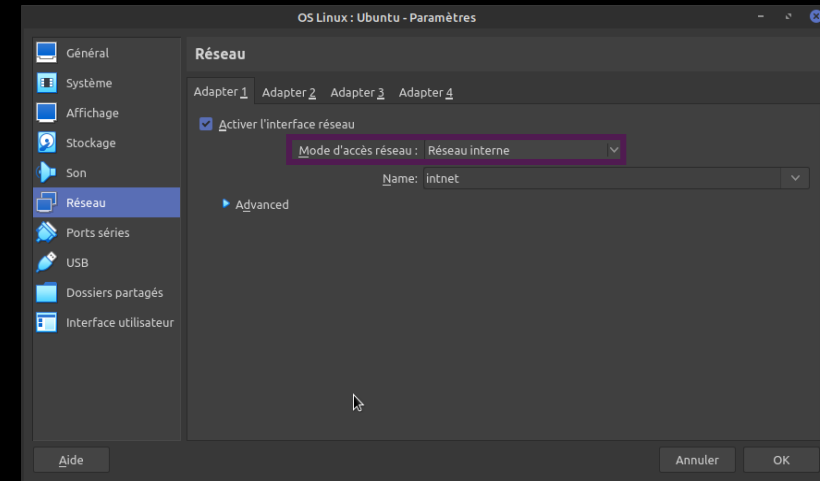
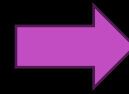
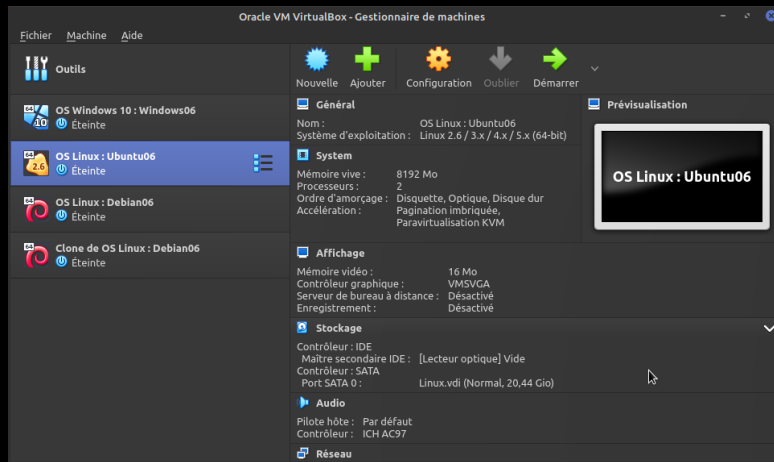


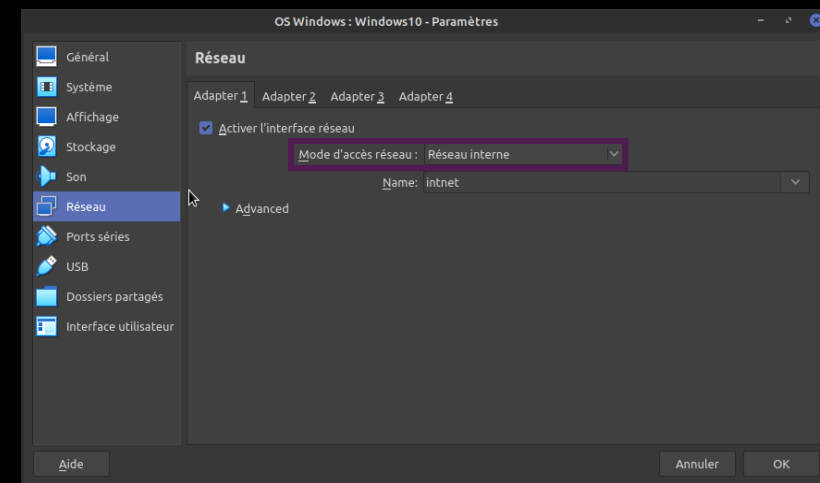
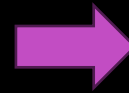
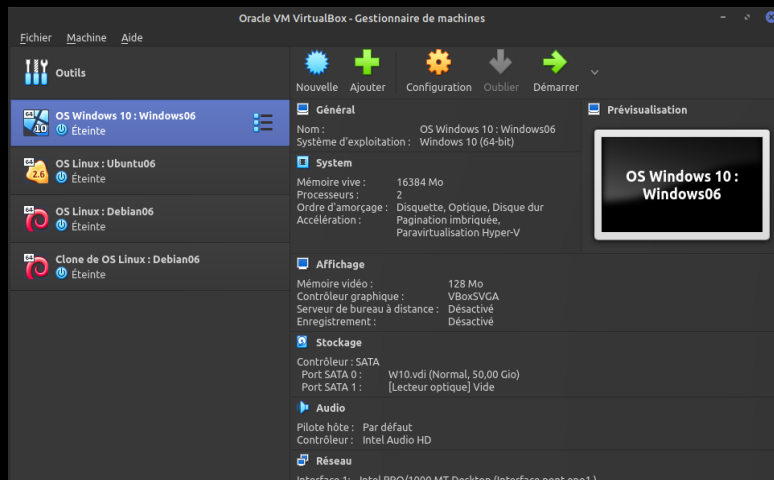
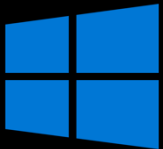
# TP - CRÉATION D'UN RÉSEAU

# Mise en place des trois systèmes d'exploitation sur VirtualBox

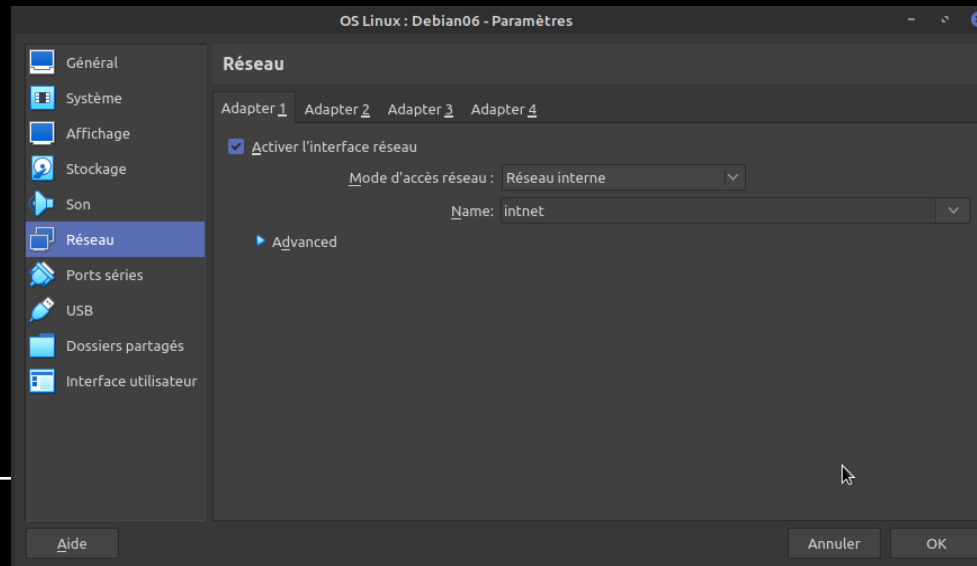
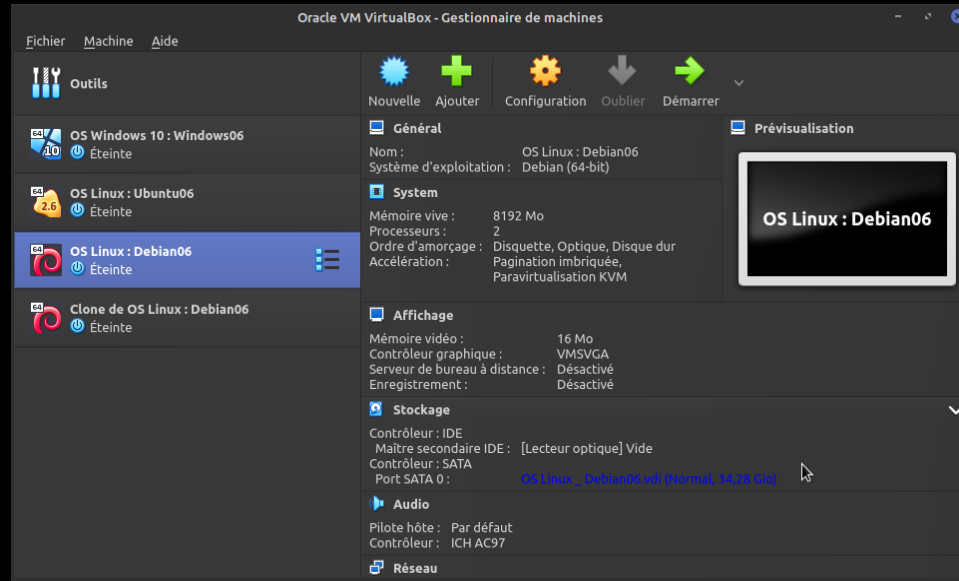
Ubuntu :



Windows :



## Debian 11 :

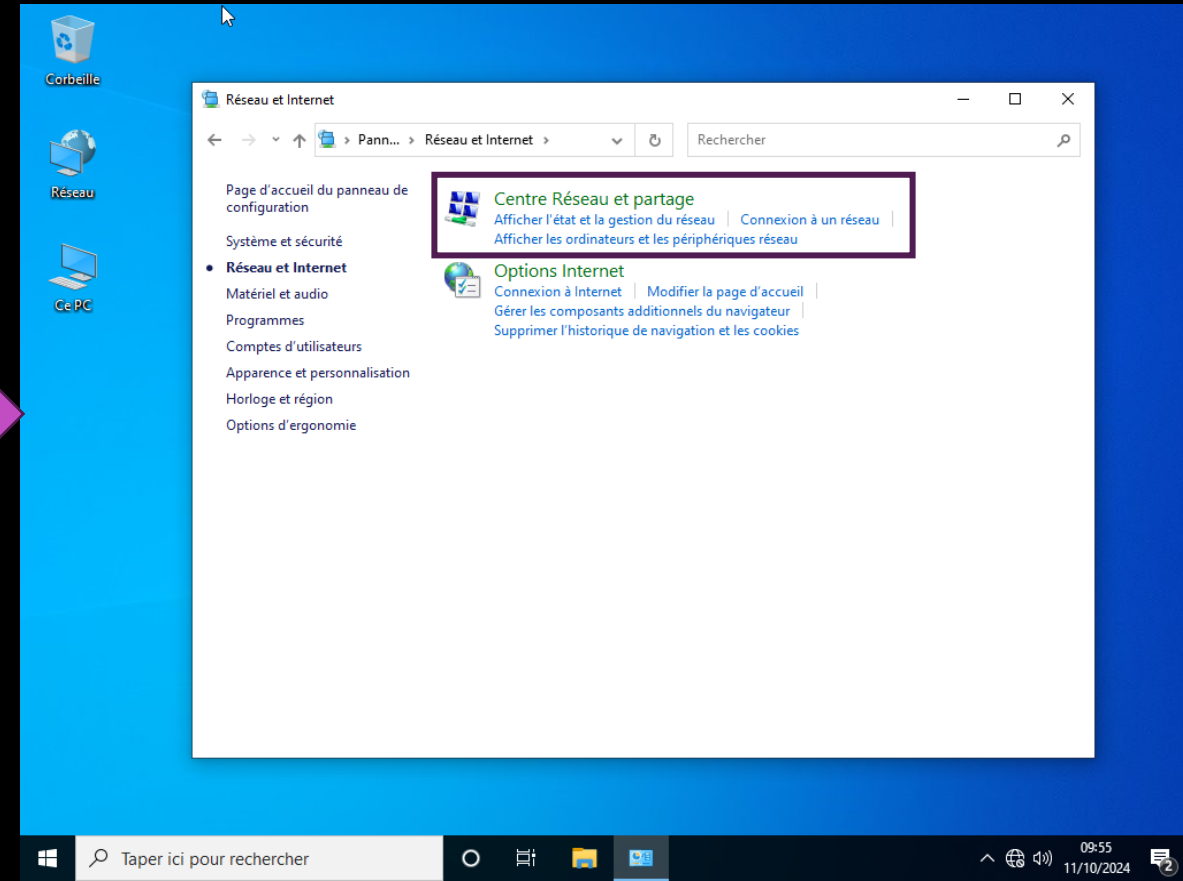
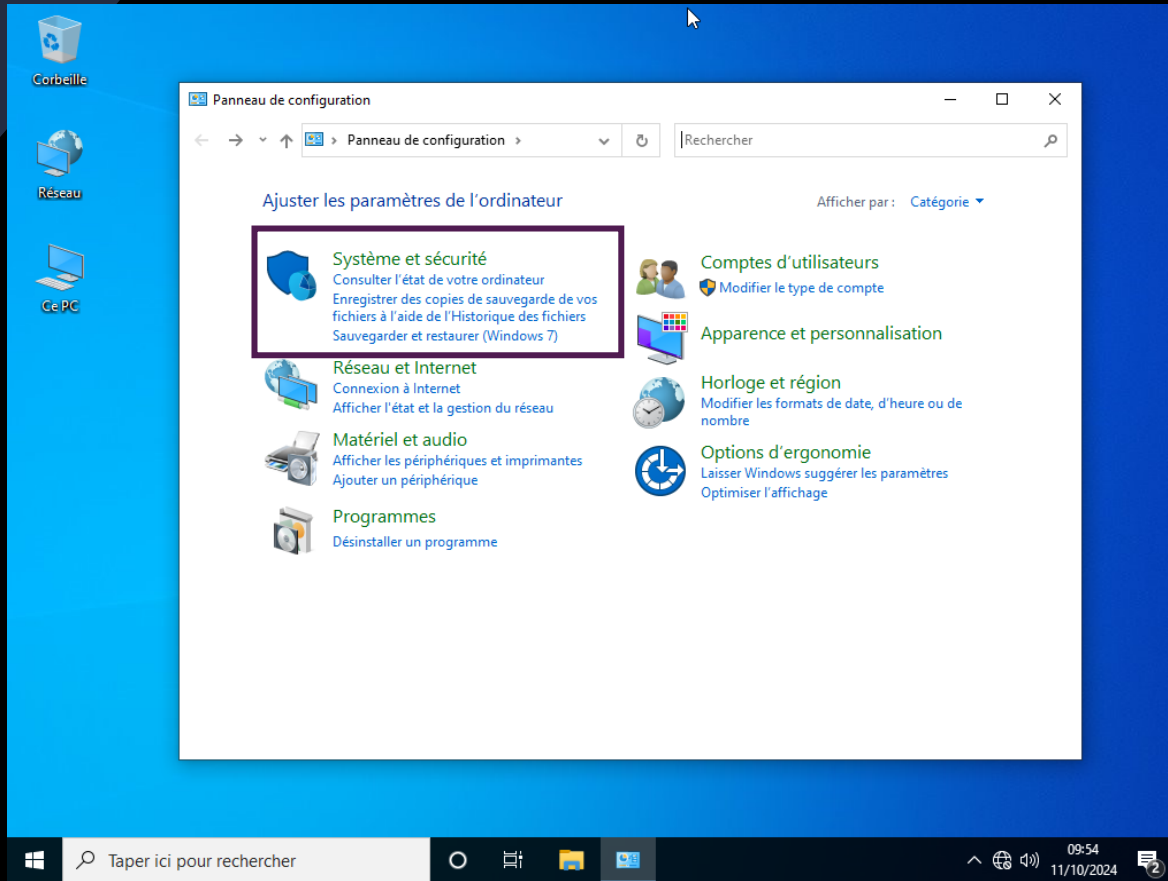


# Windows 10

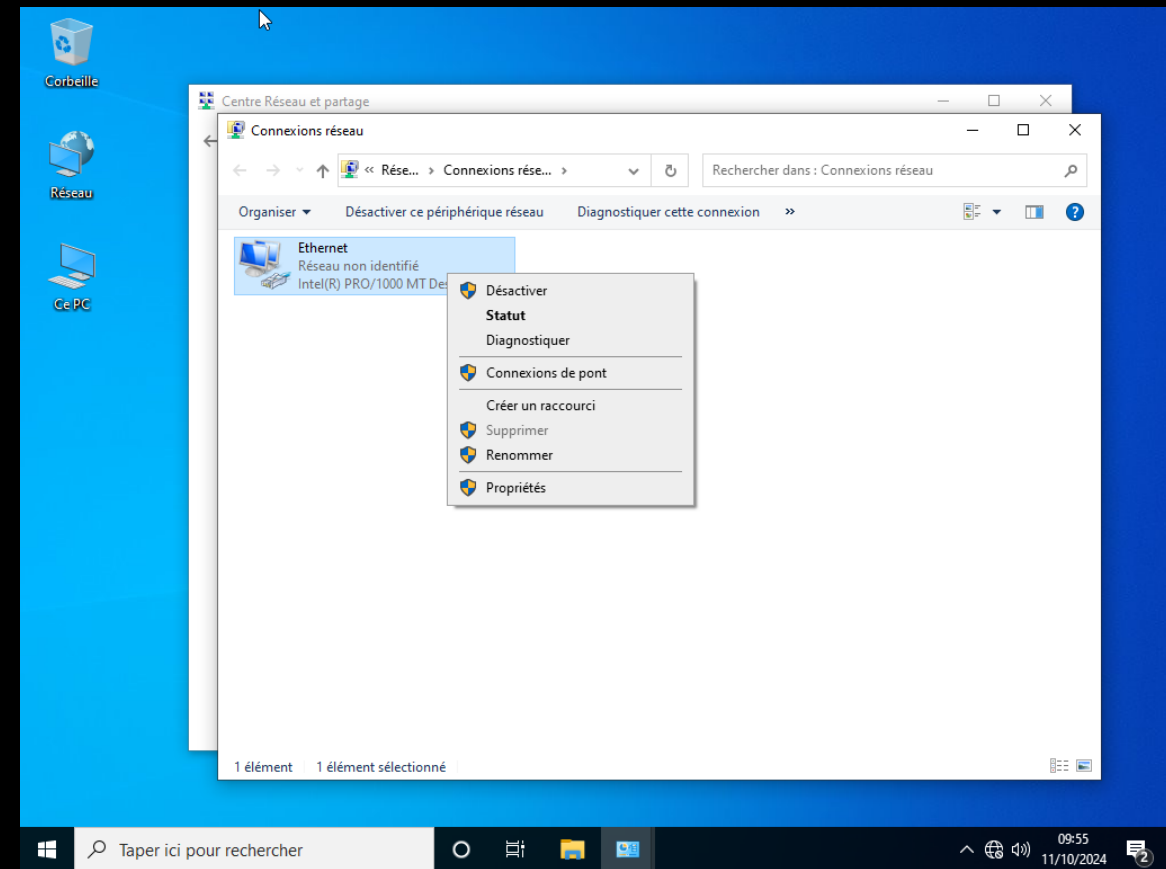
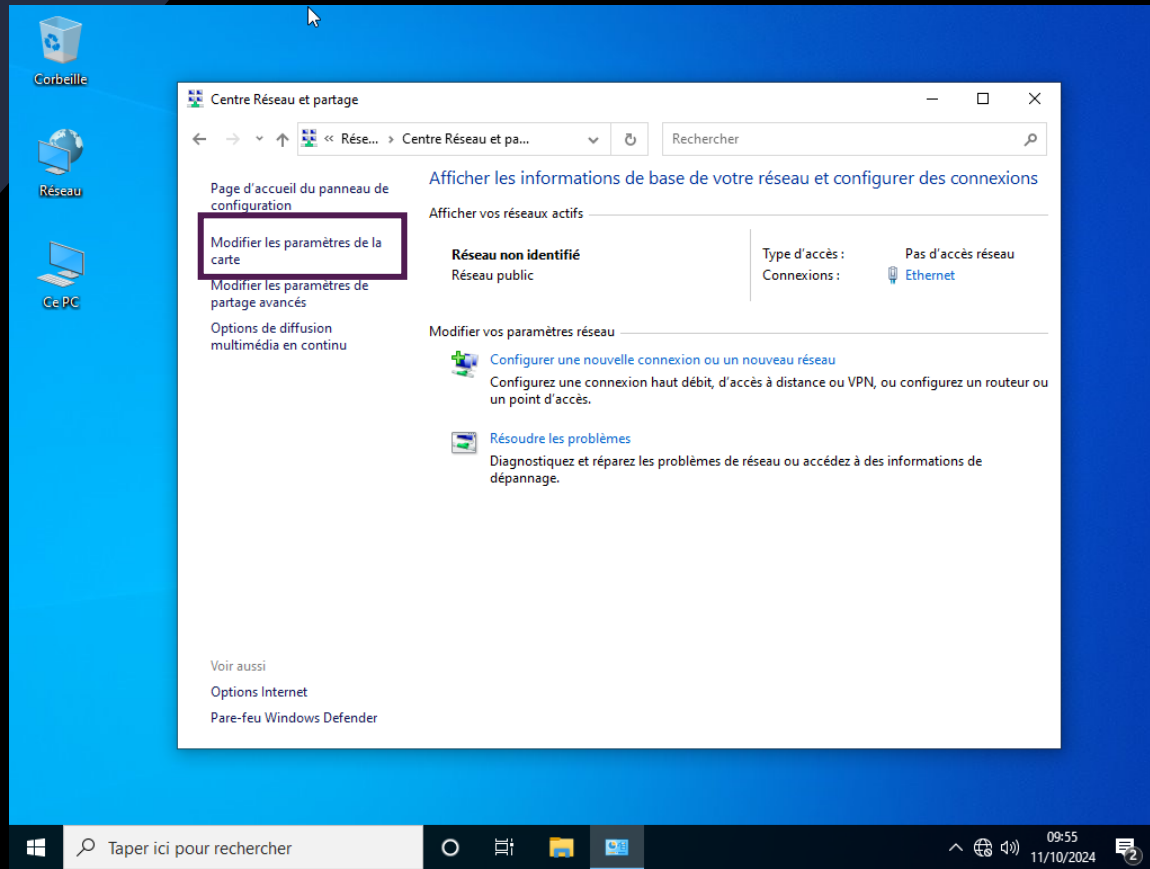


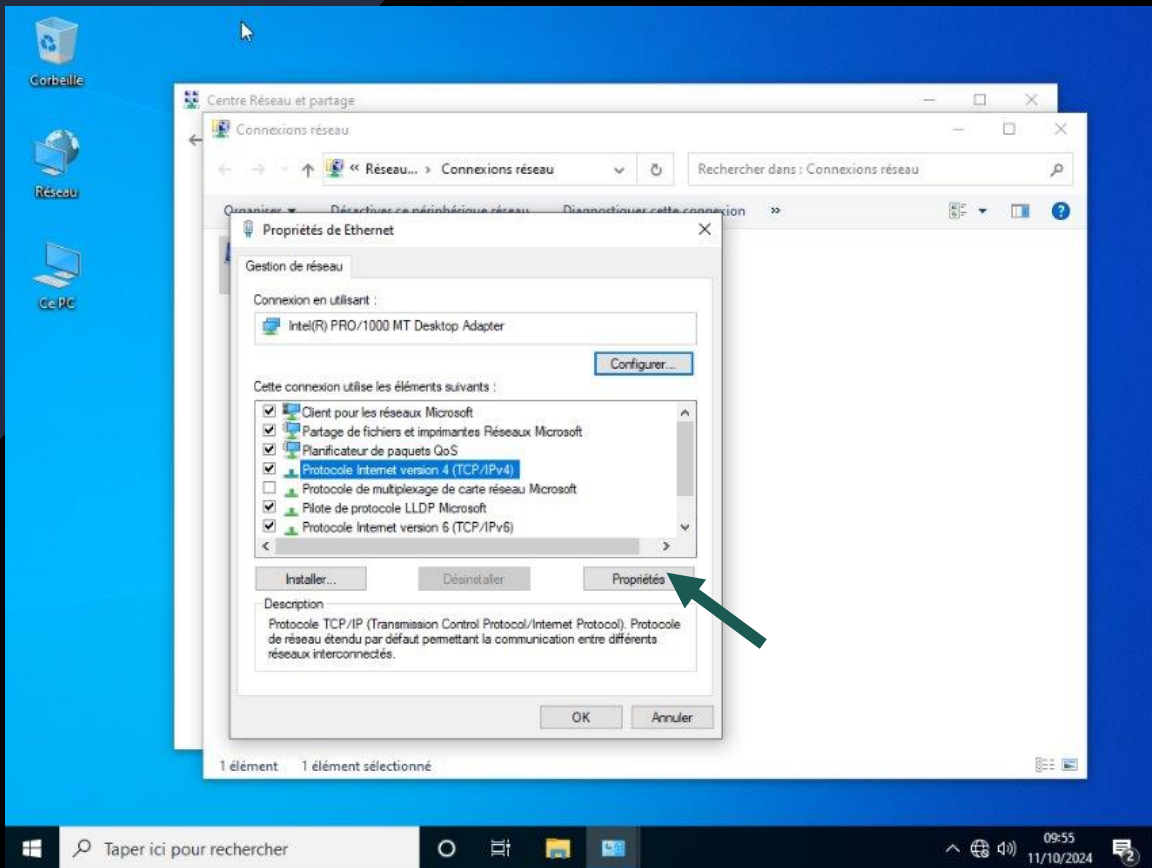
# Comment adresser une IP sur Windows 10 ?

Dans un premier temps, il faut accéder au **panneau de configuration** puis aller dans **"Système et sécurité"** et cliquer sur **"Centre Réseau et partage"**

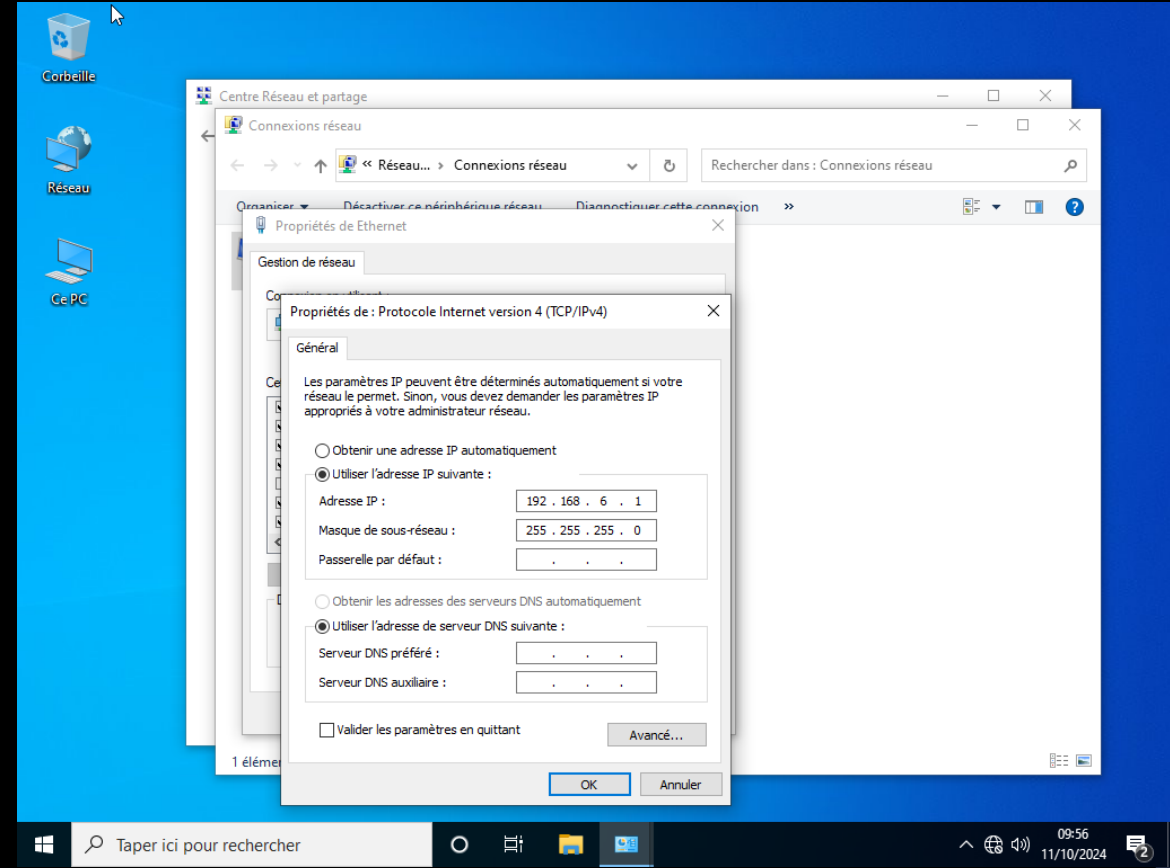


Ensuite, nous devons accéder aux propriétés de la carte Ethernet, nous irons dans "**Modifier les paramètres de la carte**", puis faites un clic droit sur "**Ethernet**" et enfin cliquer sur "**Propriétés**"





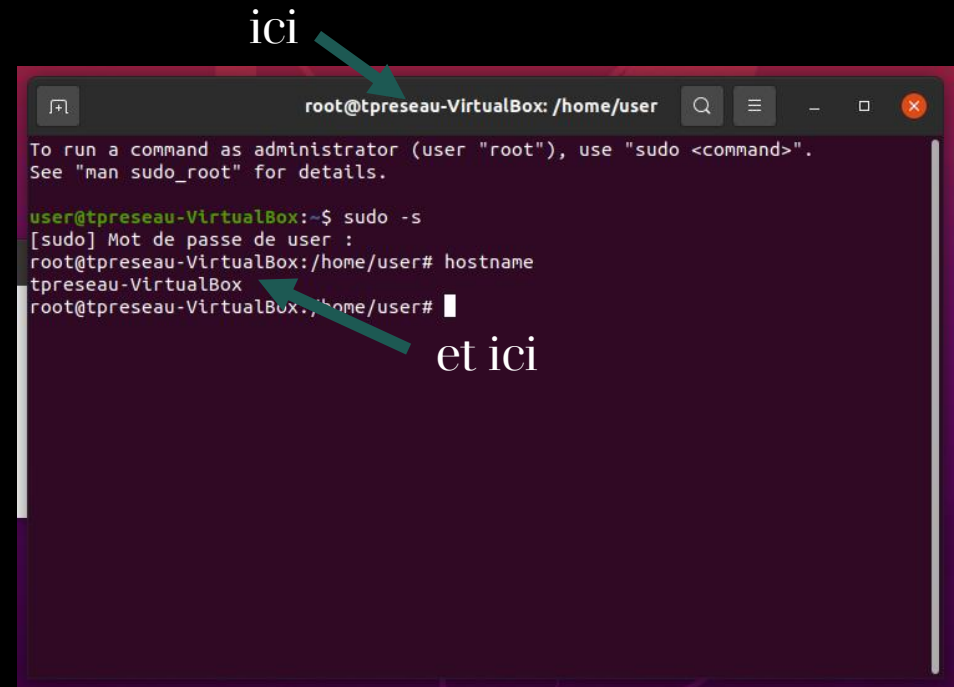
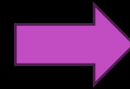
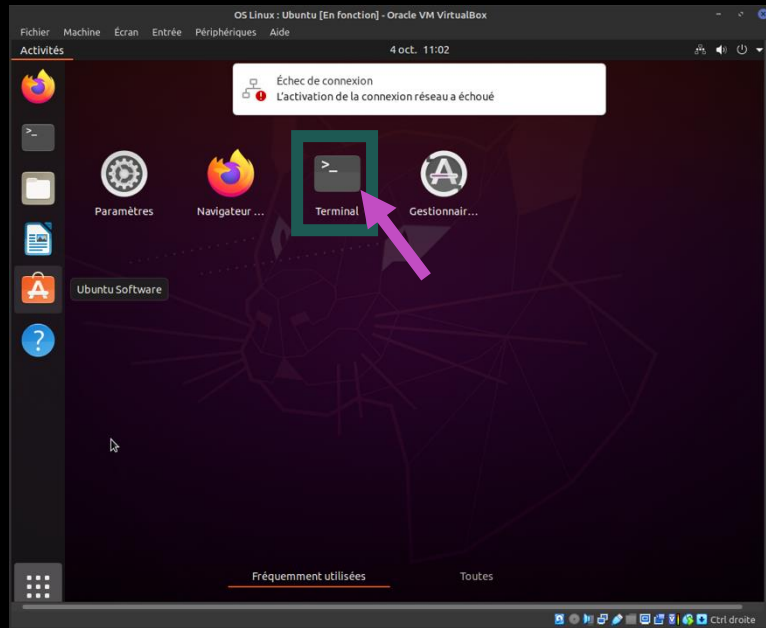
Sélectionner "**Protocole internet version 4 (TCP/IPv4)**" puis cliquer sur "**Propriétés**"



Dans les propriétés du protocole IPv4, nous allons rentrer l'adresse IP correspondante (192.168.6.1) et son masque "255.255.255.0" (classe C)



# Configuration de Linux depuis le terminal (Interface graphique)

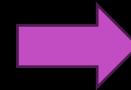


Nous devons passer en **super-utilisateur** pour avoir des permissions pour modifier le système. En rentrant la commande "sudo -s", puis le mot de passe attribué, nous pouvons désormais accéder complètement au système.

Avec la commande "hostname", celle-ci nous renseigne sur le nom de l'hôte. Il est visible directement après avoir rentré la commande et aussi visible dans l'encadrée de la fenêtre du terminal.

# Comment changer son nom d'hôte sur Linux Ubuntu ?

```
root@tpreseau-Virtualbox: /home/user
user@tpreseau-Virtualbox:~$ sudo -s
[sudo] Mot de passe de user :
root@tpreseau-Virtualbox:/home/user# hostname
tpreseau-Virtualbox
root@tpreseau-Virtualbox:/home/user# sudo nano /etc/hostname
```



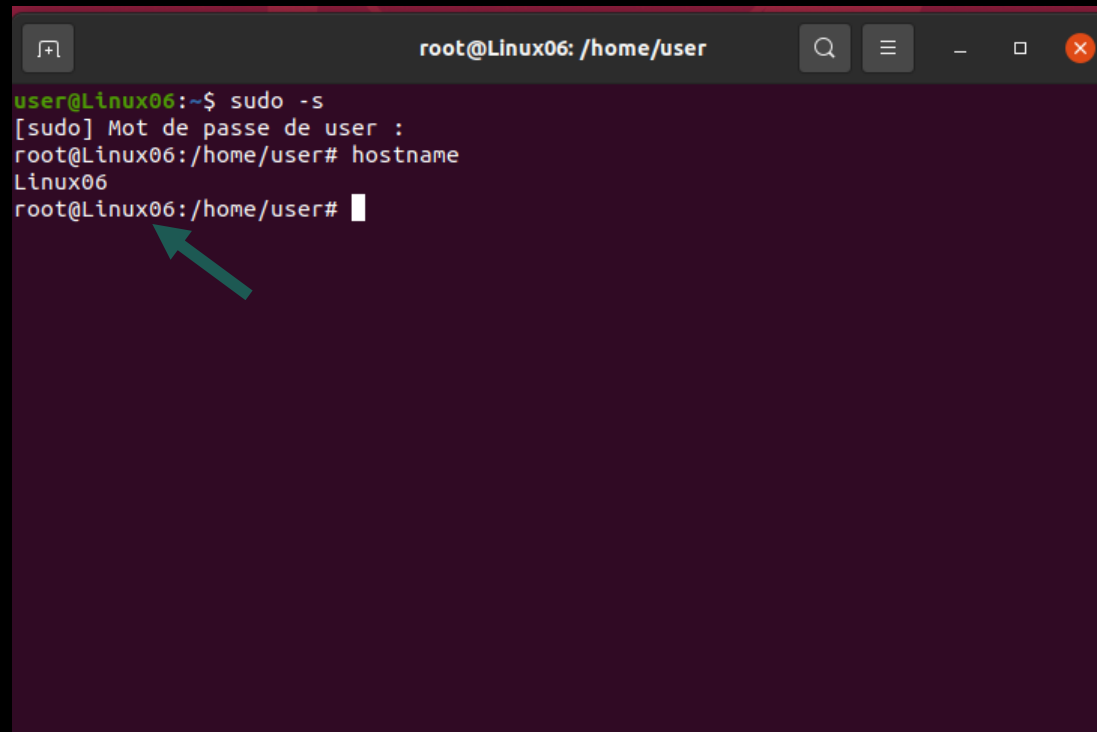
```
GNU nano 4.8 /etc/hostname Modifié
Linux06

^G Aide      ^O Écrire
^X Quitter   ^R Lire fich.
^W Chercher  ^M Remplacer
^K Couper    ^U Coller
^J Justifier ^T Orthograp.
```

Pour changer le nom d'hôte, il faut rentrer la commande "**sudo nano /etc/hostname**" dans le terminal.

Vous accéderez à ce répertoire où vous pourrez modifier directement le nom d'hôte. Il faudra enregistrer avec le raccourci "**ctrl+x**" et valider avec "**O**".

# Que se passe t'il si l'on reboot ?



```
root@Linux06: /home/user
user@Linux06:~$ sudo -s
[sudo] Mot de passe de user :
root@Linux06: /home/user# hostname
Linux06
root@Linux06: /home/user#
```

A terminal window with a dark purple background. The window title is "root@Linux06: /home/user". The terminal shows a user running "sudo -s", entering a password, and then running "hostname". The output of "hostname" is "Linux06". A green arrow points to the "Linux06" output.

Le nom d'hôte devient "Linux06" après le redémarrage

# Changer le nom d'hôte avec la commande "sudo nano /etc/hosts"

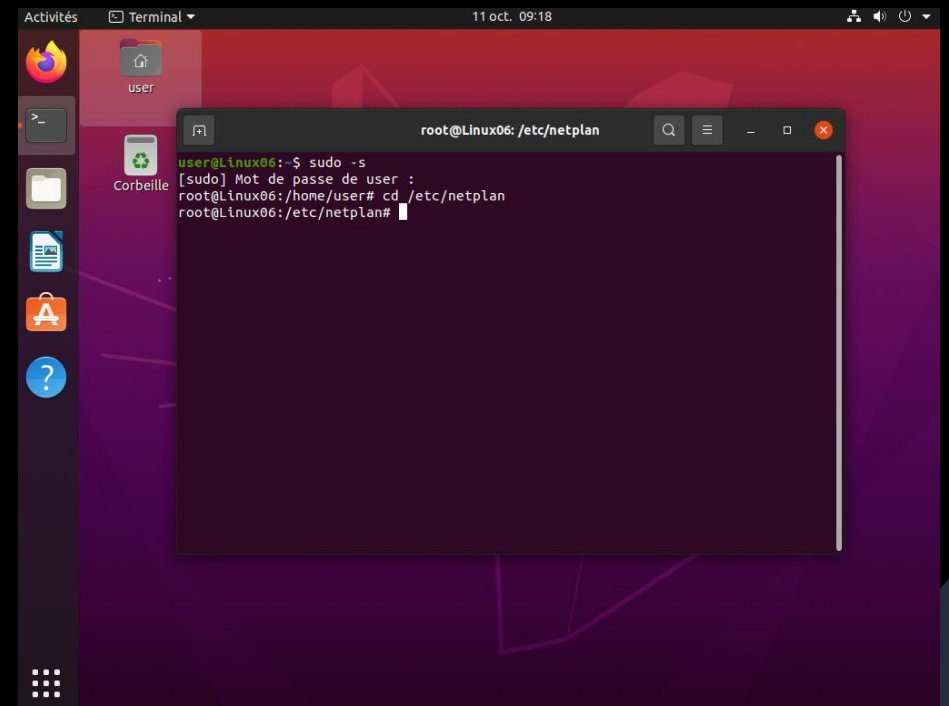
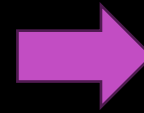
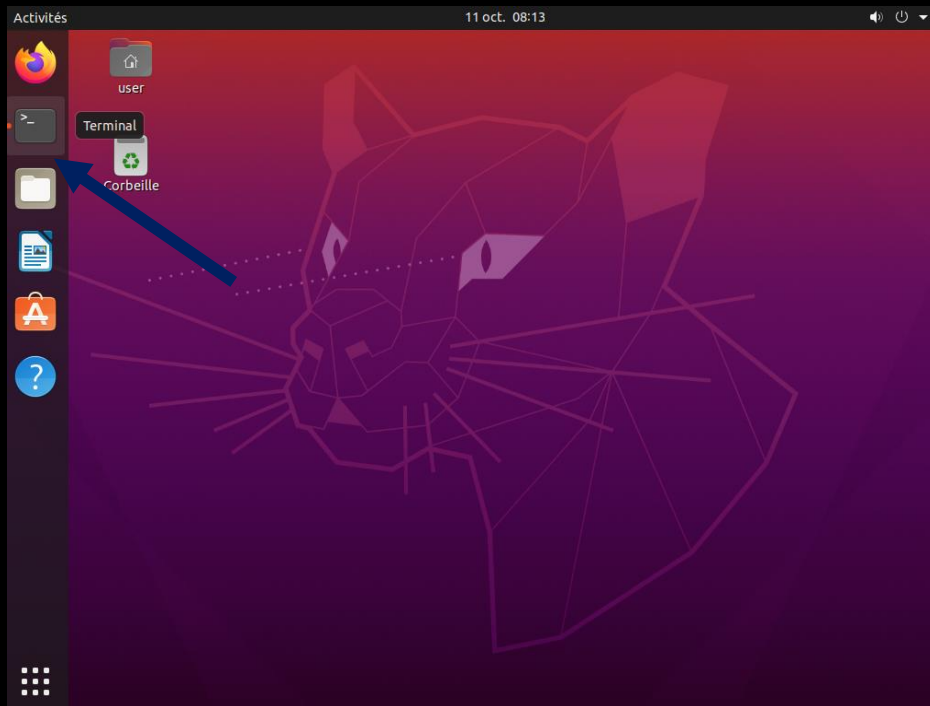
```
root@Linux06: /home/user
GNU nano 4.8 /etc/hosts Modifié
127.0.0.1 localhost
127.0.1.1 Linux06
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

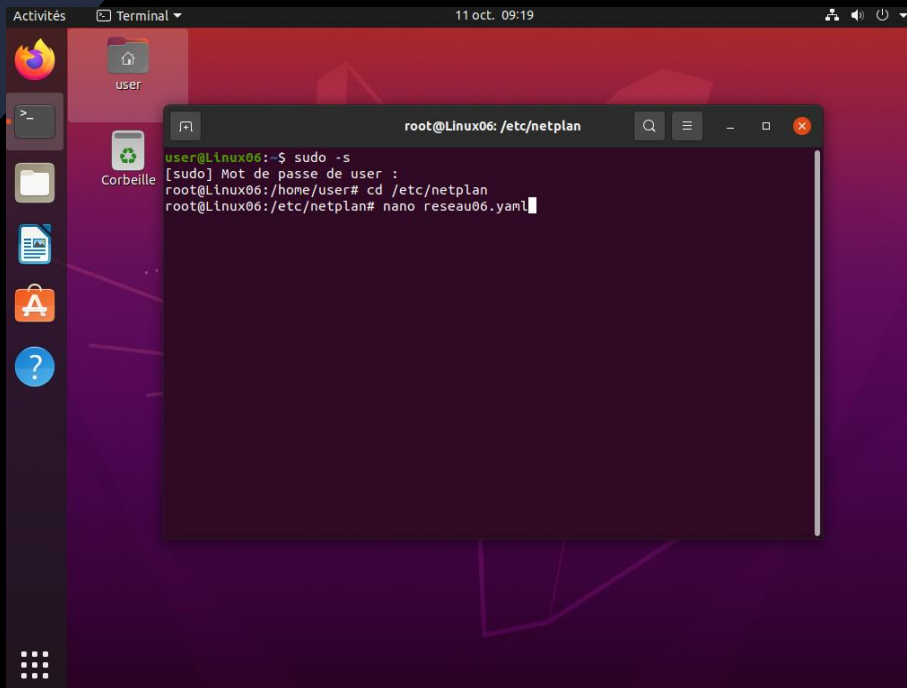
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^J Justifier ^C Pos. cur.
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^T Orthograp.^_ Aller ligne
```

# Configurer un adressage IP sur Ubuntu en interface graphique

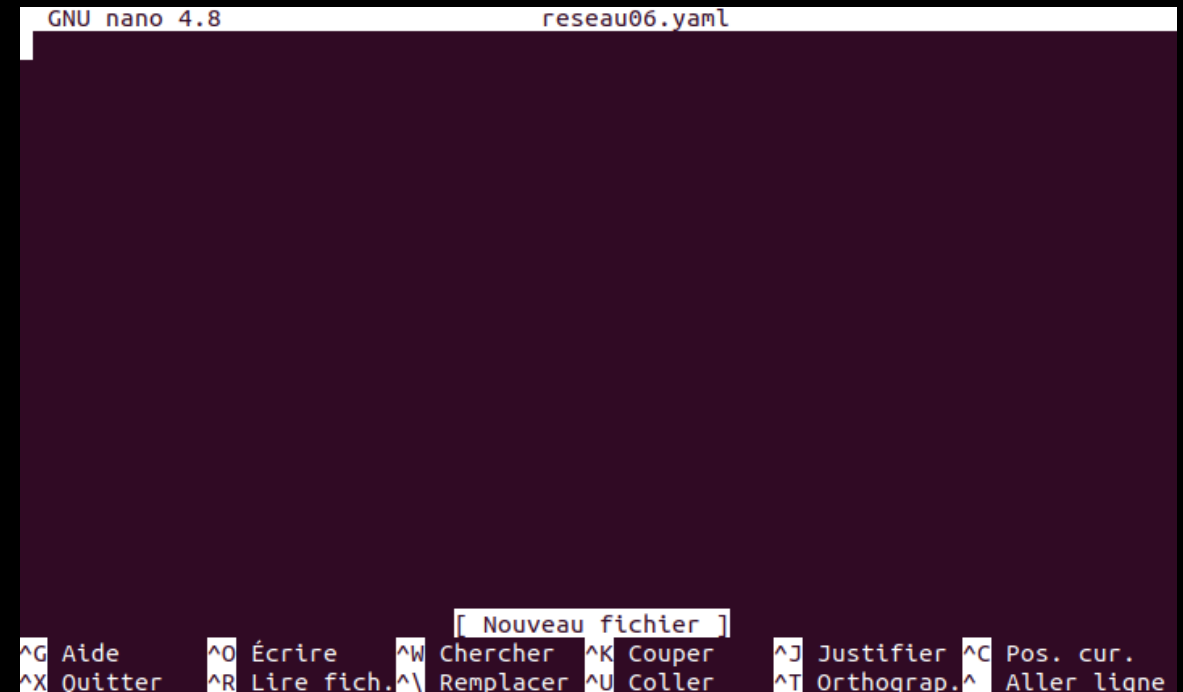
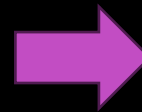
## La méthode:

Pour se faire, nous allons rejoindre le répertoire `"/etc/netplan"` depuis le terminal:





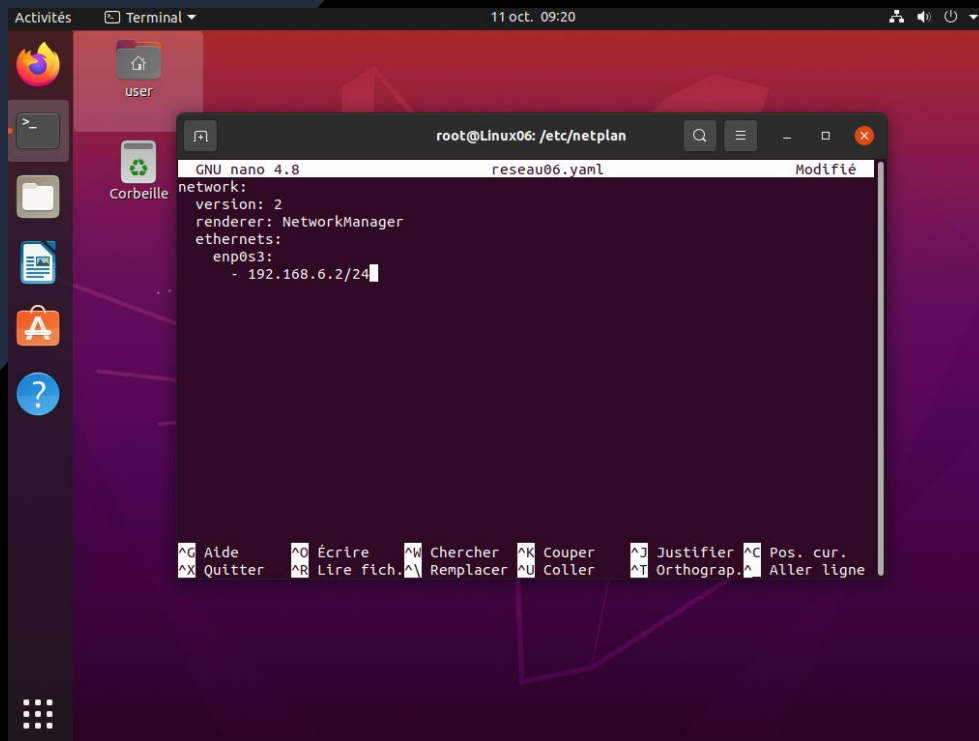
```
root@Linux06: /etc/netplan
user@Linux06:~$ sudo -s
[sudo] Mot de passe de user :
root@Linux06:/home/user# cd /etc/netplan
root@Linux06:/etc/netplan# nano reseau06.yaml
```



```
GNU nano 4.8      reseau06.yaml
[ Nouveau fichier ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper   ^J Justifier ^C Pos. cur.
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller   ^T Orthograp.^_ Aller ligne
```

Pour créer un fichier, nous allons rentrer la commande "cd /etc/netplan" pour rejoindre le répertoire en question puis indiquer le nom de notre fichier ainsi que son extension (ici **.yaml**).

Cela nous ouvre une fenêtre où nous allons rentrer des indications qui nous permettront d'adresser l'IP.



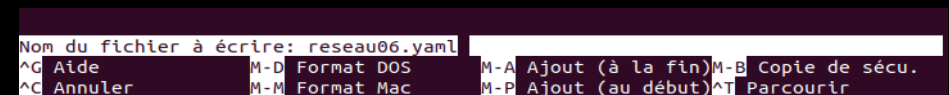
**Enp0s3:** la carte réseau

**Renderer:** Le choix du demon (gère le rendu des connexions)

Il faut également faire attention à l'indexation en respectant un espace de 2 sans tabulation sinon cela peut compromettre le fichier.

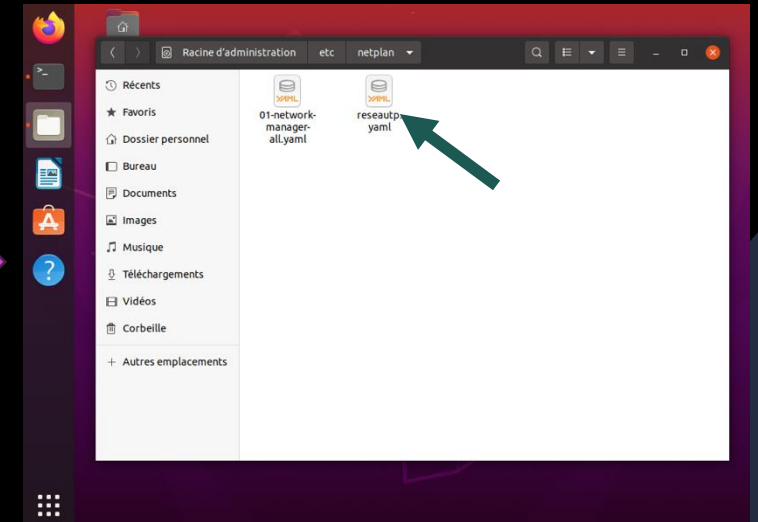
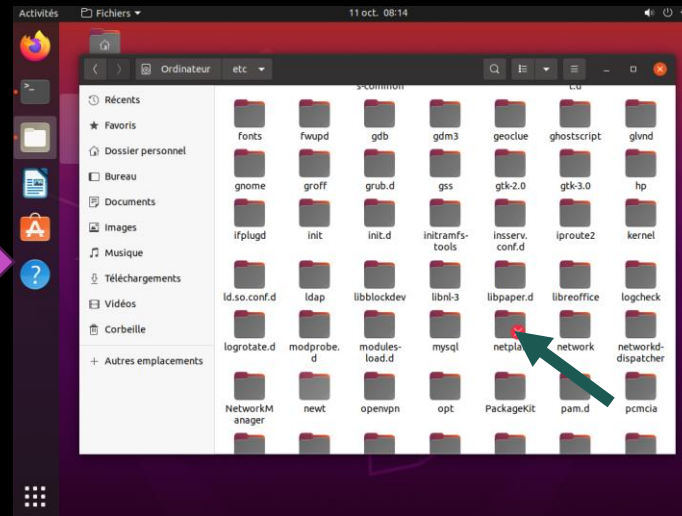
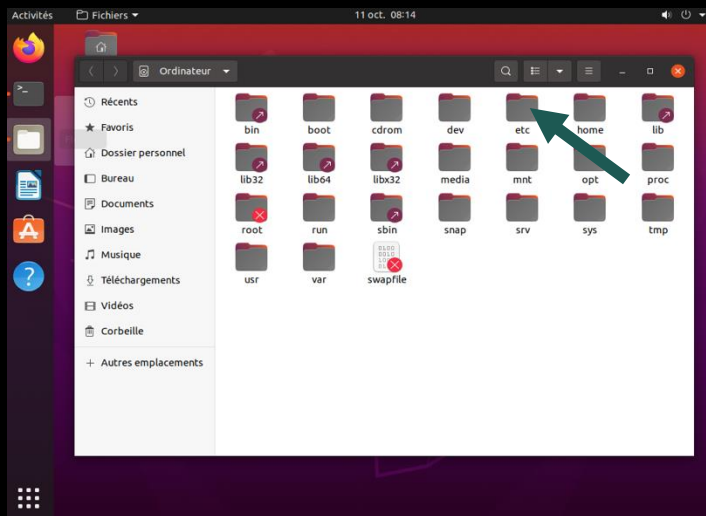
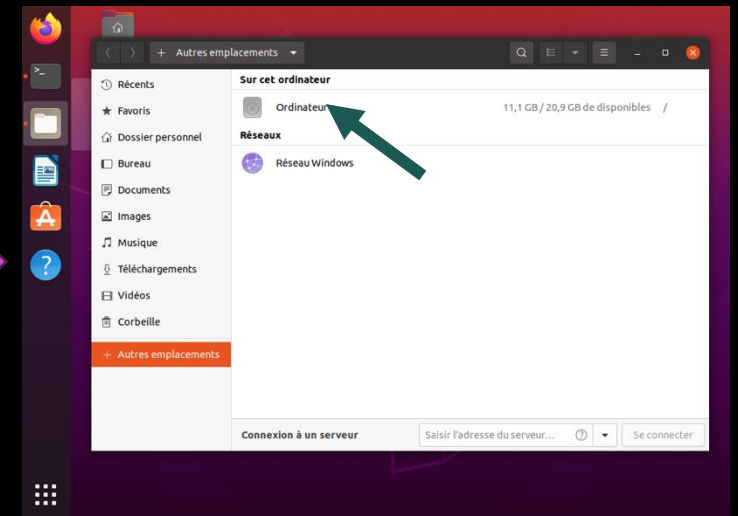
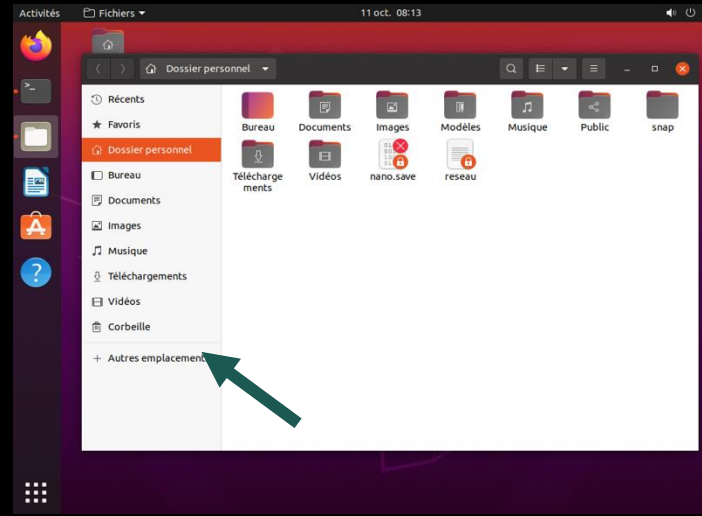
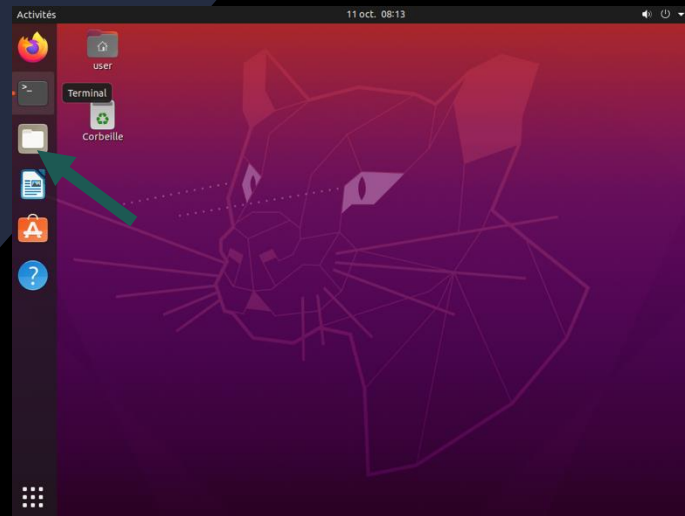


On sauvegarde avec la commande "ctrl+x" puis "O" pour confirmer son choix



Puis faites "Entrée" afin de confirmer le nom de votre fichier

# Nous pouvons voir que notre fichier "reseau06.yaml" est bien présent dans le répertoire "etc/netplan"





debian

# Configurer un adressage IP sur Debian 11 (en commande)

```
Debian GNU/Linux 11 debian06-jimmy tty1
debian06-jimmy login: root
Password:
Linux debian06-jimmy 5.10.0-33-amd64 #1 SMP Debian 5.10.226-1 (2024-10-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 11 11:42:13 CEST 2024 on tty1
root@debian06-jimmy:~# nano /etc/network/interfaces
```

Sur l'interface de commande, nous allons d'abord nous donner les droits de super-utilisateur avec le login "**root**" puis nous allons rentrer la commande "nano /etc/network/interfaces" afin d'accéder au répertoire réseau afin d'y adresser l'IP que l'on souhaite attribué.

```
GNU nano 5.4 /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
iface enp0s3 inet static
address 192.168.6.3
netmask 255.255.255.0

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^G EmplacementM-U Annuler
^X Quitter   ^R Lire fich.^N Remplacer ^U Coller    ^J Justifier ^C Aller ligneM-E Refaire
```

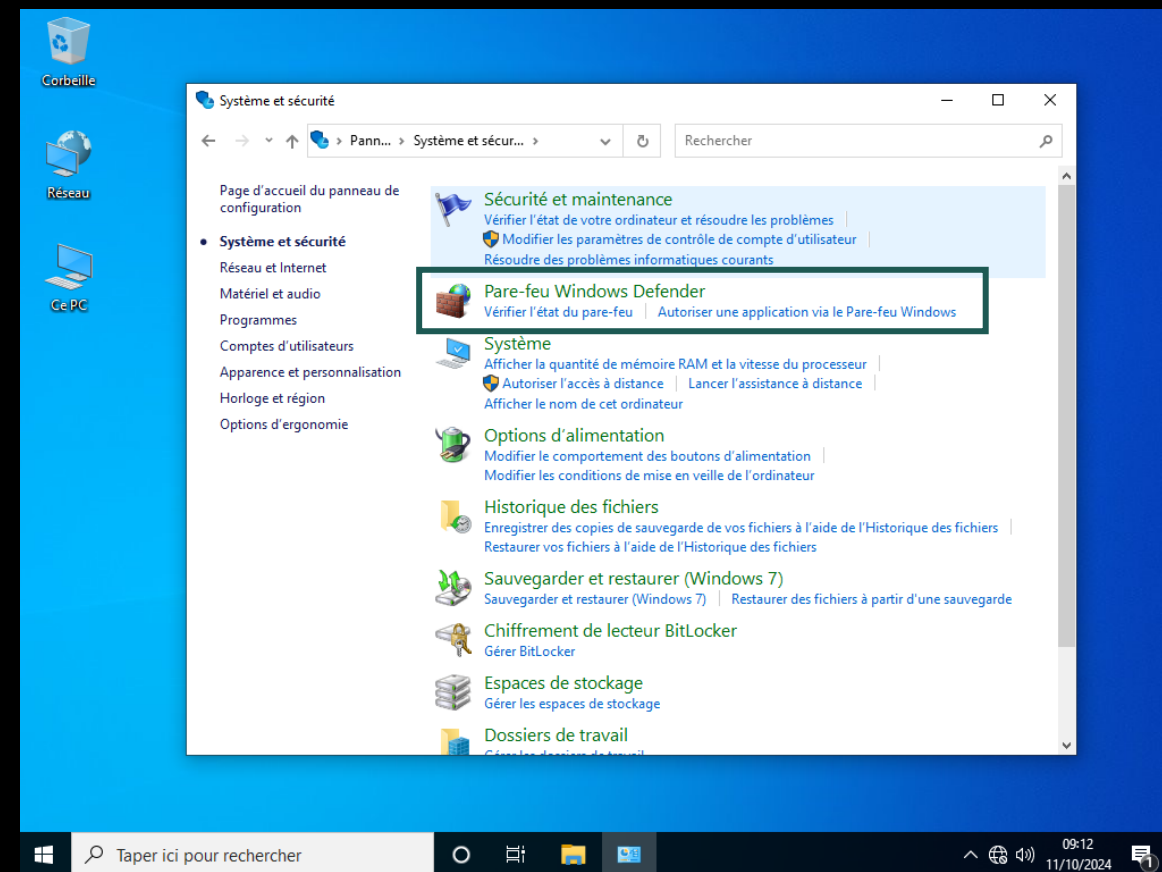
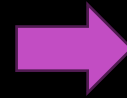
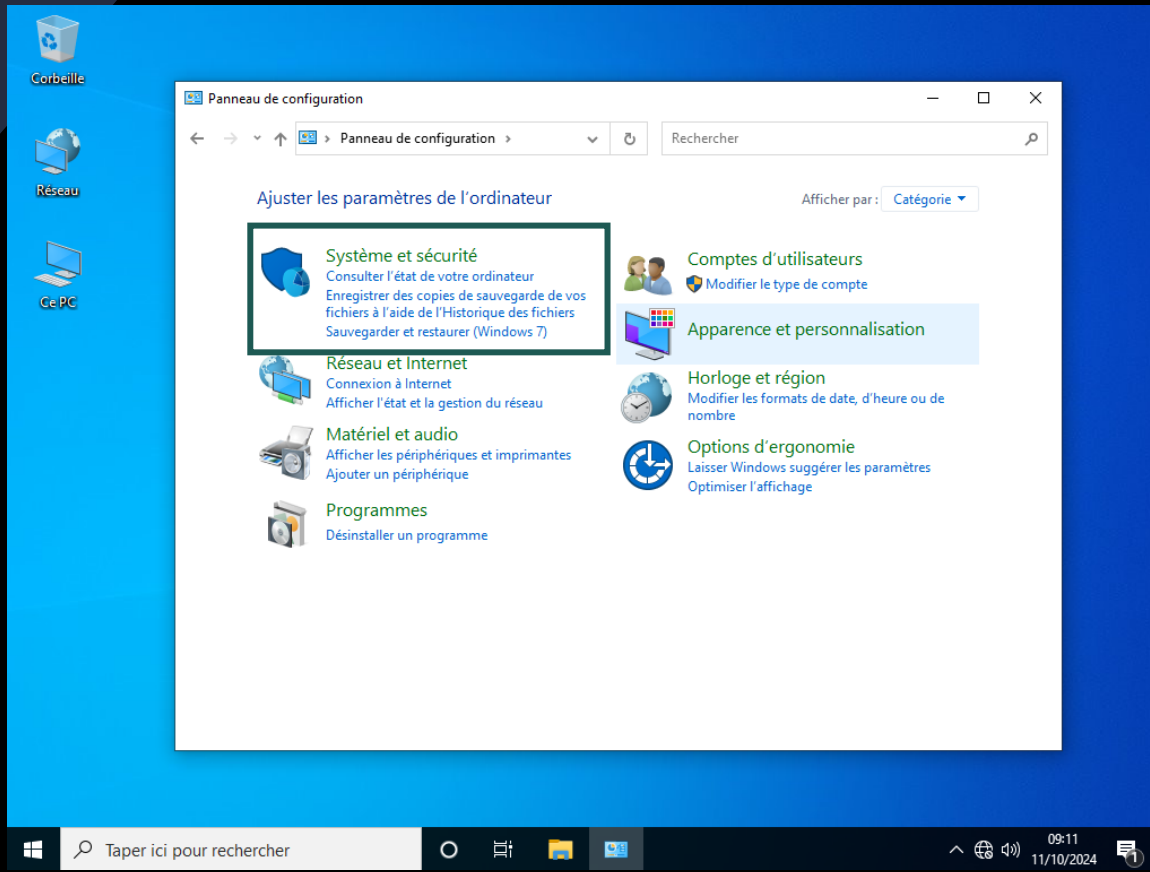
Dans le répertoire réseau, nous allons écrire l'adresse IP (ici: **192.168.6.3**) et le masque (**255.255.255.0**). Puis, faire attention à la ligne "iface enp0s3 inet static" (static peut être en DHCP par défaut, et donc pourrait ne pas fonctionner).

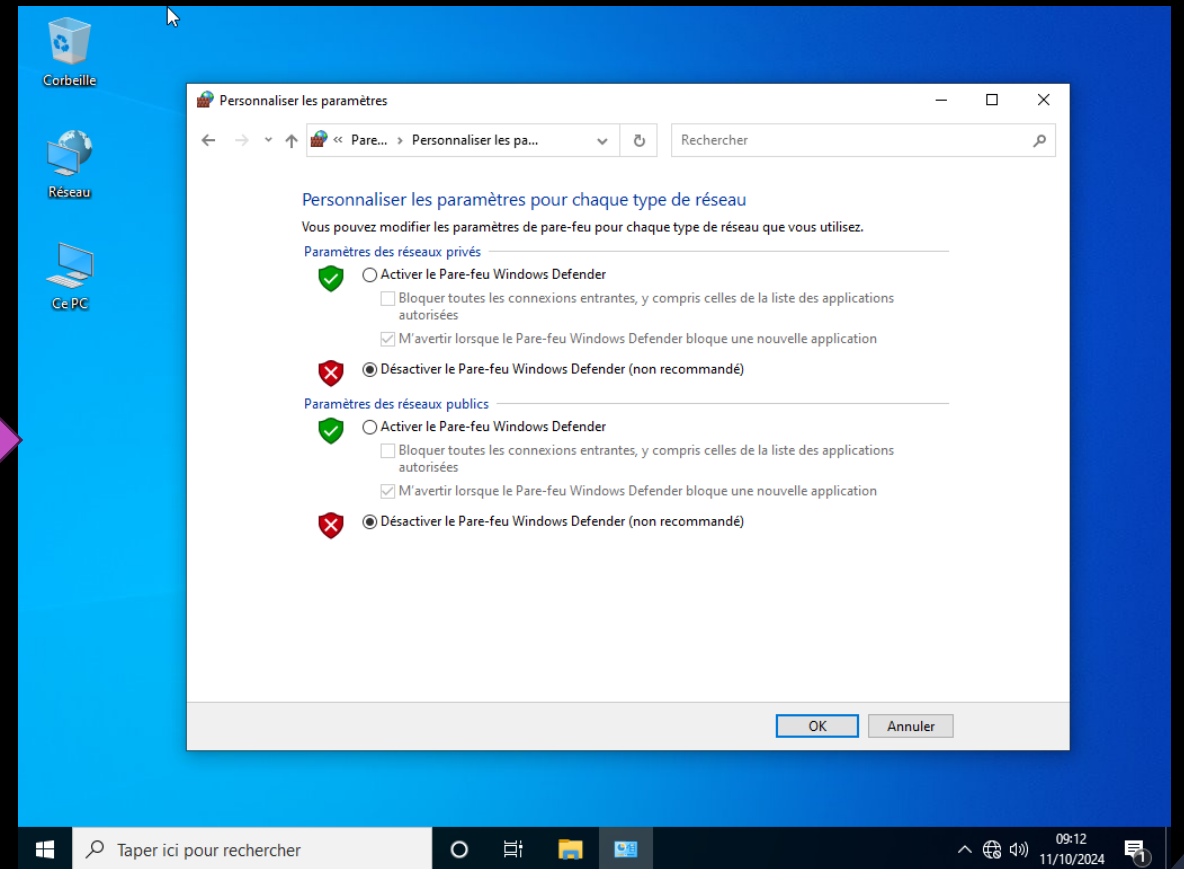
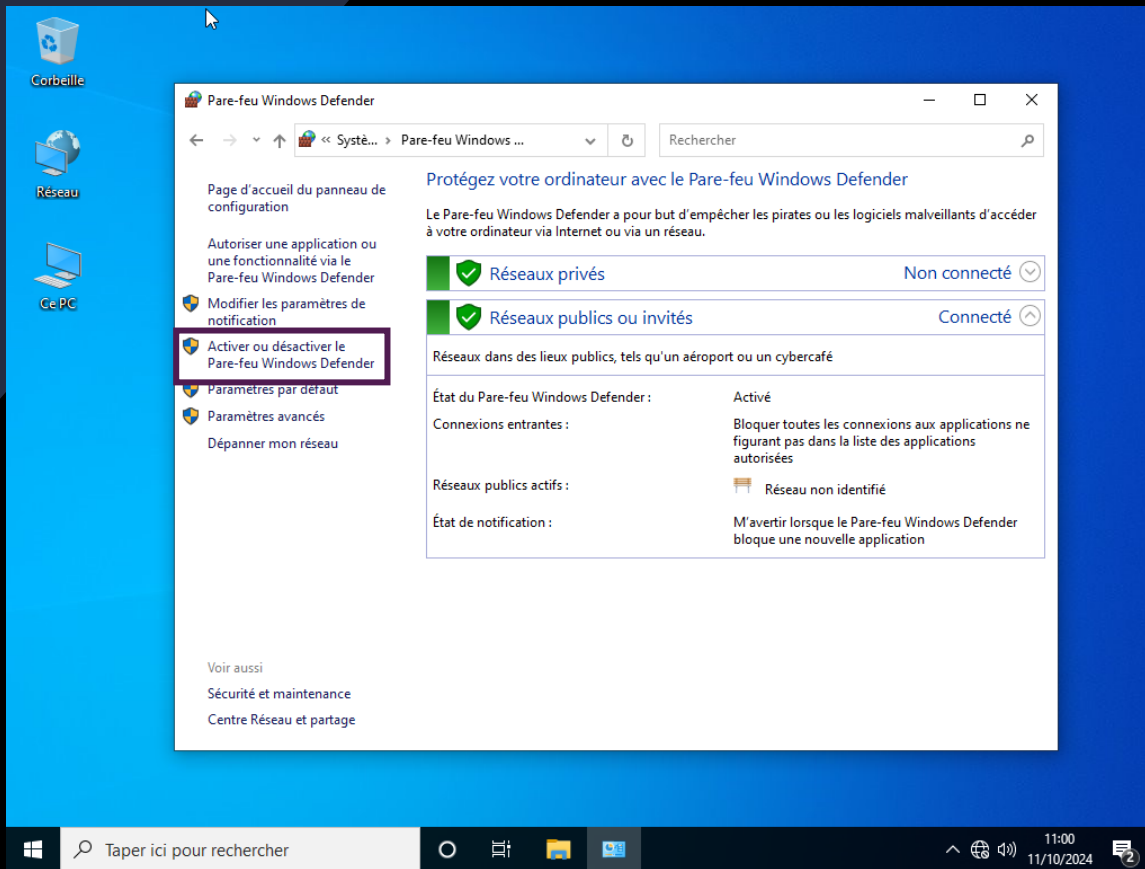
Après avoir sauvegardé votre fichier, redémarrer votre machine virtuelle afin de pouvoir prendre en compte la nouvelle IP que nous paramétrons.



# **Comment désactiver le pare-feu sur Windows 10 ?**

Pour se faire, nous retournerons dans le panneau de configuration puis dans "**Systeme et securite**" et "**Pare-feu Windows Defender**"



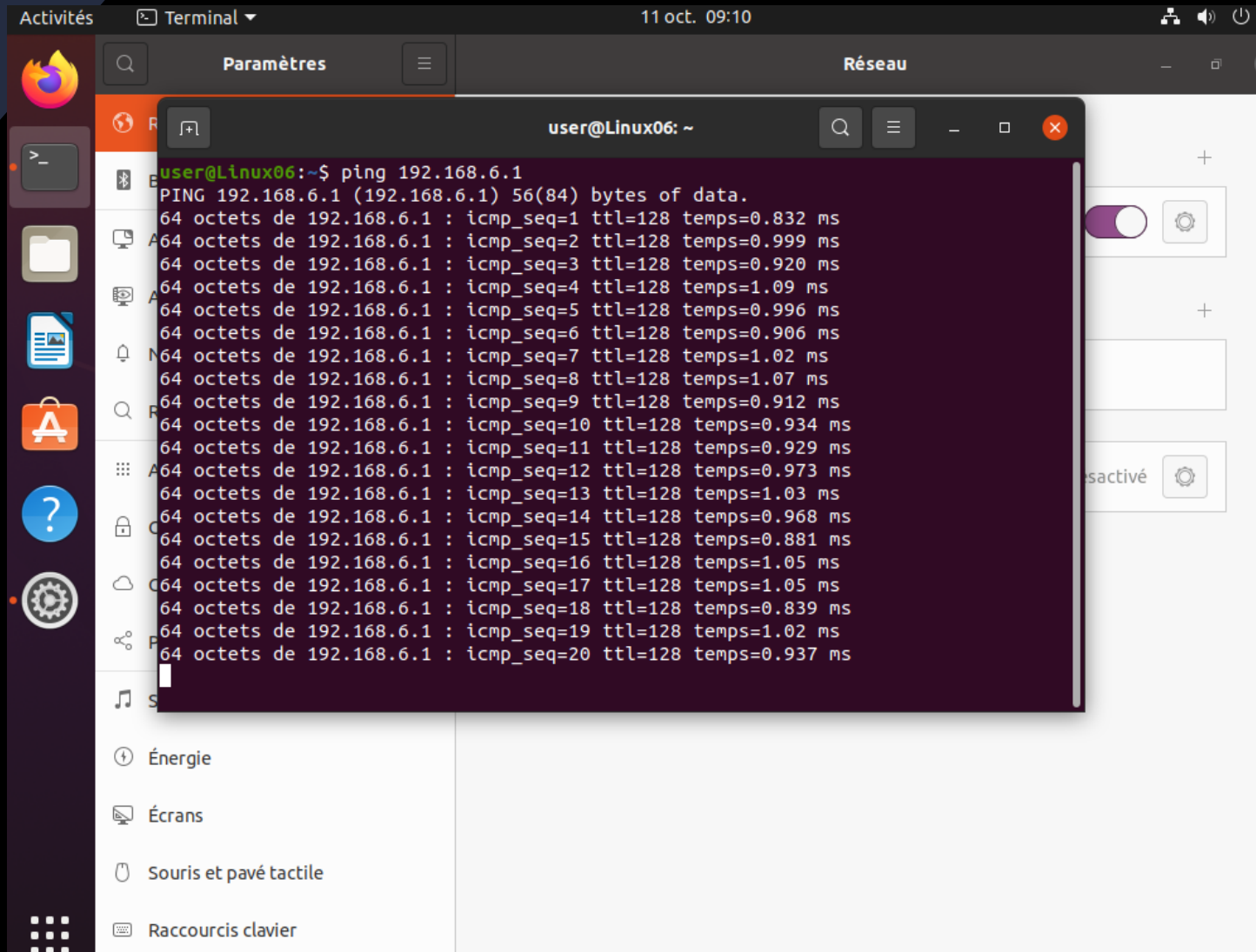


En allant dans l'onglet "**Activer ou désactiver le Pare-feu Windows Defender**", nous accédons à une page nous permettant de désactiver les options "**Windows Defender**" et le pare-feu

TESTONS MAINTENANT LA  
COMMANDE "PING" POUR  
FAIRE COMMUNIQUER NOS  
MACHINES VIRTUELLES  
ENSEMBLE.



# Et si on pingé nos 3 machines ?



```
user@Linux06: ~  
user@Linux06:~$ ping 192.168.6.1  
PING 192.168.6.1 (192.168.6.1) 56(84) bytes of data.  
64 octets de 192.168.6.1 : icmp_seq=1 ttl=128 temps=0.832 ms  
64 octets de 192.168.6.1 : icmp_seq=2 ttl=128 temps=0.999 ms  
64 octets de 192.168.6.1 : icmp_seq=3 ttl=128 temps=0.920 ms  
64 octets de 192.168.6.1 : icmp_seq=4 ttl=128 temps=1.09 ms  
64 octets de 192.168.6.1 : icmp_seq=5 ttl=128 temps=0.996 ms  
64 octets de 192.168.6.1 : icmp_seq=6 ttl=128 temps=0.906 ms  
64 octets de 192.168.6.1 : icmp_seq=7 ttl=128 temps=1.02 ms  
64 octets de 192.168.6.1 : icmp_seq=8 ttl=128 temps=1.07 ms  
64 octets de 192.168.6.1 : icmp_seq=9 ttl=128 temps=0.912 ms  
64 octets de 192.168.6.1 : icmp_seq=10 ttl=128 temps=0.934 ms  
64 octets de 192.168.6.1 : icmp_seq=11 ttl=128 temps=0.929 ms  
64 octets de 192.168.6.1 : icmp_seq=12 ttl=128 temps=0.973 ms  
64 octets de 192.168.6.1 : icmp_seq=13 ttl=128 temps=1.03 ms  
64 octets de 192.168.6.1 : icmp_seq=14 ttl=128 temps=0.968 ms  
64 octets de 192.168.6.1 : icmp_seq=15 ttl=128 temps=0.881 ms  
64 octets de 192.168.6.1 : icmp_seq=16 ttl=128 temps=1.05 ms  
64 octets de 192.168.6.1 : icmp_seq=17 ttl=128 temps=1.05 ms  
64 octets de 192.168.6.1 : icmp_seq=18 ttl=128 temps=0.839 ms  
64 octets de 192.168.6.1 : icmp_seq=19 ttl=128 temps=1.02 ms  
64 octets de 192.168.6.1 : icmp_seq=20 ttl=128 temps=0.937 ms
```

**Windows 10** : 192.168.6.1  
**Ubuntu** : 192.168.6.2  
**Debian 11** : 192.168.6.3

Le ping vers ma machine virtuelle "**Windows 10**" se fait sans soucis.

```
Debian GNU/Linux 11 debian06-jimmy tty1

debian06-jimmy login: root
Password:
Linux debian06-jimmy 5.10.0-33-amd64 #1 SMP Debian 5.10.226-1 (2024-10-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 11 11:38:24 CEST 2024 on tty1
root@debian06-jimmy:~# ping 192.168.6.1
PING 192.168.6.1 (192.168.6.1) 56(84) bytes of data.
64 bytes from 192.168.6.1: icmp_seq=1 ttl=128 time=1.50 ms
64 bytes from 192.168.6.1: icmp_seq=2 ttl=128 time=0.837 ms
64 bytes from 192.168.6.1: icmp_seq=3 ttl=128 time=0.858 ms
64 bytes from 192.168.6.1: icmp_seq=4 ttl=128 time=0.914 ms
64 bytes from 192.168.6.1: icmp_seq=5 ttl=128 time=0.957 ms
^C
--- 192.168.6.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4029ms
rtt min/avg/max/mdev = 0.837/1.014/1.504/0.248 ms
root@debian06-jimmy:~#
```

Le ping de ma machine  
**"Debian 11"** en commande  
vers ma machine **"Windows  
10"** se fait sans soucis.

```
Microsoft Windows [version 10.0.19043.928]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Test>ping 192.168.6.2

Re Envoi d'une requête 'Ping' 192.168.6.2 avec 32 octets de données :
Réponse de 192.168.6.2 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.6.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.6.2 : octets=32 temps<1ms TTL=64
Réponse de 192.168.6.2 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.6.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

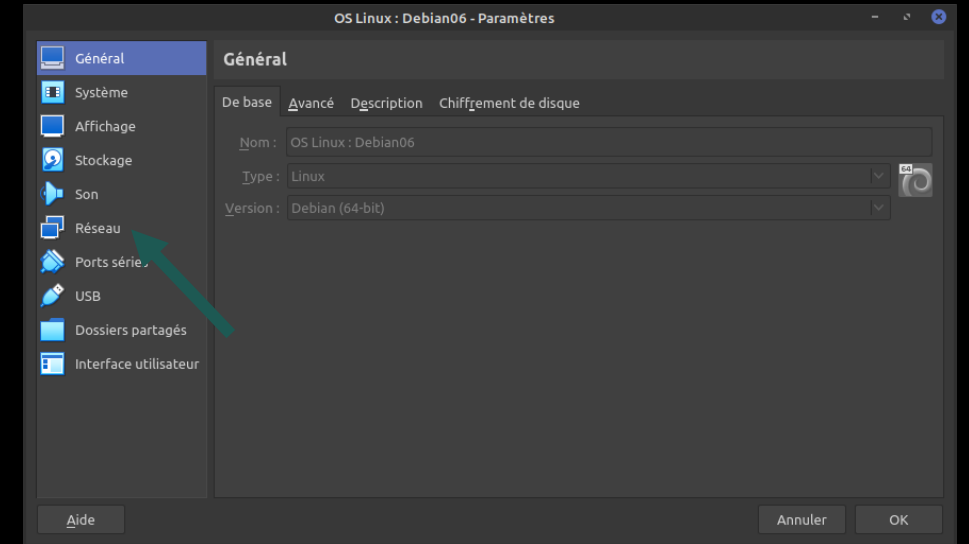
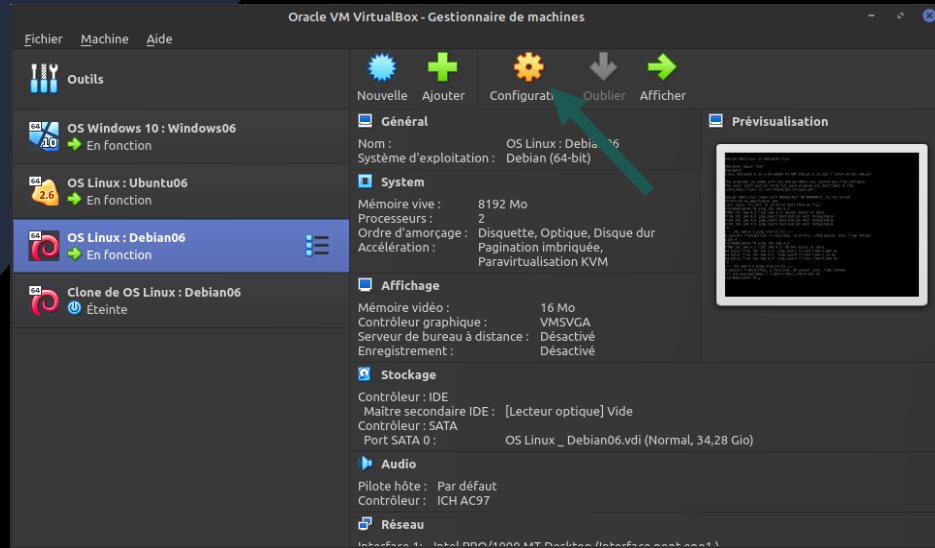
C:\Users\Test>
```

Le ping de ma machine "**Windows 10**" vers ma machine "**Ubuntu**" est opérationnel.

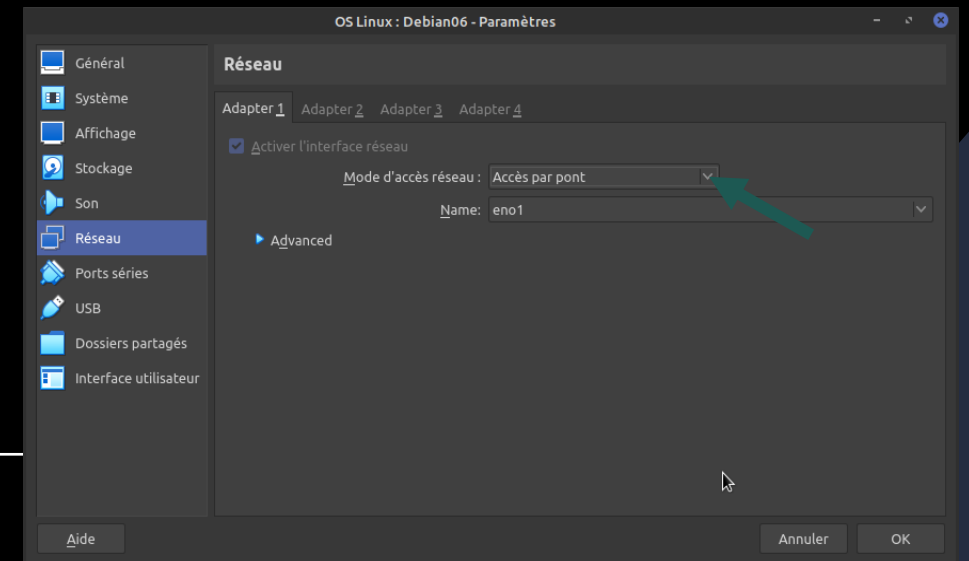
# Configuration de 6 machines virtuels pour 2 postes de pc différent



# Configurer ses postes de travail en accès par pont sur VirtualBox



Pour mettre en accès par pont, il faut **sélectionner la machine virtuelle correspondante**, cliquer sur "**Configuration**", accéder à l'onglet "**Réseau**" puis dans "**Mode d'accès réseau**" mettez "**Accès par pont**"



# Après avoir configuré les deux postes de pc différents, essayons de communiquer avec lui

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

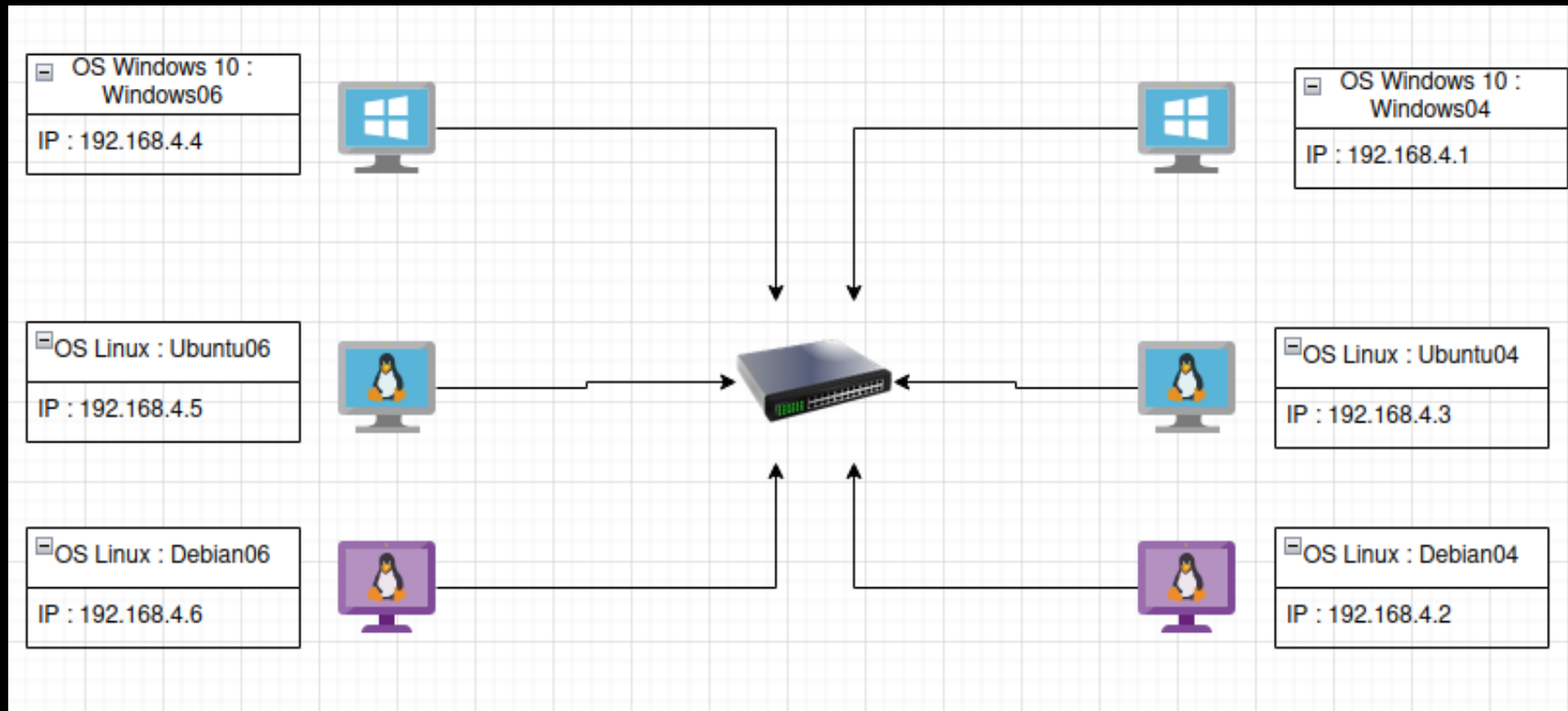
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 11 11:37:23 CEST 2024 on tty1
root@debian06-jimmy:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOW state DOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:01:81:06 brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.6/24 brd 192.168.4.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe01:8106/64 scope link
        valid_lft forever preferred_lft forever
root@debian06-jimmy:~# ping 192.168.4.2
PING 192.168.4.2 (192.168.4.2) 56(84) bytes of data:
54 bytes from 192.168.4.2: icmp_seq=1 ttl=64 time=0.953 ms
54 bytes from 192.168.4.2: icmp_seq=2 ttl=64 time=0.688 ms
54 bytes from 192.168.4.2: icmp_seq=3 ttl=64 time=0.823 ms
54 bytes from 192.168.4.2: icmp_seq=4 ttl=64 time=0.777 ms
54 bytes from 192.168.4.2: icmp_seq=5 ttl=64 time=0.575 ms
54 bytes from 192.168.4.2: icmp_seq=6 ttl=64 time=0.990 ms
54 bytes from 192.168.4.2: icmp_seq=7 ttl=64 time=0.935 ms
54 bytes from 192.168.4.2: icmp_seq=8 ttl=64 time=0.819 ms
54 bytes from 192.168.4.2: icmp_seq=9 ttl=64 time=0.824 ms
54 bytes from 192.168.4.2: icmp_seq=10 ttl=64 time=0.818 ms
54 bytes from 192.168.4.2: icmp_seq=11 ttl=64 time=0.684 ms
54 bytes from 192.168.4.2: icmp_seq=12 ttl=64 time=0.638 ms
54 bytes from 192.168.4.2: icmp_seq=13 ttl=64 time=1.01 ms
```

J'ai envoyé un ping sur la station de travail "SIO4" et celui-ci se fait sans soucis après que j'ai redéfinis mes adresses IP (192.168.6.X -> 192.168.4.X).

Dans ce cas précis, je fais un ping depuis l'adresse 192.168.4.6 (mon poste/SIO6) vers l'adresse 192.168.4.2 (autre poste)

J'ai tenté de faire le ping depuis mon Debian II en interface de commande.

# Le diagramme du réseau en accès par pont



Fait sur draw.io

FIN