

**Aix Marseille Université - Campus Luminy**

UFR des Sciences

**Rapport de TP**

Master Informatique

*Module Réseaux.*

**TP n°1 :**

***Routage Simple IPv4.***

**Réalisé par :**

ZEMMOURI Yasmine.

## 1. Mise en Réseau Simple :

### 1.1. Nouvelles VMs :

La création et le démarrage des deux machines virtuelles VM1 et VM2 se fait sans problèmes avec *vagrant* via la commande *vagrant halt* dans le dossier spécifique à chacune.

### 1.2. Observation du Trafic Réseau :

- 1- L'affectation des adresses IPv4 aux deux machines VM1 et VM2 s'est faite avec succès : les deux machines se voient par des *ping* (utilisation de la *commande ip address add 192.168.1.1 /24 dev eth1* pour VM1 et *ip address add 192.168.1.2/24 dev eth1* pour VM2). Cela signifie qu'elles communiquent entre elles en utilisant le protocole ICMP pour tester la connectivité du réseau. Les deux machines sont ainsi connectées :

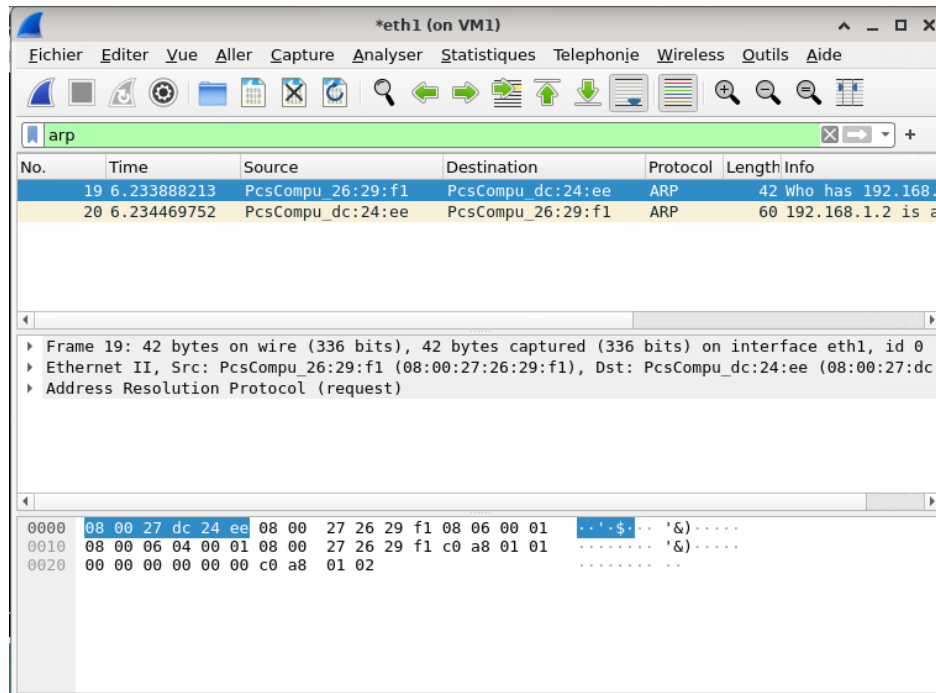
```
m1reseaux@VM1:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
 64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.903 ms
 64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.801 ms
 64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.673 ms
 64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.671 ms
 64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=0.704 ms
^C
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.671/0.750/0.903/0.089 ms

m1reseaux@VM2:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
 64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.33 ms
 64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.681 ms
 64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2.06 ms
 64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.680 ms
 64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.754 ms
^C
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4090ms
rtt min/avg/max/mdev = 0.680/1.099/2.055/0.535 ms
```

2- Pour mettre en évidence le protocole ARP via wireshark, on suit les étapes suivantes :

- Lancer Wireshark sur l'une des machines (ou les deux).
- Sélectionner l'interface réseau appropriée (dans notre cas eth1).
- Appliquer le filtre 'ARP' afin d'afficher uniquement les paquets ARP.
- Commencer la capture sur trafic réseau.
- Pinguer l'autre machine.
- Stopper la capture.

Nous aurons ainsi les échanges ARP entre VM1 et VM2 pendant le temps de cette capture.



## 2. Interconnexion de deux Sous-réseaux par un Routeur :

### 2.1. Initialisation des VMS :

1- La création et le démarrage des trois machines virtuelles se fait avec succès via la commande 'vagrant up'. La modification des fichiers 'VagrantFile' est faite initialement pour :

- ✓ Avoir un réseau interne de nom LAN1 pour VM1 via la ligne `config.vm.network "private_network", ip:"0.0.0.0", auto_config: false, virtualbox__intnet: "LAN1"`.
- ✓ Avoir un réseau interne de nom LAN2 pour VM3 via la ligne `config.vm.network "private_network", ip:"0.0.0.0", auto_config: false, virtualbox__intnet: "LAN2"`.
- ✓ Avoir deux réseaux internes de nom LAN1 et LAN2 pour VM2 via les lignes `config.vm.network "private_network", ip:"0.0.0.0", auto_config: false, virtualbox__intnet: "LAN1"`, et `config.vm.network "private_network", ip:"0.0.0.0", auto_config: false, virtualbox__intnet: "LAN2"`.

2- La table de routage initiale de chaque machine est vide : on ne trouve aucune configuration dans l'interface eth1 pour les trois machines (en addition à eth2 pour VM2). Cela veut dire que pour le moment, aucune adresses IP n'a été affectée aux réseaux internes existants. En

d'autres termes, la machine ne sait pas par quel chemin envoyer les paquets de données vers d'autres machines ou réseaux.

**VM1 :**

```
m1reseaux@VM1:~$ ip addr show

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
        valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
eth0
        valid_lft 85998sec preferred_lft 85998sec
    inet6 fe80::a00:27ff:fe8d:c04d/64 scope link
        valid_lft forever preferred_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:e6:91:15 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
```

VM2 :

```
m1reseau@VM2:~$ ip addr show

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
        valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000

    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    altname enp0s3

    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
eth0
        valid_lft 86051sec preferred_lft 86051sec
    inet6 fe80::a00:27ff:fe8d:c04d/64 scope link
        valid_lft forever preferred_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000

    link/ether 08:00:27:e4:7b:8b brd ff:ff:ff:ff:ff:ff
    altname enp0s8

4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000

    link/ether 08:00:27:d8:73:b6 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
```

### VM3 :

```
m1reseaux@VM3:~$ ip addr show

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
        valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000

    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
eth0
        valid_lft 86133sec preferred_lft 86133sec
    inet6 fe80::a00:27ff:fe8d:c04d/64 scope link
        valid_lft forever preferred_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000

    link/ether 08:00:27:a7:3c:4b brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet6 fe80::c803:86f3:1281:e663/64 scope link
noprfixroute
        valid_lft forever preferred_lft forever
```

- 3- Désactiver la route par défaut, dans le but d'affecter une route spécifique à notre réseau et contrôler notre routage :

### VM1 :

```
m1reseaux@VM1:~$ ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src
10.0.2.15

m1reseaux@VM1:~$ sudo -s
[sudo] password for m1reseaux:
root@VM1:/home/m1reseaux# ip route delete default
root@VM1:/home/m1reseaux# ip route
10.0.2.0/24 dev eth0 proto kernel scope link src
10.0.2.15
```

#### VM2 :

```
m1reseaux@VM2:~$ ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src
10.0.2.15

m1reseaux@VM2:~$ sudo -s
[sudo] password for m1reseaux:
root@VM2:/home/m1reseaux# ip route delete default
root@VM2:/home/m1reseaux# ip route
10.0.2.0/24 dev eth0 proto kernel scope link src
10.0.2.15
```

#### VM3 :

```
m1reseaux@VM3:~$ ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src
10.0.2.15

m1reseaux@VM3:~$ sudo -s
[sudo] password for m1reseaux:
root@VM3:/home/m1reseaux# ip route delete default
root@VM3:/home/m1reseaux# ip route
10.0.2.0/24 dev eth0 proto kernel scope link src
10.0.2.15
```

## 2.2. Plan d'adressage :

- 1- Plan de configuration des sous-réseaux et machines avec l'adresse 172.16.2.128/25 avec 4 bits pour les numéros de machine :

Sachant qu'une adresse IP est codée sur 32bits (4 octets), nous avons un masque de 25bits, c'est-à-dire que l'adresse réseau principale est sur 25bits :

Masque : 1 0 0 0 0 0 0 0

L'adresse 172.16.2.128 devient, avec en bleu les bits d'ID réseau :

1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0

Si on réserve 4 bits pour les numéros de machine, on aura :

1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0

Ce qui laisse seulement 3bits pour l'adressage des sous-réseaux ( $32 - (25+4) = 3$ ).

Le masque des sous-réseaux deviendra par la suite /28 :

1 0 0 0 0

- 2- On obtient par conséquent 23 sous-réseaux (**8 sous-réseaux**).
- 3- On peut mettre effectivement  $2^4 - 2 = 16 - 2 = 14$  machines sur chaque réseau [le -2 pour l'adresse du réseau et l'adresse de diffusion].
- 4- LAN 1 :
  - Adresse IP : 172.16.2.132 /28.
  - Masque : /28 → 1 0 0 0 0.
  - Adresse de diffusion (broadcast, mettre les bits machine à 1) : 172.16.2.143 /28.
- LAN 2 :
  - Adresse IP : 172.16.2.162 /28.
  - Masque : /28 → 1 0 0 0 0.
  - Adresse de diffusion (broadcast, mettre les bits machine à 1) : 172.16.2.175 /28.

## 2.3. Configurations des machines VM1 et VM3 :

- 1- Affectation l'adresse 172.16.2.131 à VM1 et 172.16.2.163 à VM3 avec NetworkManager ou avec la commande *ip address add 172.16.2.131 ou 172.16.2.163*.



### VM1 :

```
root@VM1:/home/m1reseaux# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000

    link/ether 08:00:27:17:c6:55 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.131/28 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::9fdb:3c9f:360b:f13/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

### VM3 :

```
root@VM3:/home/m1reseaux# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000

    link/ether 08:00:27:1a:84:3a brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.163/28 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::4145:4615:bc3b:87c3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

- 2- Affectation d'une passerelle en remplissant le champs 'gateway' de VM1 avec l'adresse IP de VM2 sur le LAN1 : 172.16.2.132, et le champs 'gateway' de VM3 avec l'adresse IP de VM2 sur le LAN2 : 172.16.2.162.

- ➔ On peut indiquer la passerelle via la commande `ip route add` comme suit :
- Sur VM1 : *`ip route add 172.16.2.131/28 via 172.16.2.132 dev eth1.`*
  - Sur VM3 : *`ip route add 172.16.2.163/28 via 172.16.2.162 dev eth2.`*

## 2.4. Configuration de VM2 comme un routeur entre LAN1 et LAN2 :

- 1- Vérifier la création des routes de sorties vers les deux interfaces :

```
m1reseaux@VM2:~$ ip route
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
172.16.2.128/28 dev eth1 proto kernel scope link src 172.16.2.132
172.16.2.160/28 dev eth2 proto kernel scope link src 172.16.2.162
```

Cela signifie que la configuration du routeur (VM2) est enregistrée.

- 2- Activation du drapeau de routage *ip\_forward* :

Lorsque ce drapeau est activé, le système est configuré pour acheminer les paquets entre ses interfaces réseau, ce qui lui rend possible d’agir en tant que routeur pour connecter les différents réseaux IP.

## 2.5. Vérification :

Vérification du routage en pingant VM3 via VM1, et le contraire, VM1 via VM3 :

```
m1reseaux@VM1:~$ ping 172.16.2.163
PING 172.16.2.163 (172.16.2.163) 56(84) bytes of data.
64 bytes from 172.16.2.163: icmp_seq=1 ttl=64 time=0.903 ms
64 bytes from 172.16.2.163: icmp_seq=2 ttl=64 time=0.801 ms
64 bytes from 172.16.2.163: icmp_seq=3 ttl=64 time=0.673 ms
64 bytes from 172.16.2.163: icmp_seq=4 ttl=64 time=0.671 ms
64 bytes from 172.16.2.163: icmp_seq=5 ttl=64 time=0.704 ms
^C
--- 172.16.2.163 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.671/0.750/0.903/0.089 ms

m1reseaux@VM3:~$ ping 172.16.2.131
PING 172.16.2.131 (172.16.2.131) 56(84) bytes of data.
64 bytes from 172.16.2.131: icmp_seq=1 ttl=64 time=1.33 ms
64 bytes from 172.16.2.131: icmp_seq=2 ttl=64 time=0.681 ms
64 bytes from 172.16.2.131: icmp_seq=3 ttl=64 time=2.06 ms
64 bytes from 172.16.2.131: icmp_seq=4 ttl=64 time=0.680 ms
64 bytes from 172.16.2.131: icmp_seq=5 ttl=64 time=0.754 ms
^C
--- 172.16.2.131ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4090ms
rtt min/avg/max/mdev = 0.680/1.099/2.055/0.535 ms
```

On peut également faire une capture du trafic sur VM2, notre routeur. On remarque que les paquets que VM1 envoie à VM3 passe par VM2 (de VM3 vers VM1 passe par VM2), en traduisant les différents protocoles du réseau. Ce qui montre que VM2 est un chemin entre VM1 et VM3. Il est considéré comme un routeur.

Capture en cours de eth2 (on VM2)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
132	31.477983619	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request
133	31.846960366	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request
134	31.847609644	172.16.2.163	172.16.2.131	ICMP	98	Echo (ping) request
135	32.479119227	172.16.2.163	172.16.2.131	ICMP	98	Echo (ping) request
136	32.479788771	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request
137	32.849487359	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth2, id 0

Ethernet II, Src: PcsCompu\_1a:84:3a (08:00:27:1a:84:3a), Dst: PcsCompu\_cf:37:59 (08:00:27:cf:37:59)

Internet Protocol Version 4, Src: 172.16.2.163, Dst: 172.16.2.131

Internet Control Message Protocol

0000 08 00 27 cf 37 59 08 00 27 1a 84 3a 08 00 45 00 ...7Y... ..E..

0010 00 54 60 9a 40 00 40 01 7c c8 ac 10 02 a3 ac 10 ...T..@.. ..w....

0020 02 83 08 00 83 fb ec ae 01 0a fe 43 1b 65 00 00 ...^.....C.e..

0030 00 00 a8 cf 05 00 00 00 00 00 10 11 12 13 14 15 ...db.....

0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#\$\$%

0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()\*+,-./012345

0060 36 37 67

eth2: <live capture in progress> Paquets: 178 · Affichés: 178 (100.0%) Profile: Default

Capture en cours de eth1 (on VM2)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
563	147.076125038	fe80::2f01:eaaf:bb4...	ff02::2	ICMPv6	62	Router Solicitation
564	147.478053092	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request
565	147.478883290	172.16.2.163	172.16.2.131	ICMP	98	Echo (ping) request
566	147.561947630	172.16.2.163	172.16.2.131	ICMP	98	Echo (ping) request
567	147.564065916	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request
568	148.492854530	172.16.2.131	172.16.2.163	ICMP	98	Echo (ping) request

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth1, id 0

Ethernet II, Src: PcsCompu\_17:c6:55 (08:00:27:17:c6:55), Dst: PcsCompu\_e4:98:31 (08:00:27:e4:98:31)

Internet Protocol Version 4, Src: 172.16.2.131, Dst: 172.16.2.163

Internet Control Message Protocol

0000 08 00 27 e4 98 31 08 00 27 17 c6 55 08 00 45 00 ...1... ..U..E..

0010 00 54 d4 eb 40 00 40 01 08 77 c6 55 08 00 ac 10 ...T..@.. ..w....

0020 02 a3 08 00 84 85 5e 8d 01 0f d1 43 1b 65 00 00 ...^.....C.e..

0030 00 00 64 62 04 00 00 00 00 00 10 11 12 13 14 15 ...db.....

0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#\$\$%

0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()\*+,-./012345

0060 36 37 67

eth1: <live capture in progress> Paquets: 614 · Affichés: 614 (100.0%) Profile: Default

### 3. Automatisation de la configuration :

Afin d'automatiser la configuration des VMs et d'éviter de réitérer les étapes précédentes à chaque démarrage, on crée un fichier *config.yml* afin d'y spécifier toutes les configurations nécessaires. Ces configurations seront appliquées à la VM en se servant du logiciel **ansible**, plus précisément le module **nmcli**.

Comme chaque machine a une configuration propre à elle-même, les fichiers *config.yml* seront différents :

Pour VM1 :

```
# Configuration eth1 / Ansible playbook

# RAPPEL: eth0 est à vagrant, ne pas y toucher


- hosts: localhost

  remote_user: root

  tasks:


- name: Suppression de la passerelle par défaut

  shell: ip route del default


- name: Configuration de VM1/eth1

  nmcli:

    type: ethernet

    conn_name: eth1 via ansible

    ifname: eth1

    state: present

    autoconnect: true

    ip4: 172.16.2.131/28


- name: Activation explicite de cette configuration

  shell: nmcli con up "eth1 via ansible"


- name: Configuration de la route vers LAN1 via VM2
```

```
community.general.nmcli:

type: ethernet

conn_name: eth1 via ansible

ifname: eth1

# syntaxe simple : plage passerelle

routes4: 172.16.2.128/28 172.16.2.132

state: present
```

*Pour VM2:*

```
hosts: localhost

remote_user: root

tasks:

- name: Suppression de la passerelle par défaut

  shell: ip route del default


- name: Configuration de VM2/eth1

  nmcli:

    type: ethernet

    conn_name: eth1 via ansible

    ifname: eth1

    state: present

    autoconnect: true

    ip4: 172.16.2.132/28


- name: Activation explicite de cette configuration

  shell: nmcli con up "eth1 via ansible"


- name: Configuration de VM2/eth2

  nmcli:

    type: ethernet
```

conn\_name: eth2 via ansible

ifname: eth2

state: present

autoconnect: true

ip4: 172.16.2.162/28

- name: Activation explicite de cette configuration

shell: nmcli con up "eth2 via ansible"

# Activer le routage

- name: Activation du routage IPv4

sysctl:

name: net.ipv4.ip\_forward

value: '1'

sysctl\_set: yes

*Pour VM3:*

hosts: localhost

remote\_user: root

tasks:

- name: Suppression de la passerelle par défaut

shell: ip route del default

- name: Configuration de VM3/eth1

nmcli:

type: ethernet

conn\_name: eth1 via ansible

ifname: eth1

state: present

autoconnect: true

ip4: 172.16.2.163/28

**- name: Activation explicite de cette configuration**

**shell: nmcli con up "eth1 via ansible"**

**- name: Configuration de la route vers LAN2 via VM2**

**community.general.nmcli:**

**type: ethernet**

**conn\_name: eth1 via ansible**

**ifname: eth1**

**# syntaxe simple : plage passerelle**

**routes4: 172.16.2.160/28 172.16.2.132**

**state: present**

- ✓ Cette méthode de configuration est automatique, les configurations seront prises en compte dès le démarrage de la VM, nous évitant ainsi de les exécuter manuellement une par une. Elle prévient également le problème de perte de données qui peut arriver en arrêtant la VM.
- ✓ En conclusion, cette nouvelle méthode apporte un gain de temps et prévient la perte de configurations par rapport à l'ancienne.