

# Réseau sous Linux

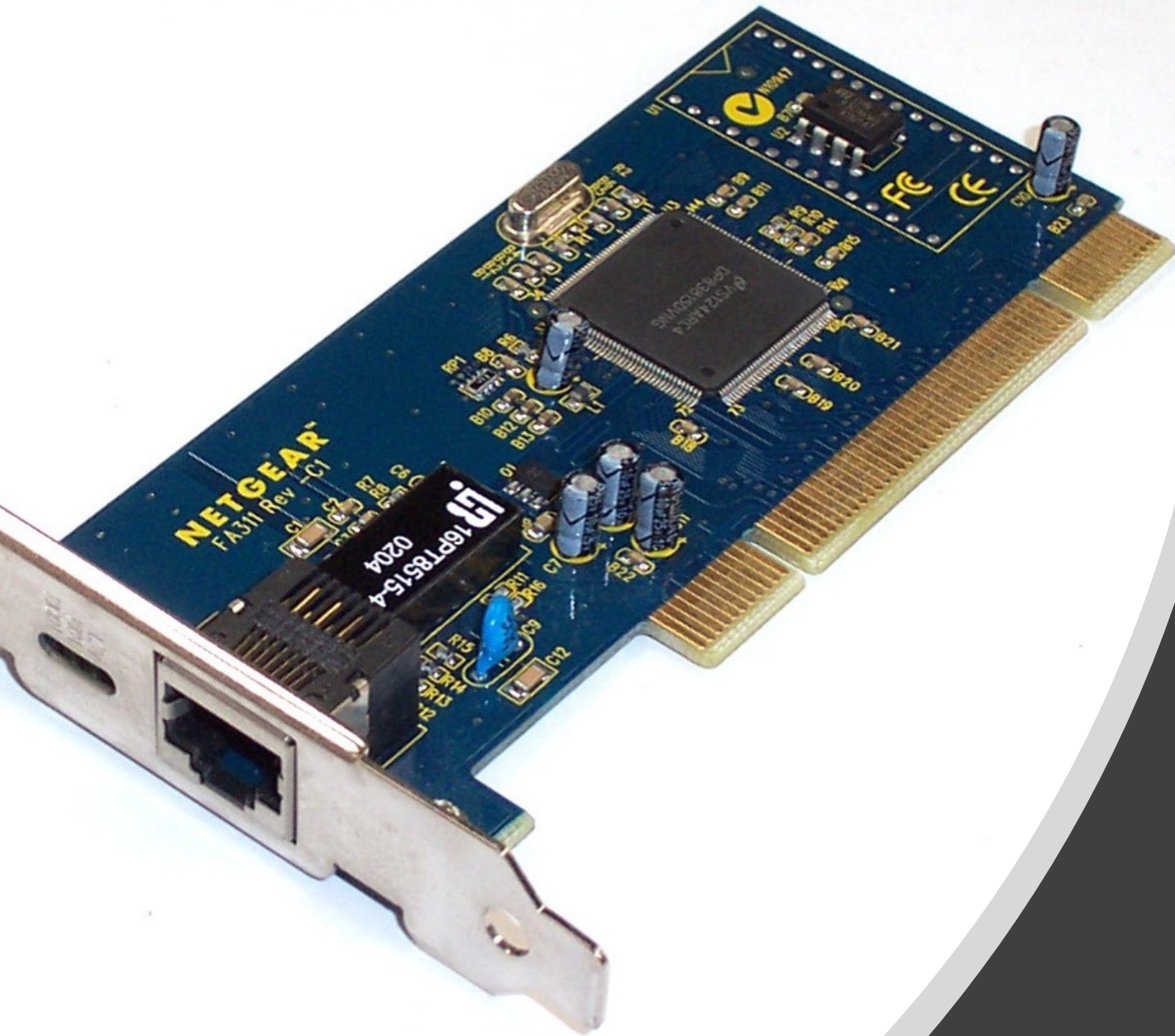
Ubuntu 22.04.1

# Règles IP

Les règles du protocole IP s'appliquent, peu importe le système d'exploitation! C'est ce qui fait en sorte que les périphériques utilisant différents SE peuvent communiquer entre eux s'ils utilisent le même protocole de communication.

Rappel des conditions du protocole IP :

1. S'identifier de façon unique
2. Être sur le même réseau IP
3. Si les ordinateurs ne sont pas sur le même réseau IP, il faut utiliser une passerelle



# Synonymes

Carte réseau

Adapteur réseau

Interface réseau

Network card

Network adapter

Network interface

Network device (dev)

# Nom de l'interface réseau (carte physique)

```
vincent@vincent-virtual-machine:~/Bureau$ sudo lshw -class network
[sudo] Mot de passe de vincent :
*-network
   description: Ethernet interface
   produit: VMXNET3 Ethernet Controller
   fabricant: VMware
   identifiant matériel: 0
   information bus: pci@0000:03:00.0
   nom logique: ens160
   version: 01
   numéro de série: 00:50:56:ad:f0:20
   taille: 10Gbit/s
   capacité: 10Gbit/s
   bits: 32 bits
   horloge: 33MHz
   fonctionnalités: pm pciexpress msi msix bus_master cap_list rom ethernet physical logical tp 1000bt-fd 10000bt-fd
   configuration : autonegotiation=off broadcast=yes driver=vmxnet3 driverversion=1.6.0.0-k-NAPI duplex=full ip=172.31.7.232 latency=0
   port=twisted pair speed=10Gbit/s
   ressources : irq:18 mémoire:fd4fc000-fd4fcfff mémoire:fd4fd000-fd4fdfff mémoire:fd4fe000-fd4fffff portE/S:4000(taille=16) mémoire
```

**sudo lshw -class network**

Modèle de la carte réseau = VMXNET3 de VMware

Nom de l'interface réseau (carte/adapteur) = ens160

# Service réseau (« renderer »)

- Sous Linux, il existe plusieurs services permettant de gérer l'accès au réseau.
- Le service utilisé varie en fonction de la distribution Linux utilisée, et de la version
- Pour Ubuntu, l'un des deux services suivants peut être utilisé :
  1. NetworkManager (attention aux majuscules!)
  2. systemd-networkd
- NetworkManager supporte les connexions Wifi et les commandes « **nm**tui » et « **nm**cli ». Il est conçu pour détecter un changement de topologie et réagir automatiquement (ex. un usager débranche son laptop du réseau filaire, NetowrkManager va automatiquement tenter d'utiliser une connexion Wifi pour ne pas que l'utilisateur perde accès au réseau)
- Pour Ubuntu Desktop, le service NetworkManager est utilisé, alors que pour Ubuntu Serveur, c'est systemd-networkd (un serveur n'utilise pas le wifi d'habitude)

# Service réseau

Comment identifier le service utilisé par votre Linux ?

```
vincent@vincent-virtual-machine:~/Bureau$ systemctl status systemd-networkd
○ systemd-networkd.service - Network Configuration
   Loaded: loaded (/lib/systemd/system/systemd-networkd.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
TriggeredBy: ○ systemd-networkd.socket
   Docs: man:systemd-networkd.service(8)
vincent@vincent-virtual-machine:~/Bureau$ systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-09-19 13:41:39 EDT; 1h 28min ago
   Docs: man:NetworkManager(8)
  Main PID: 821 (NetworkManager)
    Tasks: 3 (limit: 4625)
   Memory: 10.7M
      CPU: 4.500s
   CGroup: /system.slice/NetworkManager.service
           └─821 /usr/sbin/NetworkManager --no-daemon
```

Le service en cours d'exécution est celui qui gère l'accès au réseau.

```
vincent@vincent:~$ lsb_release -a | grep -i desc
No LSB modules are available.
Description:    Ubuntu 22.04.1 LTS
vincent@vincent:~$ sudo systemctl status NetworkManager
Unit NetworkManager.service could not be found.
vincent@vincent:~$ sudo systemctl status systemd-networkd
● systemd-networkd.service - Network Configuration
   Loaded: loaded (/lib/systemd/system/systemd-networkd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-09-19 19:34:46 UTC; 5min ago
TriggeredBy: ● systemd-networkd.socket
   Docs: man:systemd-networkd.service(8)
  Main PID: 848 (systemd-networkd)
    Status: "Processing requests..."
    Tasks: 1 (limit: 4534)
   Memory: 2.8M
      CPU: 43ms
   CGroup: /system.slice/systemd-networkd.service
           └─848 /lib/systemd/systemd-networkd
```



# Pour connaître sa configuration IP actuelle

- `ip address` → `ip a`
- `ip route` → `ip r`
- `resolvectl status`



L'option « -c » permet de mettre de la couleur dans l'affichage de `ip` :

```
vincent@vincent-virtual-machine:~/Bureau$ ip -c 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
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:50:56:ad:f0:20 brd ff:ff:ff:ff:ff:ff
   altname enp3s0
   inet 10.10.10.2/24 brd 10.10.10.255 scope global noprefixroute ens160
       valid_lft forever preferred_lft forever
   inet6 fe80::250:56ff:fead:f020/64 scope link
       valid_lft forever preferred_lft forever
vincent@vincent-virtual-machine:~/Bureau$
```

```

vincent@vincent-virtual-machine:~/Bureau$ ip address
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: ens160 <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:ad:f0:20 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 172.31.7.232/16 brd 172.31.255.255 scope global dynamic noprefixroute ens160
        valid_lft 5277sec preferred_lft 5277sec
    inet6 fe80::7202:78ff:1a2a:36f4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
vincent@vincent-virtual-machine:~/Bureau$ ip route
default via 172.31.0.1 dev ens160 proto dhcp metric 100
169.254.0.0/16 dev ens160 scope link metric 1000
172.31.0.0/16 dev ens160 proto kernel scope link src 172.31.7.232 metric 100
vincent@vincent-virtual-machine:~/Bureau$ resolvectl status
Global
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    resolv.conf mode: stub

Link 2 ens160
    Current Scopes: DNS
    Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    Current DNS Server: 172.31.0.1
    DNS Servers: 172.31.0.1
    DNS Domain: labinfo.local
vincent@vincent-virtual-machine:~/Bureau$

```

Interface réseau = ens160

Adresse IPv4 = 172.31.7.232  
 Masque sous-réseau = /16 = 255.255.0.0  
 Légende:  
 /8 => 255.0.0.0  
 /16 => 255.255.0.0  
 /24 => 255.255.255.0

Adresse de la passerelle = 172.31.0.1

Adresse serveur DNS = 172.31.0.1



# Changer sa configuration IP

Configuration statique ou DHCP



ubuntu

# Sur Ubuntu, on utilise « netplan »

1. On identifie le fichier utilisé par netplan, situé dans le répertoire /etc/netplan

```
vincent@vincent-virtual-machine:~$ ls /etc/netplan/  
01-network-manager-all.yaml  
vincent@vincent-virtual-machine:~$
```

```
vincent@vincent:~$ ls /etc/netplan/  
00-installer-config.yaml  
vincent@vincent:~$ _
```

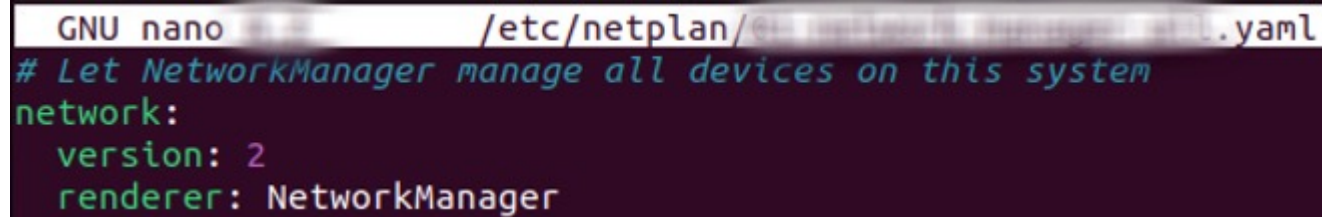
2. On édite le fichier .yaml en fonction de nos besoins (voir diapo suivante)  
Attention! **L'indentation** est très importante avec le format YAML

```
vincent@vincent-virtual-machine:~$ sudo nano /etc/netplan/01-network-manager-all.yaml
```

```
vincent@vincent:~$ sudo nano /etc/netplan/00-installer-config.yaml
```

3. On applique la nouvelle configuration avec la commande :  
**sudo netplan apply**

# Configuration DHCP (fichier yaml)

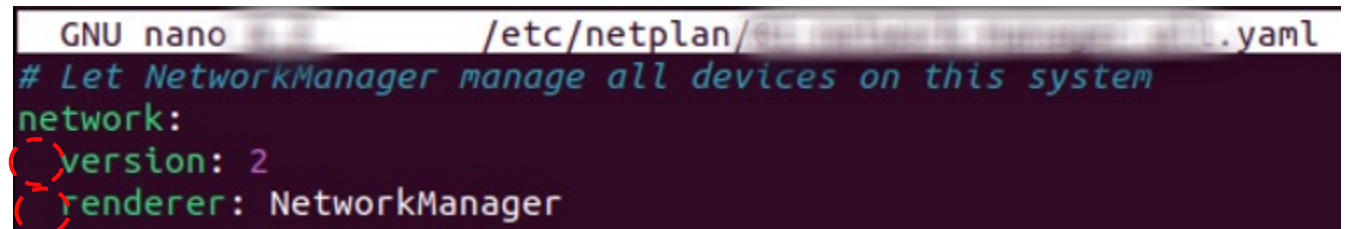


```
GNU nano /etc/netplan/01-netcfg.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

- Par défaut, Ubuntu utilisera le service systemd-networkd pour la gestion des configurations réseaux. Sur Ubuntu Desktop (avec interface graphique), la ligne « renderer » est ajoutée pour spécifier que c'est le service NetworkManager qui sera utilisé. Si la ligne « renderer » n'est pas dans le fichier, alors Ubuntu utilisera systemd-networkd.
- Une ligne qui débute par le signe « # » est considérée comme un commentaire. C'est-à-dire que cette ligne ne sera pas traitée par le système, et a donc AUCUN impact sur la configuration du réseau.
- L'indentation est primordiale dans le format yaml

2 espaces ou 1 tab

2 espaces ou 1 tab



```
GNU nano /etc/netplan/01-netcfg.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

# Configuration statique (fichier yaml)

```
GNU nano 2.9.3 /etc/netplan/01-netcfg.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens160:
      dhcp4: false
      addresses: [10.10.10.2/24]
      nameservers:
        addresses: [10.10.10.1, 1.1.1.1]
      routes:
        - to: default
          via: 10.10.10.1
```

Interface réseau

Adresse IPv4 = 10.10.10.2  
Masque sous-réseau = /24

Adresse serveur DNS = 10.10.10.1 et 1.1.1.1

Adresse de la passerelle = 10.10.10.1

Voir le fichier txt donné en exemple.



# Configuration statique (fichier yaml)

```
GNU nano 2.9.3 /etc/netplan/01-netcfg.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens160:
      dhcp4: false
      addresses: [10.10.10.2/24]
      nameservers:
        addresses: [10.10.10.1, 1.1.1.1]
      routes:
        - to: default
          via: 10.10.10.1
```

( ) 1 tab

( ) 2 tab

( ) 3 tab

( ) 4 tab

( ) 5 tab

**! indentation!**

# Anciennes pratiques / Autres distro Linux

- Dans les anciennes versions d'Ubuntu (<18.04), et dans d'autres distributions Linux (ex. Debian, Kali, RedHat, etc.), la façon de gérer la configuration réseau peut être différente.

Quelques exemples :

- Les commandes **ifconfig** et **route** du package **net-tools** étaient utilisées pour obtenir la config IP actuelle, elles ont été remplacées par **ip a** et **ip r**
- Les fichiers **/etc/network/interfaces** (Debian) ou **/etc/sysconfig/network-scripts/ifcfg-eth0** (Redhat) pour implémenter une configuration IP statique ou dynamique (DHCP)
- Le fichier **/etc/resolv.conf** pour configurer les serveur DNS

Il faut donc [lire la doc](#)umentation appropriée selon le Linux utilisé!