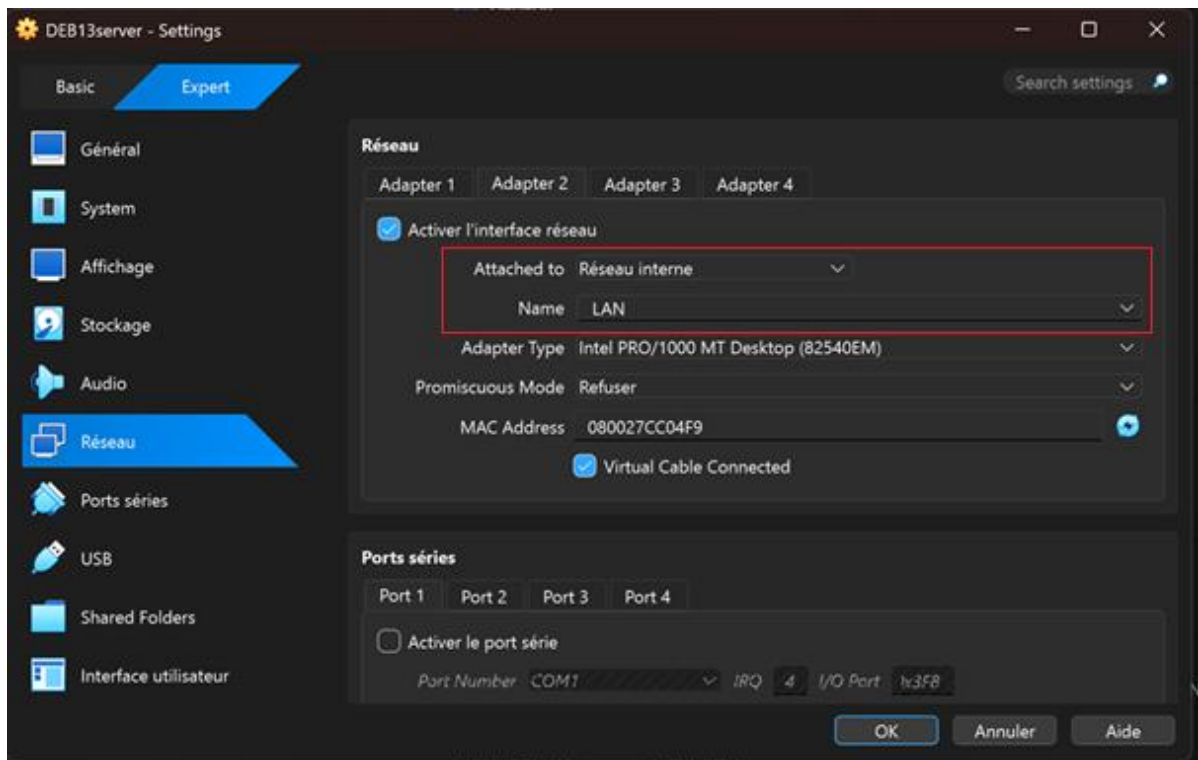
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 12.511/14.211/18.129/2.278 ms
root@DS1: ~#ping -c 4 ac-nice.fr
PING ac-nice.fr (185.75.143.93) 56(84) bytes of data.
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=1 ttl=50 time=29.8 ms
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=2 ttl=50 time=26.7 ms
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=3 ttl=50 time=29.5 ms
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=4 ttl=50 time=28.5 ms

--- ac-nice.fr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3010ms
rtt min/avg/max/mdev = 26.693/28.611/29.834/1.217 ms
root@DS1: ~#_

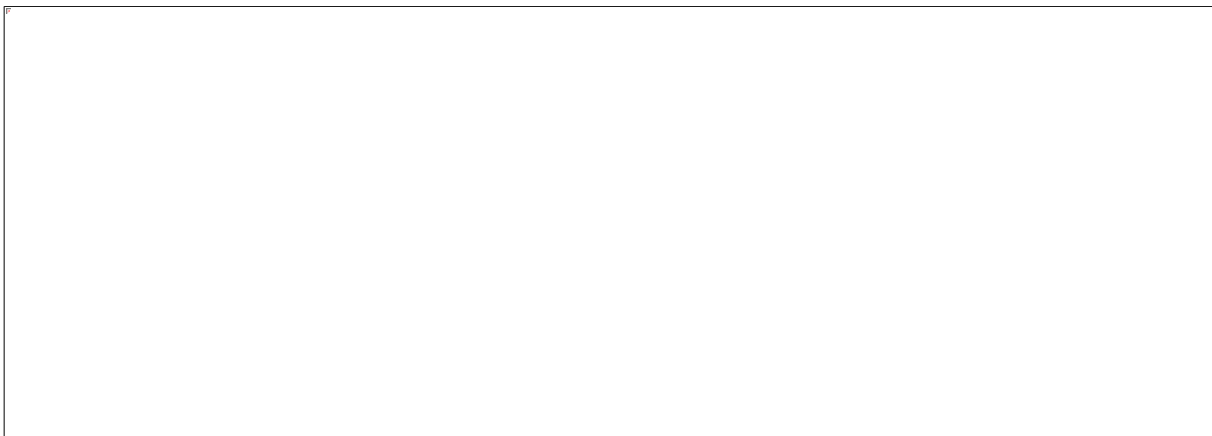
```

## 2. Ajout de l'interface enp0s8

Nous éteignons la machine, puis, depuis l'interface Gestion machines de VirtualBox. Nous sélectionnons le second adaptateur réseau puis nous sélectionnons le mode « Réseau interne » et nous le nommons en *LAN*



Puis, nous pouvons démarrer la machine à nouveau, et nous vérifions que la carte réseau a bien été ajoutée en tapant la **commande *ip a (enp0s8)***



Donc, nous allons modifier le fichier **/etc/interfaces** avec l'éditeur *Nano* pour la configuration de **enp0s8**

```

GNU nano 8.4 /etc/network/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 lo inet loopback

# The primary network interface

allow-hotplug enp0s3
auto enp0s3
iface enp0s3 inet static
address 172.17.101.220
netmask 255.255.0.0
network 172.17.0.0
broadcast 172.17.255.255
gateway 172.17.250.3
dns-nameservers 172.17.254.1

allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.4.254
netmask 255.255.255.0
network 192.168.4.0
broadcast 192.168.4.255

# This is an autoconfigured IPv6 interface
#iface enp0s3 inet6 auto

```

Nous activons la carte (*ifup*) et nous vérifions la configuration de la carte

```

root@DS1: ~#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 08:00:27:91:10:4d brd ff:ff:ff:ff:ff:ff
    altname enx08002791104d
    inet 172.17.101.220/16 brd 172.17.255.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe91:104d/64 scope link dadfailed tentative proto kernel_ll
        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:cc:04:f9 brd ff:ff:ff:ff:ff:ff
    altname enx080027cc04f9
    inet 192.168.4.254/24 brd 192.168.4.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fecc:4f9/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
root@DS1: ~#_

```

Enfin pour vérifier la bonne configuration nous effectuons un ping sur les 2 interfaces

```

sio@sio-VirtualBox: ~
sio@sio-VirtualBox:~$ ping -c 4 192.168.4.254
PING 192.168.4.254 (192.168.4.254) 56(84) bytes of data.
64 bytes from 192.168.4.254: icmp_seq=1 ttl=64 time=8.60 ms
64 bytes from 192.168.4.254: icmp_seq=2 ttl=64 time=4.86 ms
64 bytes from 192.168.4.254: icmp_seq=3 ttl=64 time=2.31 ms
64 bytes from 192.168.4.254: icmp_seq=4 ttl=64 time=3.71 ms

--- 192.168.4.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 2.306/4.869/8.601/2.336 ms
sio@sio-VirtualBox:~$ ping -c 4 172.17.101.220
PING 172.17.101.220 (172.17.101.220) 56(84) bytes of data.
64 bytes from 172.17.101.220: icmp_seq=1 ttl=64 time=2.37 ms
64 bytes from 172.17.101.220: icmp_seq=2 ttl=64 time=3.54 ms
64 bytes from 172.17.101.220: icmp_seq=3 ttl=64 time=5.54 ms
64 bytes from 172.17.101.220: icmp_seq=4 ttl=64 time=5.93 ms

--- 172.17.101.220 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3015ms
rtt min/avg/max/mdev = 2.371/4.345/5.931/1.455 ms
sio@sio-VirtualBox:~$

```

Enfin, nous affichons la table de routage avec la commande *ip route*

```

sio@sio-VirtualBox:~$ ip route
default via 192.168.4.254 dev enp0s3 proto static metric 100
192.168.4.0/24 dev enp0s3 proto kernel scope link src 192.168.4.1 metric 100

```

### 3. Transformation du serveur en routeur

Nous allons transformer le serveur DS1 en routeur pour faire transiter les paquets arrivant de *enp0s8* vers *enp0s3*

Pour activer le routage, nous faisons une commande *echo 1* et nous redirigeons la sortie dans le fichier */proc/sys/net/ipv4/ip\_forward*

Puis nous vérifions avec la commande *cat* que la modification a été faite

```

root@DS1: ~#cat /proc/sys/net/ipv4/ip_forward
1
root@DS1: ~#

```

Pour que le routage soit mis en place après chaque démarrage de la machine, nous créons dans le répertoire `/etc/sysctl.d/`, un fichier nommé `sysctl.conf`, où il comprendra la ligne `net.ipv4.ip_forward=1` :

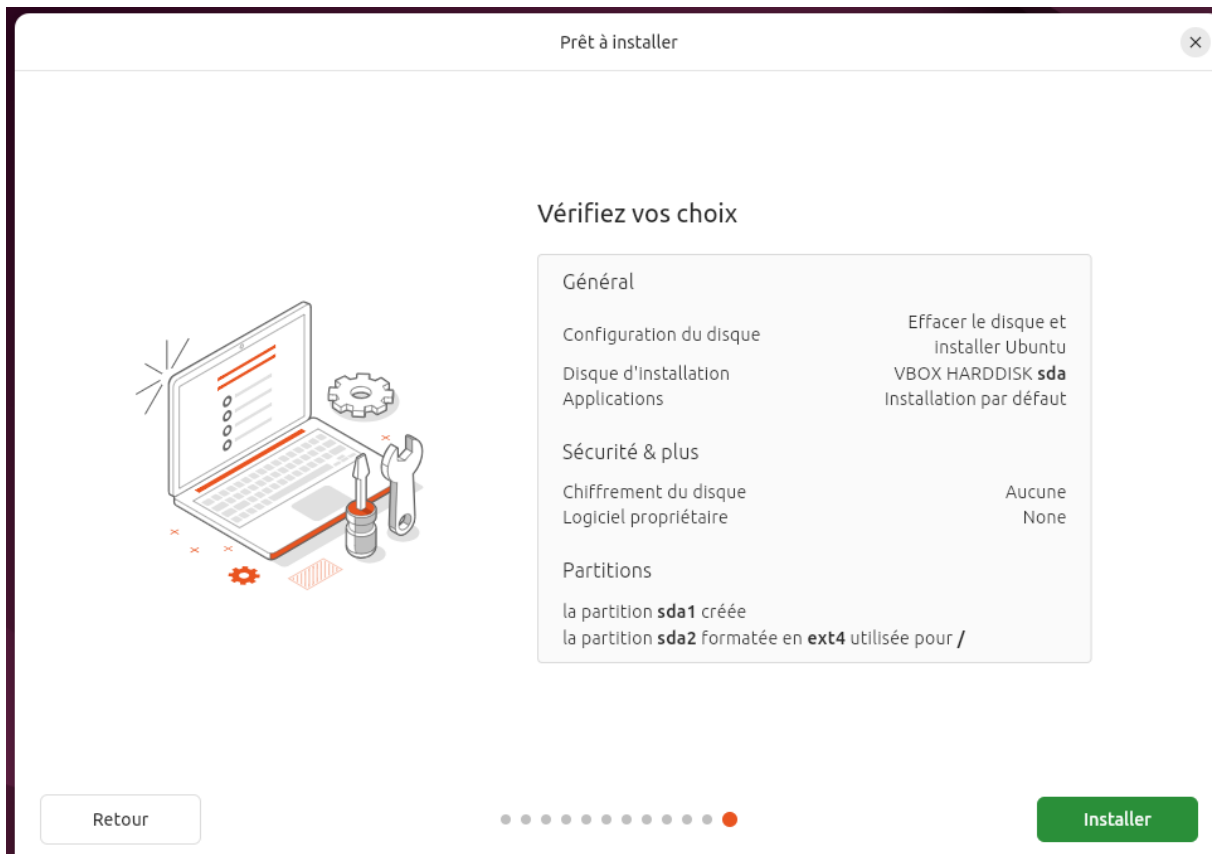
```
GNU nano 8.4 /etc/sysctl.d/sysctl.conf
net.ipv4.ip_forward=1
```

Nous redémarrons le PC, avec la commande `reboot` pour vérifier que le changement est bien mis en place dans le fichier `/proc/sys/net/ipv4/ip_forward`

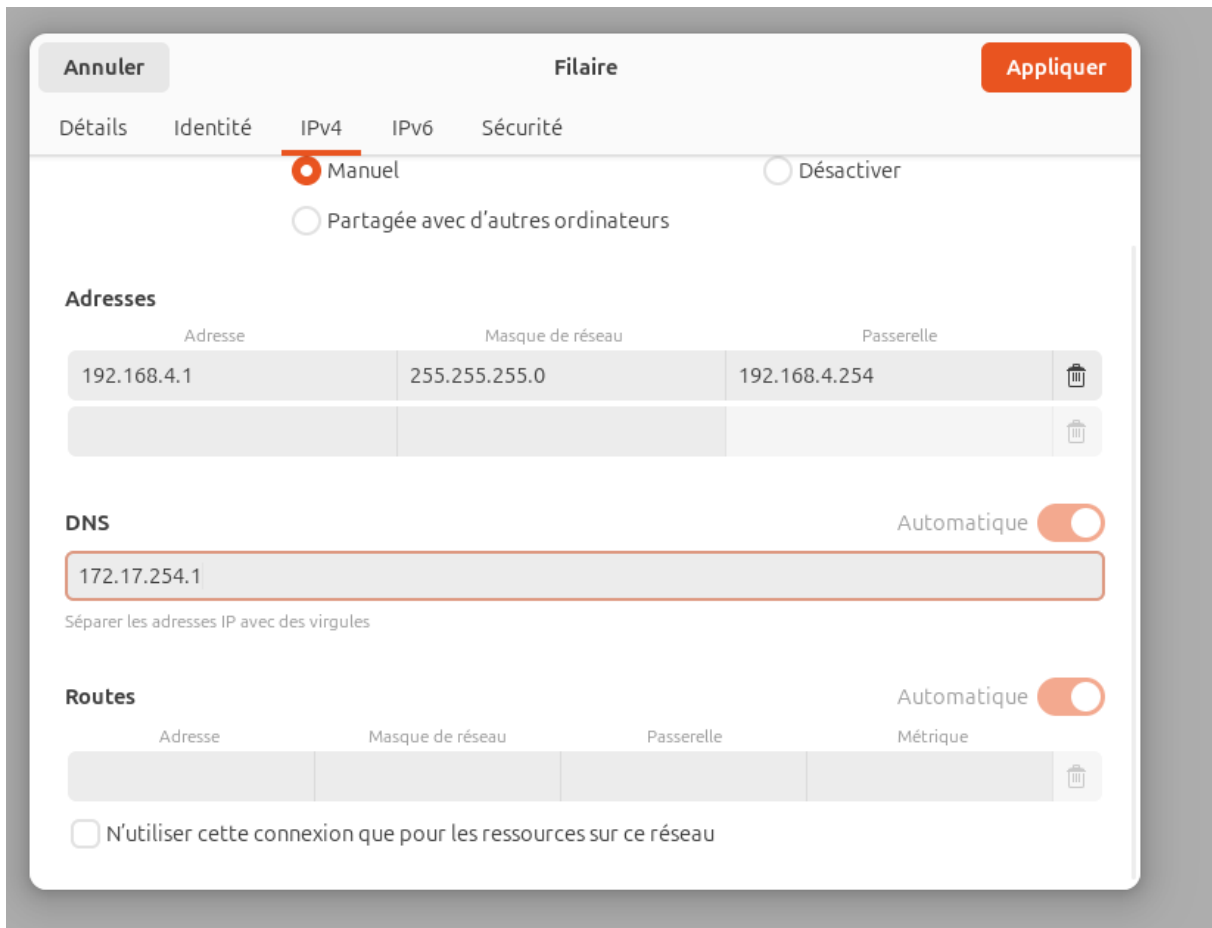
```
root@DS1: ~#cat /proc/sys/net/ipv4/ip_forward
1
root@DS1: ~#_
```

Les changements sont donc bien effectifs après le redémarrage.

Nous installons Ubuntu avec *précaution...*



Après avoir fini, nous installons les Additions Invités puis nous modifions les paramètres de la carte réseau.



Et nous vérifions la configuration de la carte réseau grâce à la commande `ip a`

```
sio@sio-VirtualBox:~$ 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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:9d:3a:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.1/24 brd 192.168.4.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
sio@sio-VirtualBox:~$
```

Puis nous consultons la table de routage avec la commande *ip route*

```

sio@io-VirtualBox:~$ ip route
default via 192.168.4.254 dev enp0s3 proto static metric 20100
192.168.4.0/24 dev enp0s3 proto kernel scope link src 192.168.4.1 metric 100
sio@io-VirtualBox:~$

```

Nous essayons de faire un ping pour vérifier la connectivité entre UD1 et les deux interfaces de DS1

```

sio@io-VirtualBox:~$ ping -c 4 172.17.101.220
PING 172.17.101.220 (172.17.101.220) 56(84) bytes of data.
64 bytes from 172.17.101.220: icmp_seq=1 ttl=64 time=7.06 ms
64 bytes from 172.17.101.220: icmp_seq=2 ttl=64 time=5.01 ms
64 bytes from 172.17.101.220: icmp_seq=3 ttl=64 time=2.50 ms
64 bytes from 172.17.101.220: icmp_seq=4 ttl=64 time=5.73 ms

--- 172.17.101.220 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 2.499/5.073/7.055/1.657 ms
sio@io-VirtualBox:~$ ping -c 4 192.168.4.254
PING 192.168.4.254 (192.168.4.254) 56(84) bytes of data.
64 bytes from 192.168.4.254: icmp_seq=1 ttl=64 time=2.10 ms
64 bytes from 192.168.4.254: icmp_seq=2 ttl=64 time=2.88 ms
64 bytes from 192.168.4.254: icmp_seq=3 ttl=64 time=2.61 ms
64 bytes from 192.168.4.254: icmp_seq=4 ttl=64 time=4.22 ms

--- 192.168.4.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 2.100/2.950/4.215/0.781 ms
sio@io-VirtualBox:~$

```

Nous essayons de ping le serveur ROI...

```

sio@sio-VirtualBox:~$ ping -c 4 172.17.254.1
PING 172.17.254.1 (172.17.254.1) 56(84) bytes of data.
From 192.168.4.254 icmp_seq=1 Destination Host Unreachable
From 192.168.4.254 icmp_seq=2 Destination Host Unreachable
From 192.168.4.254 icmp_seq=3 Destination Host Unreachable
From 192.168.4.254 icmp_seq=4 Destination Host Unreachable

--- 172.17.254.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3065ms
pipe 4
sio@sio-VirtualBox:~$ ping -c 4 172.17.254.1

```

Mais cela ne marche pas, car le serveur ne sait pas à qui s'adresser car il ne connaît pas le réseau 192.168.4.0.

Il faut donc faire une opération NAT pour que le serveur ROI puisse répondre à la carte du PC qu'il connaît (car ils sont dans le réseau 172.17.0.0)

#### 4. Configuration du NAT sur le serveur DS1

Pour ce faire nous devons installer le paquet iptables

```

-bash: alias: !grep --color=auto : non trouvé
root@DS1: ~#apt-get install iptables
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 :
  libip4tc2 libip6tc2 libnetfilter-contrack3 libnfnetlink0
Paquets suggérés :
  firewalld
Les NOUVEAUX paquets suivants seront installés :
  iptables libip4tc2 libip6tc2 libnetfilter-contrack3 libnfnetlink0
0 mis à jour, 5 nouvellement installés, 0 à enlever et 30 non mis à jour.
Il est nécessaire de prendre 458 kB dans les archives.
Après cette opération, 2 799 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O

```

Après installation, nous créons la règle suivante :

```

root@DS1: ~#iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.4.0/24 -j MASQUERADE

```

Nous spécifions dans quelle table (*NAT*) et quelle chaîne (*POSTROUTING*), ainsi que la carte qui fait le rôle de contact extérieur (outside) ainsi que le remplacement de l'adresse IP source du paquet (*192.168.4.0/24*) par celle de l'interface (*MASQUERADE*)

Nous vérifions que la règle a bien été ajoutée

```

root@DS1: ~#iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target      prot opt in     out     source      destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target      prot opt in     out     source      destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target      prot opt in     out     source      destination

Chain POSTROUTING (policy ACCEPT 1 packets, 70 bytes)
 pkts bytes target      prot opt in     out     source      destination
  2    120 MASQUERADE all  --  any    enp0s3  192.168.4.0/24  anywhere
root@DS1: ~#_

```

Pour qu'à chaque démarrage, la translation NAT soit effectuée, nous installons le paquet iptables-persistent

```

root@DS1: ~#apt-get install iptables-persistent
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 :
 netfilter-persistent
Les NOUVEAUX paquets suivants seront installés :
 iptables-persistent netfilter-persistent
0 mis à jour, 2 nouvellement installés, 0 à enlever et 30 non mis à jour.
Il est nécessaire de prendre 18,5 kB dans les archives.
Après cette opération, 96,3 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] 0
Réception de : 1 http://deb.debian.org/debian trixie/main amd64 netfilter-persistent all 1.0.23 [7 948 B]
Réception de : 2 http://deb.debian.org/debian trixie/main amd64 iptables-persistent all 1.0.23 [10,5 kB]
18,5 ko réceptionnés en 2s (9 079 o/s)

```

Configuration de iptables-persistent

Les règles actuelles peuvent être enregistrées dans le fichier de configuration « /etc/iptables/rules.v6 ». Ces règles seront chargées au prochain redémarrage de la machine.

Les règles ne sont enregistrées automatiquement que lors de l'installation du paquet. Veuillez consulter la page de manuel de iptables-save(8) pour connaître la manière de garder à jour le fichier des règles.

Faut-il enregistrer les règles IPv6 actuelles ?

Oui
  Non

Puis nous relançons notre machine avec la commande reboot.

Une fois le redémarrage effectué, nous regardons avec la commande iptables -t nat -L -v pour vérifier si la règle est toujours présente

```

root@DS1: ~#iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 9 packets, 2445 bytes)
 pkts bytes target    prot opt in     out     source destination
Chain INPUT (policy ACCEPT 2 packets, 612 bytes)
 pkts bytes target    prot opt in     out     source destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source destination
  0      0 MASQUERADE all  --  any    enp0s3  192.168.4.0/24  anywhere
root@DS1: ~#_

```

Une fois que nous avons bien vérifié que la règle a bien été mise, nous retournons sur notre UDI pour re tester la commande vers le serveur ROI qui était jusque-là non-réussite

```

sio@sio-VirtualBox:~$ ping -c 4 172.17.254.1
PING 172.17.254.1 (172.17.254.1) 56(84) bytes of data.
64 bytes from 172.17.254.1: icmp_seq=1 ttl=127 time=1980 ms
64 bytes from 172.17.254.1: icmp_seq=2 ttl=127 time=1635 ms
64 bytes from 172.17.254.1: icmp_seq=3 ttl=127 time=610 ms
64 bytes from 172.17.254.1: icmp_seq=4 ttl=127 time=47.9 ms

--- 172.17.254.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 47.900/1068.126/1980.119/775.236 ms, pipe 2
sio@sio-VirtualBox:~$

```

La commande *ping* a été réalisée avec succès.

Sur DS1 nous installons le paquet *tcpdump*

```

root@DS1: ~#apt-get install tcpdump
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 :
  libpcap0.8t64
Les NOUVEAUX paquets suivants seront installés :
  libpcap0.8t64 tcpdump
0 mis à jour, 2 nouvellement installés, 0 à enlever et 30 non mis à jour.
Il est nécessaire de prendre 640 kB dans les archives.
Après cette opération, 1 775 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] _

```

A l'aide de la commande `tcpdump` nous pouvons écouter et capturer les trames sur l'interface `enp0s3` (et donc au même moment nous faisons notre `ping` sur **UDI**)

```

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link type EN10MB (Ethernet), snapshot length 262144 bytes
16:26:32.011449 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 1, length 64
16:26:32.036005 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 1, length 64
16:26:33.014603 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 2, length 64
16:26:33.063644 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 2, length 64
16:26:34.019839 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 3, length 64
16:26:34.032359 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 3, length 64
16:26:35.025172 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 4, length 64
16:26:35.037444 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 4, length 64

```

L'IP a donc changé de `192.168.4.1` en `172.17.101.220`, car il s'agit de la carte « *outside* », la translation a bien eu lieu

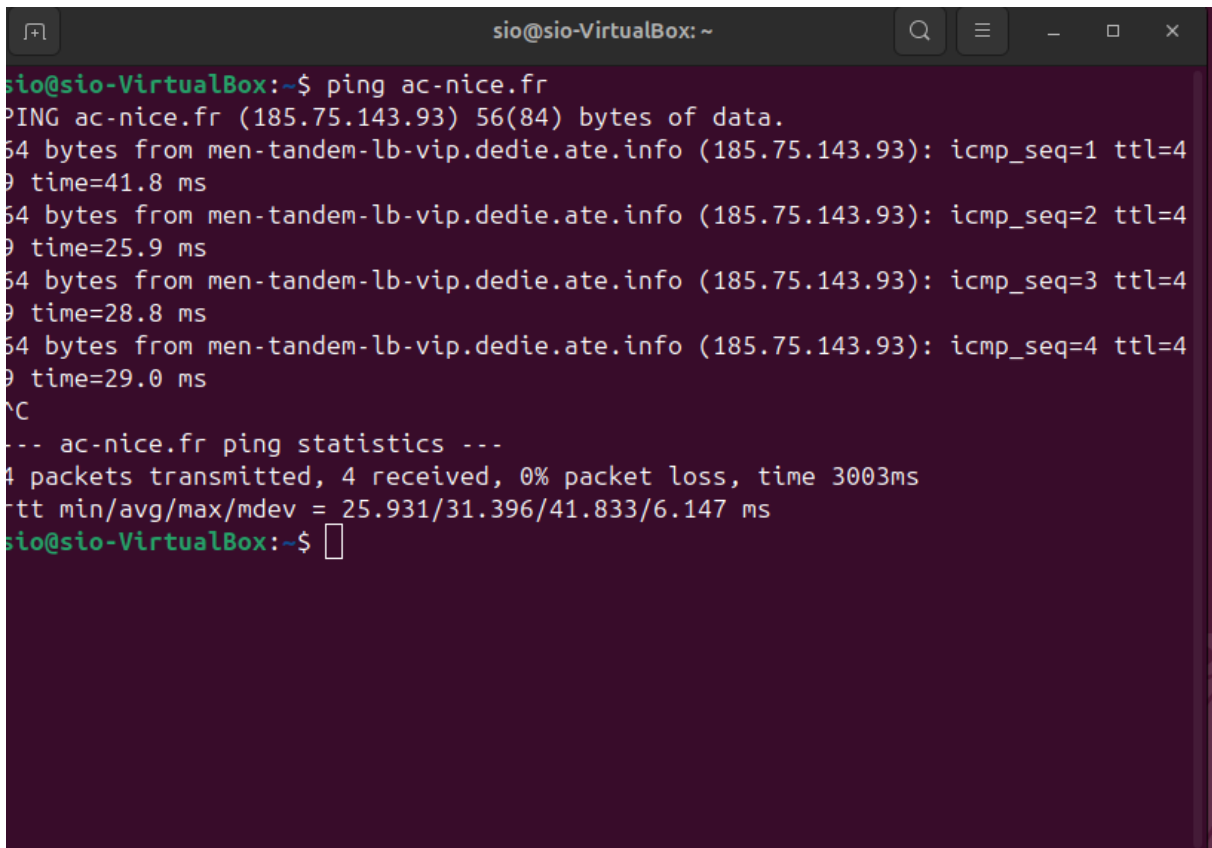
Et nous faisons de même avec l'interface `enp0s8`

```

root@DS1: ~#tcpdump -i enp0s8 icmp -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link type EN10MB (Ethernet), snapshot length 262144 bytes
16:29:12.103241 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 1, length 64
16:29:13.139455 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 2, length 64
16:29:13.475102 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 1, length 64
16:29:13.865870 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 2, length 64
16:29:14.141583 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 3, length 64
16:29:14.565835 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 3, length 64
16:29:15.145922 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 4, length 64
16:29:15.204264 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 4, length 64

```

Nous pouvons donc ping un serveur qui autorise les *requêtes ICMP* comme *ac-nice.fr* depuis UD1



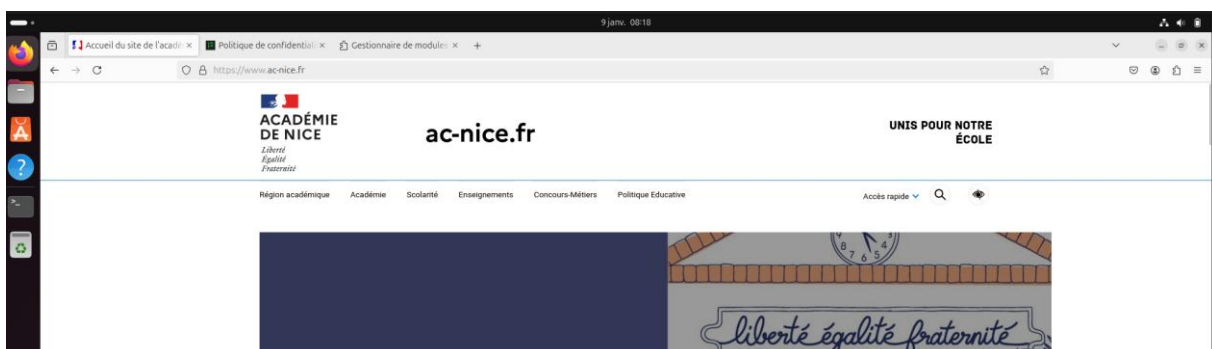
```

sio@sio-VirtualBox: ~
sio@sio-VirtualBox:~$ ping ac-nice.fr
PING ac-nice.fr (185.75.143.93) 56(84) bytes of data:
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=1 ttl=49 time=41.8 ms
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=2 ttl=49 time=25.9 ms
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=3 ttl=49 time=28.8 ms
64 bytes from men-tandem-lb-vip.dedie.ate.info (185.75.143.93): icmp_seq=4 ttl=49 time=29.0 ms
^C
--- ac-nice.fr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 25.931/31.396/41.833/6.147 ms
sio@sio-VirtualBox:~$

```

Le ping est un succès, la *résolution DNS* et la *translation* marchent, car nous avons la réponse qui nous est parvenu.

Donc nous pouvons aussi y accéder avec un navigateur comme *Firefox*



Cela veut donc dire que la *translation* et la *résolution DNS* marchent, car nous avons la réponse qui nous est parvenue.

Pour illustrer la translation à l'aide d'un schéma :

*Enp0s3 : Après la translation, la carte « outside »*

*Enp0s8 : Avant la translation, la carte « inside »*

```

root@001: ~# tcpdump -i enp0s3 icmp -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type ETHER (Ethernet), snapshot-length 262144 bytes
16:26:32.011449 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 1, length 64
16:26:32.036405 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 1, length 64
16:26:33.014603 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 2, length 64
16:26:33.063644 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 2, length 64
16:26:34.019839 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 3, length 64
16:26:34.032359 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 3, length 64
16:26:35.025172 IP 172.17.101.220 > 172.17.254.1: ICMP echo request, id 11886, seq 4, length 64
16:26:35.037444 IP 172.17.254.1 > 172.17.101.220: ICMP echo reply, id 11886, seq 4, length 64

root@001: ~# tcpdump -i enp0s8 icmp -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type ETHER (Ethernet), snapshot-length 262144 bytes
16:29:12.103241 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 1, length 64
16:29:13.139455 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 2, length 64
16:29:13.479100 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 1, length 64
16:29:13.865870 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 2, length 64
16:29:14.141503 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 3, length 64
16:29:14.565835 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 3, length 64
16:29:15.145922 IP 192.168.4.1 > 172.17.254.1: ICMP echo request, id 12509, seq 4, length 64
16:29:15.204264 IP 172.17.254.1 > 192.168.4.1: ICMP echo reply, id 12509, seq 4, length 64
    
```