



4. Observez la détection de la nouvelle carte avec la commande `lspci` et `ip` :

```
lspci | fgrep Ether      # détecter le périphérique
ip link                  # périphérique connu par la couche IP ?
ip addr                  # a-t-il une adresse ? (normalement non)
```

5. Configurez cette interface avec la commande `nmtui-edit` (**Network Manager Text User Interface**) et les paramètres :

- Nom : `enp0s8`
- Périphérique : `enp0s8`
- Configuration IPv4 : `manuel`
- Adresse IP : `192.168.0.10/24` (trois premiers octets pour le réseau)
- Passerelle : **rien**
- Serveur DNS : **rien**

6. Vérifiez la configuration (et redémarrez le réseau si nécessaire) :

```
ip addr
systemctl restart network      # si nécessaire
```

### 3 Brancher le client C1 sur le réseau privé

**Objectif** : Ajouter une nouvelle machine virtuelle qui va se connecter via le réseau privé virtuelle. Nous l'appellerons `C1`.

1. Modifiez la configuration réseau de `C1` pour que la carte soit branchée sur `intnet` (**important !**).
2. Faites démarrer `C1` et observez la détection de la carte réseau avec `ip link`.
3. À ce stade, le réseau ne doit pas fonctionner.
4. Fixez la configuration :

```
ip addr add 192.168.0.20/24 dev enp0s3      # adresse IP
echo "nameserver 10.0.2.3" > /etc/resolv.conf # DNS
ping -c 2 192.168.0.10                       # accès à la VM depuis C1
ip route add default via 192.168.0.10      # choisir VM comme routeur
```

5. À ce stade, le routage depuis `C1` ne fonctionne pas car la politique de routage n'est pas activée sur `VM`.

### 4 Transformer VM en routeur

1. Il nous reste à transformer `VM` en routeur faisant du NAT (translation d'adresses). Activez la fonction de routage des paquets du noyau Linux :

```
sysctl -w net.ipv4.ip_forward=1             # immédiatement
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf # pour pérenniser
```

2. Commencez par arrêter le **firewall** :

```
systemctl stop firewalld
systemctl disable firewalld
```

3. Activez ensuite les `iptables` :

```
dnf -y install iptables-services
systemctl enable iptables
```

4. En utilisant la chaîne `POSTROUTING` de la table `nat` faites en sorte que `VM` devienne un routeur qui accepte la translation d'adresses (cible `SNAT`) mais uniquement pour les machines virtuelles du réseau privée.

```
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

5. Sauvegardez la règle pour quelle soit appliquée au démarrage de `VM` :

```
/usr/libexec/iptables/iptables.init save
```

6. À ce stade, `C1` doit pleinement avoir accès au réseau (via le routeur `VM` qui réalise la translation des adresses). Provoquez-en pour charger les mises-à-jour :

```
yum -y update
```

## 5 Installez le serveur DNSMASQ

**Objectif** : Nos machines (serveurs et clients) ne sont évidemment pas officielles ni accessibles depuis l'internet. Il est néanmoins intéressant de prévoir un domaine DNS afin de donner un nom à ces machines et faciliter l'utilisation. Pour ce faire, nous allons déployer un serveur DNS cache qui assure ce service en prenant ces informations dans le fichier `/etc/hosts`.

1. Installez sur `VM` le serveur DNSMASQ :

```
dnf -y install dnsmasq
```

2. avec les commandes ci-dessous, ajoutez ces lignes au fichier `/etc/hosts` :

```
# enlever les anciennes lignes
sed -i -e '/idl.fr/d' -e '/^$/d' /etc/hosts

cat <<FIN >> /etc/hosts

192.168.0.10   srv.idl.fr
192.168.0.100 client0.idl.fr
192.168.0.101 client1.idl.fr
192.168.0.102 client2.idl.fr
192.168.0.103 client3.idl.fr

FIN
```

3. Lancez le service et observez les traces :

```
systemctl start dnsmasq
systemctl enable dnsmasq
systemctl status dnsmasq
```

4. Vous pouvez maintenant vérifier que la résolution des noms est bien opérationnelle :

```
# adresse officielle
dig @127.0.0.1 www.google.fr

# adresse privée
dig @127.0.0.1 srv.idl.fr

# reverse privée
dig @127.0.0.1 -x 192.168.0.102
```

## 6 Utilisez le serveur DHCP

**Objectif** : Utiliser un serveur DHCP sur **VM** pour distribuer les adresses aux machines virtuelles (notamment **C1**).

1. Installez sur **VM** le serveur DHCP :

```
dnf -y install dhcp-server
```

2. avec les commandes ci-dessous, ajoutez ces lignes dans le fichier **/etc/dhcp/dhcpd.conf** :

```
cat <<EOF >> /etc/dhcp/dhcpd.conf

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.200;
    option domain-name-servers 192.168.0.10;
    option domain-name "idl.fr";
    option subnet-mask 255.255.255.0;
    option routers 192.168.0.10;
    default-lease-time 600;
    max-lease-time 7200;
}

EOF
```

3. Lancez le service et observez les traces :

```
systemctl start dhcpd
systemctl enable dhcpd
systemctl show dhcpd
```

## 7 Vérifier le poste client

- Après redémarrage, **C1** doit maintenant obtenir son adresse à partir de **VM** par DHCP. Vous pouvez voir les traces de la demande sur **VM** grâce à la commande **tail**. Vous pouvez aussi faire

```
# sur srv
systemctl status dhcpd
```

- Toujours sur **C1** vous pouvez vérifier la bonne résolution des noms

```
# pour connaitre les noms
hostname
hostnamectl

# pour vérifier les paramètres du client DNS (le resolver)
cat /etc/resolv.conf

# pour tester la résolution DNS
dig www.google.fr

# pour tester le réseau
curl http://www.google.fr

ssh srv
```

- Clonez C1 en C2 (clones liés) et tentez le lancement de deux VM. Vérifiez la configuration réseau. Vérifiez notamment que vous pouvez de C1 passer à C2 et inversement.