

Mini labo réseau :

Objectif :

- Création d'un réseaux local isolé sur l'ordinateur avec plusieurs machines virtuelles.
- Apprentissage de la configuration des adresses IP statiques.
- Test de la connectivité entre les différentes machines.
- Découverte des outils de diagnostic réseau ([ping](#), [traceroute](#), [netcat](#), [iperf](#)).
- Positionner les bases pour des futurs projets en sécurité, administration système et routage.

Prérequis :

- Un PC sous Windows ou Linux avec VirtualBox ou VMware Workstation.
- 2 ou 3 machines virtuelles Linux (Ubuntu ou Debian conseillés)
- Quelques notions de base sur les adresses IP (IP, masque réseau).
- Connaissance minimale des commandes Linux ([ip a](#), [ping](#), [nano](#))

Rappel :

1- Notions de base sur les adresses IP :

a- IP :

Une adresse IP est codée sur 32 bits, elle permet d'identifier la machine sur un réseau.

Chaque machine se trouvant sur le même réseau possède une adresse IP différente. Cette dernière change aussi selon le réseau et les adresses IP disponibles sur ce dernier.

Par exemple :

Machine A : 192.168.10.3

Machine B : 192.168.10.9

b- Masque réseau :

Un masque réseau est un nombre qui sert à séparer l'adresse IP en deux parties. Pour mieux comprendre comment ce masque fonctionne faisons une analogie simple.

Imaginons une ville :

- La ville représente le réseau
- La rue représente la partie réseau de l'adresse IP
- La maison représente la partie hôte de l'adresse IP (plus précisément l'appareil dans la rue)

Prenons un exemple :

L'adresse IP → 192.168.4.20

Le masque → 255.255.255.0

Les trois 255 correspondent à la partie réseau donc à 192.168.4

Le 0, lui, correspond à la partie hôte donc ici à 20.

Si on reprend notre analogie pour l'adapter à notre exemple :

Le masque (255.255.255.0) représente la ville.

192.168.4 correspond à la rue avec toutes les maisons appartiennent au même réseau.

.10 représente la maison numéro 10 plus précisément la machine en particulier.

2- Commandes Linux :

a- IP a :

IP a est une commande sous Linux qui permet de voir l'adresse IP de la machine.

b- Ping :

Ping est une commande que l'on peut utiliser sous Windows et Linux. Elle permet de vérifier la connexion entre deux différentes machines qui se trouvent sur le même réseau. On utilise aussi cette commande pour vérifier la connexion avec Internet.

c- Nano :

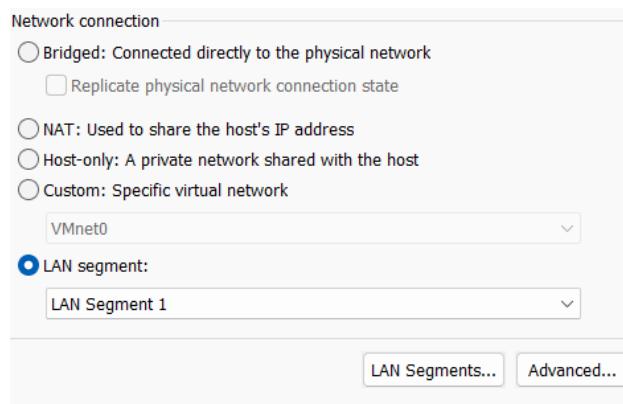
Nano est une commande sous Linux qui permet d'ouvrir le logiciel Nano, ce logiciel est un éditeur de texte. On peut ouvrir un fichier pour le modifier ou créer directement un nouveau fichier. Il faut juste utiliser nano suivi par le nom du fichier que l'on veut.

Etapes :

1- Préparation de l'environnement :

Commençons par créer deux VM Linux au minimum, Ubuntu ou Debian pour le système d'exploitation. Pour l'installation des systèmes d'exploitation il suffira de suivre les indications données lors de l'installation.

Une fois nos VM installées, nous allons les connectés en réseaux interne, Internal Network, pour créer un Lan, réseau local isolé, donc ici en LAN segment 1.



2- Configuration des adresses IP :

Tout d'abord nous donnons l'accès sudo à l'utilisateur sur les deux VM.

```
user@Mini-labo-VM2:~$ su -  
Mot de passe :  
root@Mini-labo-VM2:~# visudo  
root@Mini-labo-VM2:~# exit  
déconnexion
```

```
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
user    ALL=(ALL:ALL) ALL
```

Maintenant, nous allons attribuer des adresses IP statiques, adresses données manuellement qui ne change pas même après le redémarrage du PC.

VM1 :

```
user@Mini-labo-VM1:~$ 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 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:42:7d:f1 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.106.131/24 brd 192.168.106.255 scope global dynamic noprefixroute ens33  
        valid_lft 1598sec preferred_lft 1598sec  
    inet6 fe80::20c:29ff:fe42:7df1/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
user@Mini-labo-VM1:~$ sudo ip addr add 192.168.10.1/24 dev ens33  
[sudo] Mot de passe de user :  
user@Mini-labo-VM1:~$ 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 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:42:7d:f1 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.106.131/24 brd 192.168.106.255 scope global dynamic noprefixroute ens33  
        valid_lft 1519sec preferred_lft 1519sec  
    inet 192.168.10.1/24 scope global ens33  
        valid_lft forever preferred_lft forever  
    inet6 fe80::20c:29ff:fe42:7df1/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

VM2 :

```
user@Mini-labo-VM2:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:bb:40:18 brd ff:ff:ff:ff:ff:ff
    altnet enp2s1
        inet 192.168.106.132/24 brd 192.168.106.255 scope global dynamic noprefixroute ens33
            valid_lft 1495sec preferred_lft 1495sec
        inet6 fe80::20c:29ff:febb:4018/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
user@Mini-labo-VM2:~$ sudo ip addr add 192.168.10.2/24 dev ens33
[sudo] Mot de passe de user :
user@Mini-labo-VM2:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:bb:40:18 brd ff:ff:ff:ff:ff:ff
    altnet enp2s1
        inet 192.168.106.132/24 brd 192.168.106.255 scope global dynamic noprefixroute ens33
            valid_lft 1447sec preferred_lft 1447sec
        inet 192.168.10.2/24 scope global ens33
            valid_lft forever preferred_lft forever
        inet6 fe80::20c:29ff:febb:4018/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
user@Mini-labo-VM2:~$ █
```

3- Test de la connectivité :

Vérifions que les deux machines virtuelles peuvent bien communiquer entre elles.

Pour cela, nous allons utiliser **ping** sur chacune des machines avec l'adresse IP de l'autre.

VM1 :

```
user@Mini-labo-VM1:~$ ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.889 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.558 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.686 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.928 ms
64 bytes from 192.168.10.2: icmp_seq=5 ttl=64 time=0.722 ms
64 bytes from 192.168.10.2: icmp_seq=6 ttl=64 time=0.656 ms
^C
--- 192.168.10.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5071ms
rtt min/avg/max/mdev = 0.558/0.739/0.928/0.129 ms
user@Mini-labo-VM1:~$ █
```

VM2 :

```
user@Mini-labo-VM2:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.665 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.732 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.610 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.621 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=0.633 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=0.615 ms
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=0.576 ms
64 bytes from 192.168.10.1: icmp_seq=8 ttl=64 time=0.633 ms
^C
--- 192.168.10.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7164ms
rtt min/avg/max/mdev = 0.576/0.635/0.732/0.043 ms
user@Mini-labo-VM2:~$
```

4- Utilisation d'outils réseau :

a- Traceroute :

Commençons par tester l'outil **traceroute** sur chacune des machines. Cet outil va permettre de voir le chemin des paquets.

VM1 :

```
user@Mini-labo-VM1:~$ traceroute 192.168.10.2
traceroute to 192.168.10.2 (192.168.10.2), 30 hops max, 60 byte packets
 1  192.168.10.2 (192.168.10.2)  0.573 ms  0.489 ms  0.336 ms
```

VM2 :

```
user@Mini-labo-VM2:~$ traceroute 192.168.10.1
traceroute to 192.168.10.1 (192.168.10.1), 30 hops max, 60 byte packets
 1  192.168.10.1 (192.168.10.1)  0.835 ms  0.805 ms  0.702 ms
```

b- Iperf3 :

Maintenant testons **iperf3**, cet outil va servir à tester la bande passante. Pour cela nous allons tester la bande passante en utilisant la VM2 comme cliente et la VM1 comme serveur.

Si le terminal affiche « commande introuvable » alors se référer à [Problèmes rencontrés et solutions : 1- Utilisation d'outils réseau : a- Iperf3](#).

VM1 :

```
user@Mini-labo-VM1:~$ iperf3 -s
-----
Server listening on 5201 (test #1)
-----
Accepted connection from 192.168.10.2, port 37886
[ 5] local 192.168.10.1 port 5201 connected to 192.168.10.2 port 37898
[ ID] Interval          Transfer     Bitrate
[ 5]  0.00-1.00   sec   369 MBytes  3.09 Gbits/sec
[ 5]  1.00-2.00   sec   328 MBytes  2.75 Gbits/sec
[ 5]  2.00-3.00   sec   331 MBytes  2.78 Gbits/sec
[ 5]  3.00-4.00   sec   191 MBytes  1.60 Gbits/sec
[ 5]  4.00-5.00   sec   89.7 MBytes  752 Mbites/sec
[ 5]  5.00-6.00   sec   43.9 MBytes  369 Mbites/sec
[ 5]  6.00-7.00   sec   44.3 MBytes  371 Mbites/sec
[ 5]  7.00-8.00   sec   33.5 MBytes  281 Mbites/sec
[ 5]  8.00-9.00   sec   35.5 MBytes  298 Mbites/sec
[ 5]  9.00-10.00  sec   36.0 MBytes  302 Mbites/sec
[ 5] 10.00-10.04  sec   1.24 MBytes  247 Mbites/sec
-----
[ ID] Interval          Transfer     Bitrate
[ 5]  0.00-10.04  sec   1.47 GBytes  1.26 Gbits/sec
                                                receiver
-----
Server listening on 5201 (test #2)
```

Le -s est pour l'option serveur, cela indique que l'on considère cette VM comme notre serveur.

VM2 :

```
user@Mini-labo-VM2:~$ iperf3 -c 192.168.10.1
Connecting to host 192.168.10.1, port 5201
[ 5] local 192.168.10.2 port 37898 connected to 192.168.10.1 port 5201
[ ID] Interval          Transfer     Bitrate      Retr  Cwnd
[ 5]  0.00-1.00   sec   369 MBytes  3.10 Gbits/sec   42  2.17 MBytes
[ 5]  1.00-2.00   sec   328 MBytes  2.75 Gbits/sec  272  1.65 MBytes
[ 5]  2.00-3.00   sec   335 MBytes  2.80 Gbits/sec   0  1.74 MBytes
[ 5]  3.00-4.02   sec   190 MBytes  1.58 Gbits/sec   4  1.27 MBytes
[ 5]  4.02-5.00   sec   90.0 MBytes  764 Mbites/sec   0  1.35 MBytes
[ 5]  5.00-6.00   sec   43.8 MBytes  367 Mbites/sec   0  1.41 MBytes
[ 5]  6.00-7.00   sec   41.2 MBytes  347 Mbites/sec   0  1.43 MBytes
[ 5]  7.00-8.01   sec   36.2 MBytes  302 Mbites/sec   0  1.46 MBytes
[ 5]  8.01-9.00   sec   35.0 MBytes  295 Mbites/sec   0  1.49 MBytes
[ 5]  9.00-10.00  sec   36.2 MBytes  304 Mbites/sec   0  1.50 MBytes
-----
[ ID] Interval          Transfer     Bitrate      Retr
[ 5]  0.00-10.00  sec   1.47 GBytes  1.26 Gbits/sec  318
[ 5]  0.00-10.04  sec   1.47 GBytes  1.26 Gbits/sec
                                                sender
                                                receiver
iperf Done.
```

Le -c représente l'option client, donc que cette VM est notre client.

c- Netcat :

Avec **netcat** on va pouvoir simuler une communication entre les deux VM. On va afficher un message sur la VM2.

Le message sera : « Test communication »

VM1 :

```
user@Mini-labo-VM1:~$ nc 192.168.10.2 1234
```

```
Test communication
```

1234 correspond au port que l'on va utiliser, le port doit être le même sur les deux machines.

On écrit notre message sur la VM1.

VM2 :

```
user@Mini-labo-VM2:~$ nc -l -p 1234
```

```
Test communication
```

Le -l correspond à la fonction listen, cela va permettre de passer en mode écoute. On transforme notre VM en serveur qui attend une connexion.

Le -p indique la fonction port, ici cela va permettre d'identifier le port que l'on va devoir écouter.

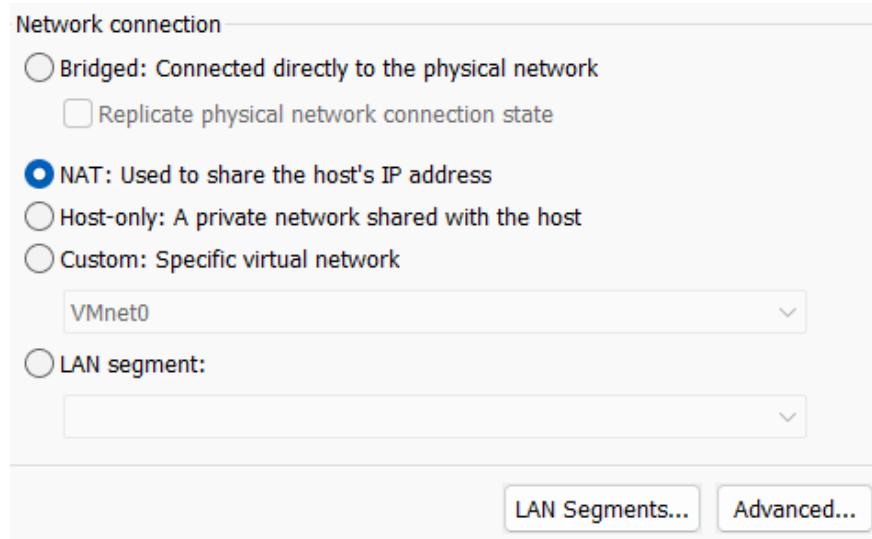
Problèmes rencontrés et solutions :

1- Utilisation d'outils réseau :

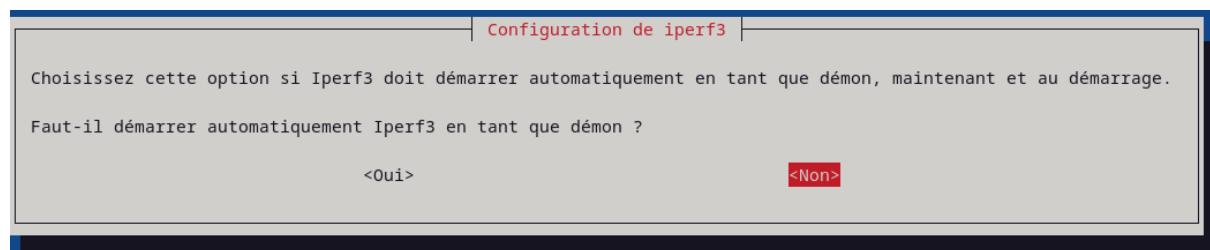
a- Iperf3 :

Si la commande est introuvable, alors il faut l'installer sur les deux VM.

On doit d'abord connecter nos VM en NAT pour pouvoir accéder à Internet afin de télécharger **Iperf3**.



```
user@Mini-labo-VM2:~$ sudo apt update
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
user@Mini-labo-VM2:~$ sudo apt install iperf3 -y
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 :
  libiperf0 libsctp1
```



Puis on repasse en LAN segment 1.

Conclusion et apprentissage :

1- Conclusion :

La mise en place de ce mini-labo réseau a permis de créer un environnement isolé et contrôlé composé de plusieurs machines virtuelles interconnectées.

Ces machines ont été configurées avec des adresses IP statiques et, ainsi, elles ont pu communiquer entre elles grâce aux tests de connectivité ([ping](#)).

Les outils de diagnostic tels que [traceroute](#), [iperf3](#) et [netcat](#) ont permis d'analyser le cheminement des paquets, de prendre les mesures de la bande passante et de simuler des communications réseau entre les deux machines.

Avec ce projet, on démontre la possibilité de reproduire une petite topologie réseau que l'on peut utiliser pour des expérimentations en sécurité, en administration système et en protocole réseau. Tout cela sur un ordinateur personnel.

2- Apprentissage :

Voici ce que l'on peut retenir de ce projet :

- La création d'un réseau local isolé à l'aide de machines virtuelles.
- La configuration manuelle des adresses IP statiques et vérifier leur cohérence.
- Le test de la connectivité réseau et diagnostiquer d'éventuels problèmes.
- L'utilisation des outils réseau essentiels ([ping](#), [traceroute](#), [netcat](#), [iperf3](#)).