

# Compte Rendu - SAE 103

## Installation et configuration de systèmes d'exploitation.

# Sommaire

Glossaire.....	3
Introduction.....	4
Installer deux machines virtuelles.....	5
VirtualBox.....	5
Installation de Debian.....	6
Installation de Windows.....	8
Faire “ discuter ” deux machines entre-elles.....	10
Utilisation d’Apache pour transférer des fichiers.....	12
Partage de fichier en dehors d’Apache.....	14
Linux vers Windows.....	14
Windows vers Linux.....	16
Controlling Linux from a Windows VM using SSH.....	18
Problème rencontré.....	21
Conclusion.....	22

## Glossaire

Ce glossaire présente les termes techniques, logiciel utilisé à travers notre compte-rendu.

Mot-Clé	Définition - Utilité
Oracle Apache	Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde.
Nautilus-Admin	Nautilus Admin est une extensions de l'explorateur de fichier par défaut de Debian, permettant en outre l'accès au dossier en tant qu'admin.
VM ou MV	Machine Virtuelle: Simulation la présence de ressources matérielles et logicielles telles que la mémoire, le processeur, le disque dur, voire le système d'exploitation et les pilotes, permettant d'exécuter des programmes dans les mêmes conditions que celles de la machine simulée
Proxy	Un proxy est un composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs échanges.
ISO, Image ISO	Il s'agit d'un fichier contenant une copie "conforme", à l'identique d'un disque. Ce fichier "clone", contient l'intégralité du support d'origine, et en particulier tout ce qui est nécessaire à l'installation et à l'utilisation d'un logiciel, d'un jeu, d'une vidéo, ou, le plus souvent, d'un système d'exploitation.

Mode réseau	Exemple d'utilisation
Accès par pont	Vous souhaitez que votre VM devienne un serveur web et donc qu'elle soit vue sur le réseau comme une machine physique avec sa propre adresse IP, accessible depuis l'extérieur.
Réseau interne	Vous utilisez plusieurs VM pour simuler des réseaux privés qui n'ont pas d'accès vers l'extérieur ni vers la machine hôte.
Réseau NAT	Même type d'utilisation que pour le NAT, avec plusieurs VM qui souhaitent pouvoir communiquer entre elles (impossible avec le mode NAT classique)
Réseau privé d'hôte	Similaire au réseau interne, mais permettant aussi une communication avec l'hôte.

## Introduction

Nous sommes étudiants en 1ère année de BUT Informatique à l'IUT de Maubeuge.  
Dans le cadre de la ressource SAE 103, nous avons mis en place deux machines virtuelles à travers l'utilisation de VirtualBox.

Nous avons pu ainsi créer une machine virtuelle contenant Windows 10 et une deuxième sous Debian 12.

À travers, cette SAE nous avons fait communiquer nos deux machines entre-elle à travers un réseau NAT.

Nous avons aussi réalisé le partage de fichier à travers la mise en place d'un serveur Apache, mais aussi en utilisant Samba.

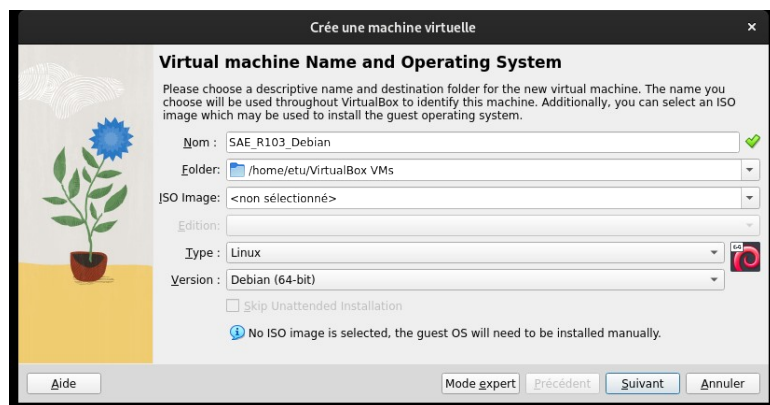
Enfin, nous sommes arrivés à utiliser notre machine virtuelle sous Debian à travers des commandes envoyé depuis la machine sous Windows, le tout étant sécurisé avec le SSH.

Nous vous remercions d'avance, pour le temps que vous nous accordez en lisant ce compte-rendu.

## Installer deux machines virtuelles

### VirtualBox

Afin de pouvoir travailler et effectuer les différentes tâches que nous avons à faire, nous avons besoin d'utiliser des Machines Virtuelle. Pour cela, nous allons utiliser le logiciel Oracle VirtualBox qui permet de gérer des Machines Virtuelle.



Pour créer une nouvelle machine, nous devons lui attribuer un nom, son emplacement et le type d'OS que vous souhaitons installer.

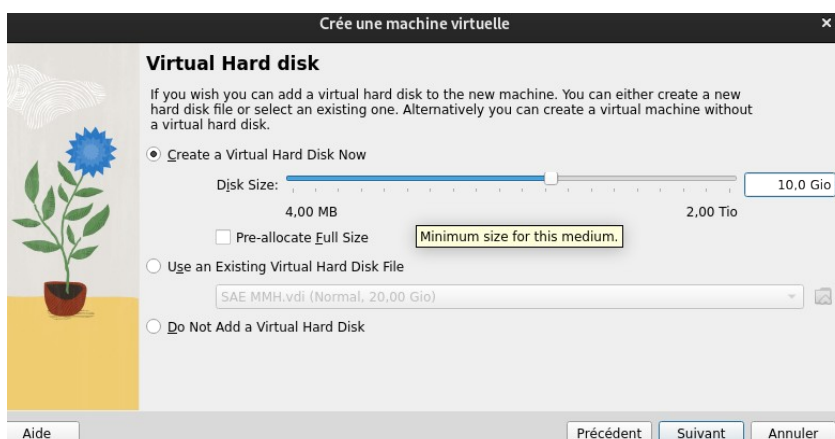
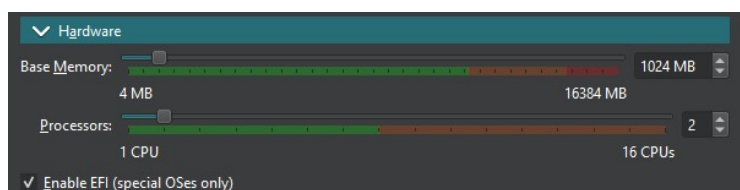
**Table 3.2. Recommended Minimum System Requirements**

Install Type	RAM (minimum)	RAM (recommended)	Hard Drive
No desktop	256 megabytes	512 megabytes	4 gigabytes
With Desktop	1 gigabytes	2 gigabytes	10 gigabytes

Pour Debian, le minimum requis pour la mémoire vive est de 1 Gigabits et pour l'espace de stockage de 10 Gigabits.

Pour Windows, le minimum requis pour la mémoire vive est de 2 Gigabits et pour l'espace de stockage de 32 Gigabits.

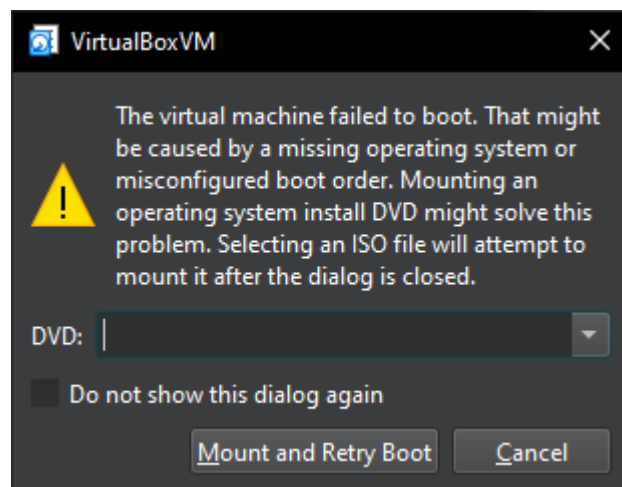
RAM :	1 gigaoctet (Go) pour système 32 bits ou 2 Go pour système 64 bits
Taille du disque dur :	16 Go pour système 32 bits ou 32 Go pour système 64 bits



Après avoir créé la machine virtuelle, nous pouvons la lancer. Lors de son premier lancement celle-ci va retourner un échec. Ceci est normal, car nous n'avons pas précisé leurs de la mise en place le chemin vers l'ISO.

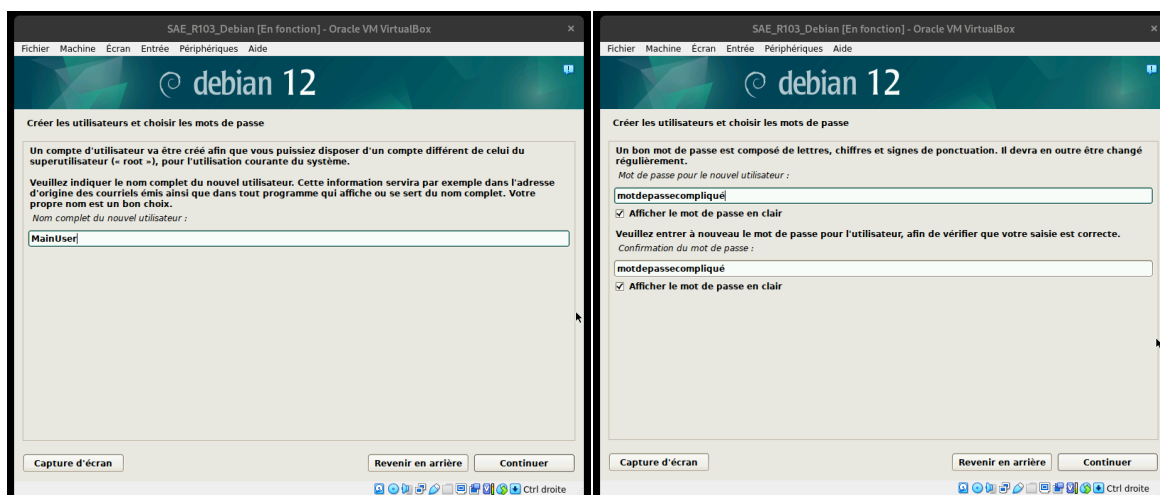
Nous avons fait cela, car lors de nos essais précédents des erreurs d'installation se sont produites, cette façon de faire à permis de les éviter. Il nous reste donc à mettre tout simplement le chemin de l'ISO et de refaire le lancement.

De plus, nous donc fait cette manipulation deux fois, une fois pour Debian et une pour Windows avec leurs spécificités.

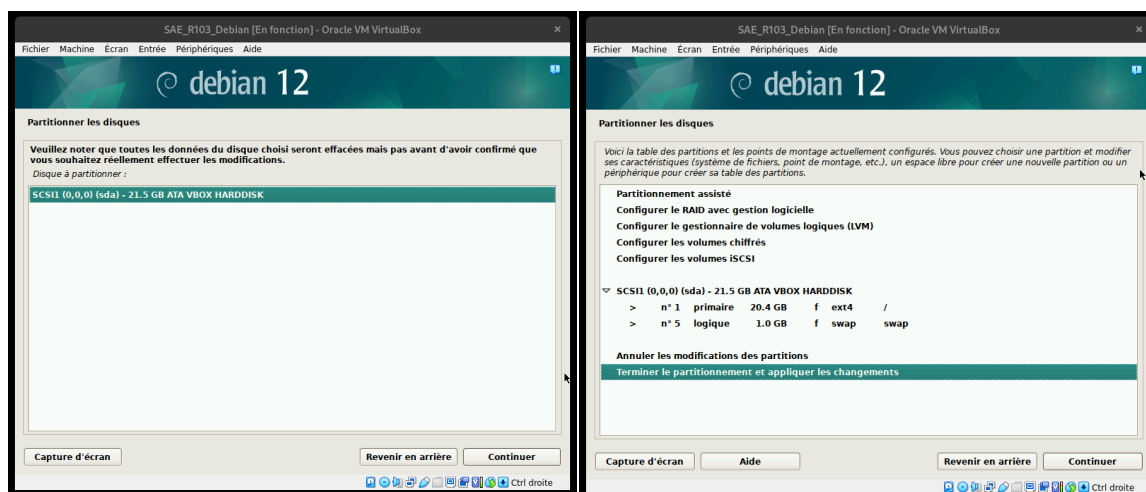


## Installation de Debian

Lors de l'installation de Debian, nous devons mettre en place un utilisateur et son mot de passe.



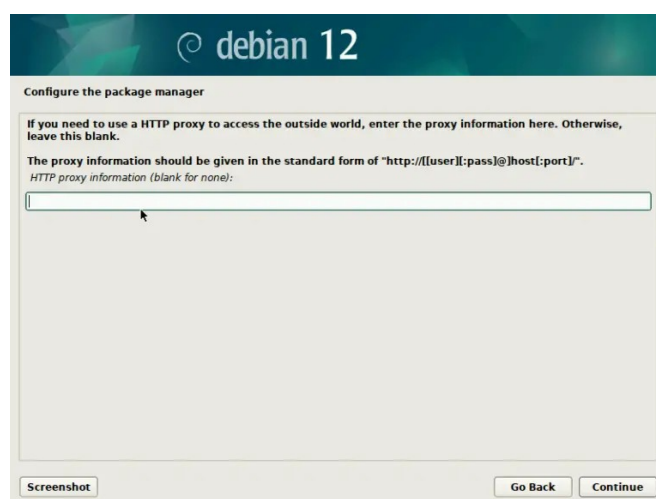
Pour la partition des disques, nous pouvons laisser l'installateur le faire automatiquement pour nous, étant donné que nous n'avons pas besoin de séparer notre disque dur. Vu que dans la machine virtuelle celui-ci est déjà compartimenté dans notre disque dur physique par Virtual Box.



Dernièrement, nous devons mettre en place le proxy, pour que notre installateur Debian est accès à Internet et puisse télécharger et installer les packages manquant, dont l'interface Graphique.

Le proxy est :

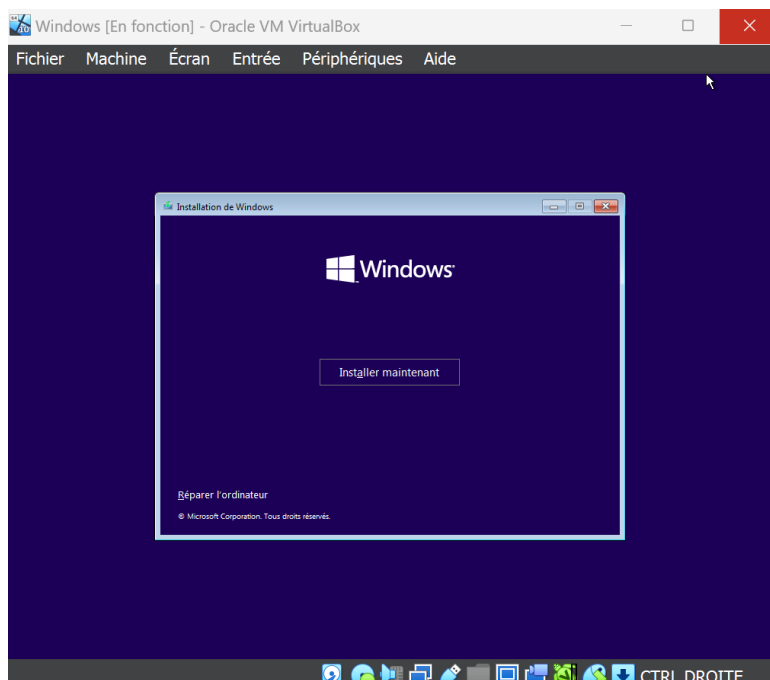
["http://proxy1-rech.uphf.fr:3128"](http://proxy1-rech.uphf.fr:3128).



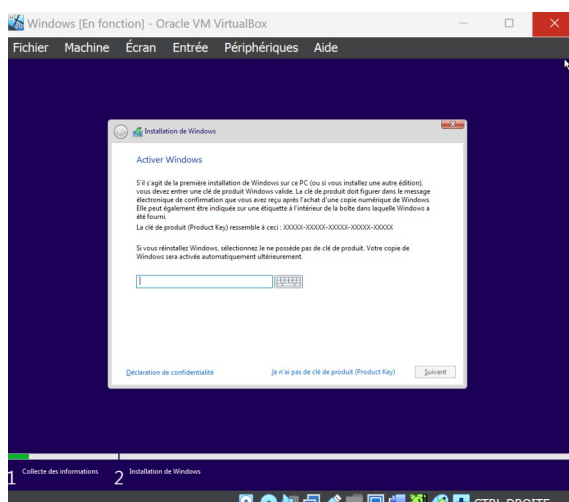
Il ne nous reste plus qu'à patienter le temps de son installation. Nous venons de terminer l'installation de Debian sur notre machine virtuelle.

## Installation de Windows

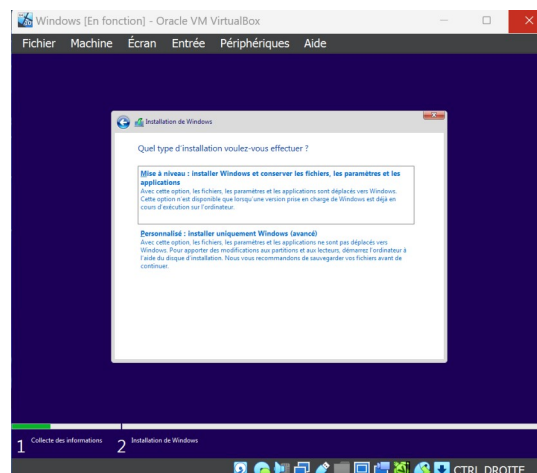
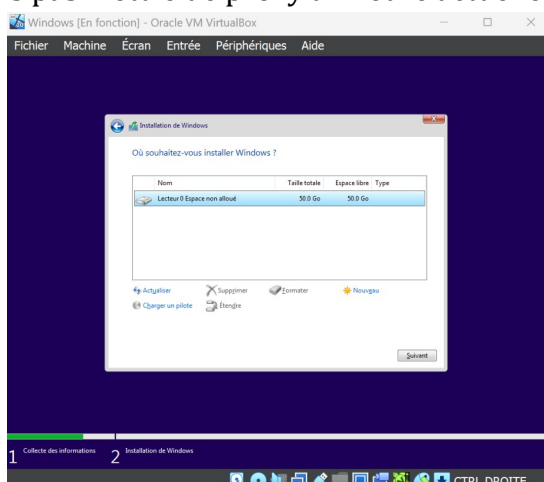
Après avoir créé une machine virtuelle sur VirtualBox et avec l'ISO monté, nous pouvons procéder à l'installation de Windows.



Nous pouvons passer l'activation de notre Windows, les fonctions basiques de Windows vont nous permettre de réaliser les tâches de notre projet.

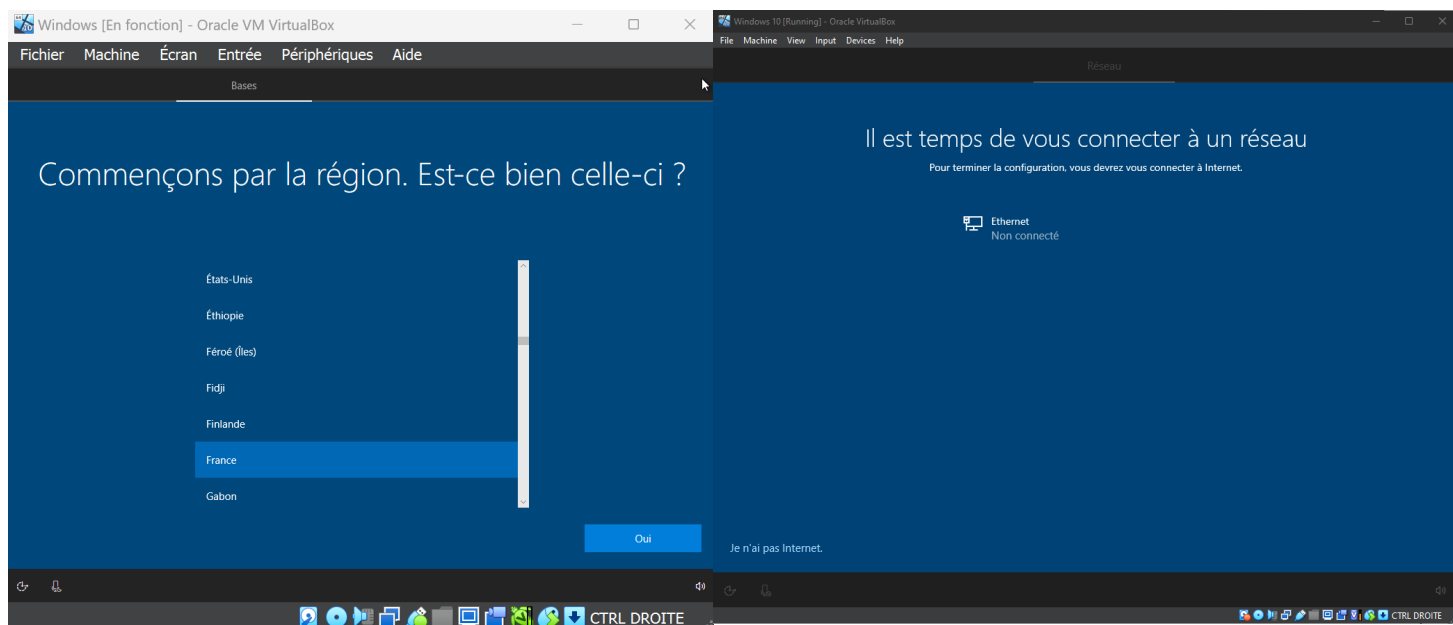


Enfin, nous pouvons laisser les espaces du disque par défaut, même raison que pour Debian. Par contre nous devons bien choisir de faire une installation personnalisée, car nous ne pouvons pas mettre de proxy à l'heure actuelle.



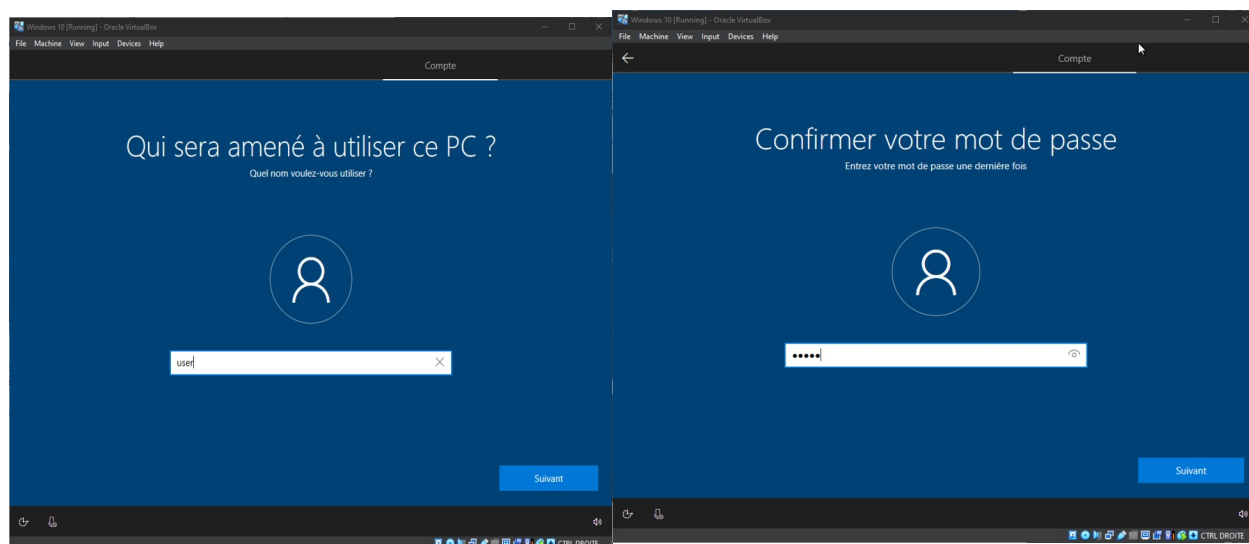


Une fois cette partie finie, nous pouvons continuer sur la partie générale de l'installation de Windows, celle-ci permet la personnalisation de Windows par rapport aux envies de l'utilisateur. Dans notre cas, nous allons sélectionner le minimum possible, n'ayant pas besoin de plus.



N'ayant pas la possibilité de mettre de proxy, nous continuons en cliquant sur "Je n'ai pas Internet" et nous continuons avec la version de Windows limité.

Vient ensuite la création de notre utilisateur, dans notre cas, nous allons mettre User en tant que nom et userMDP en tant que mot de passe.

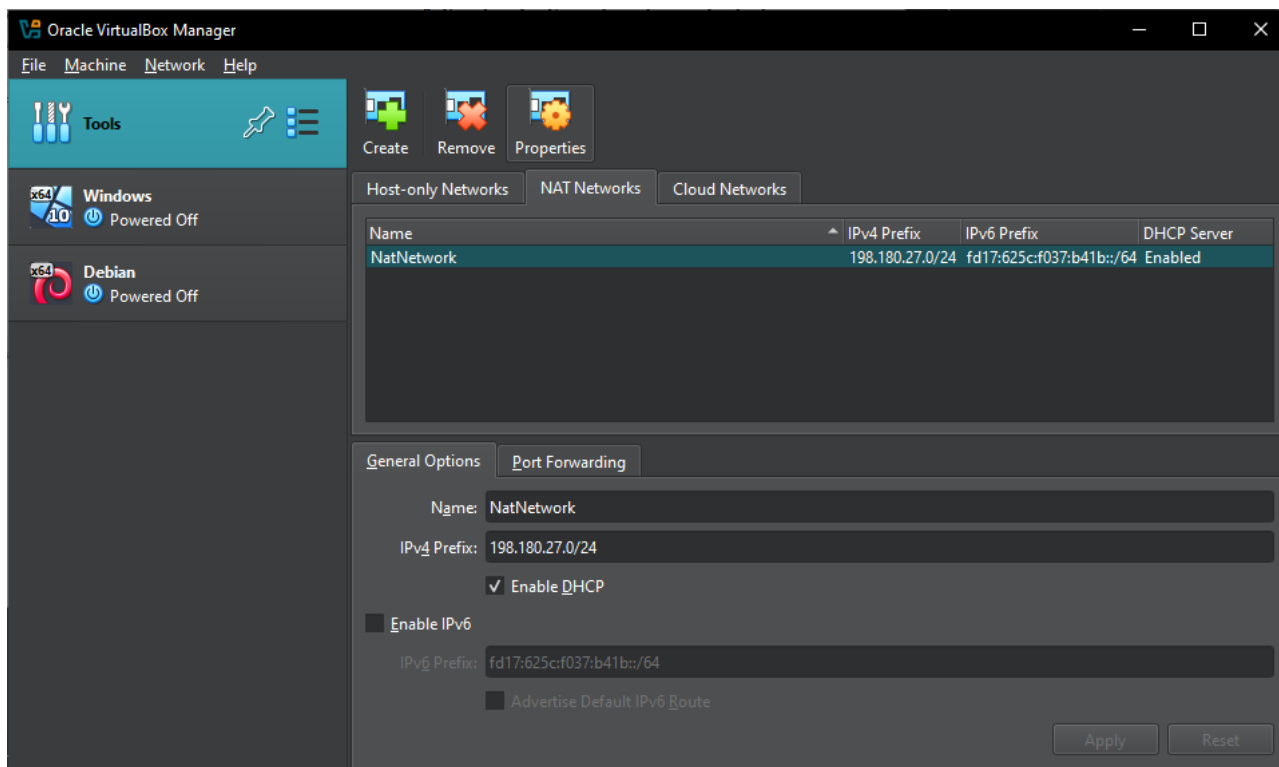


Enfin, nous pouvons refuser tous les services supplémentaires. Nous venons de finir l'installation de Windows sur notre machine virtuelle.

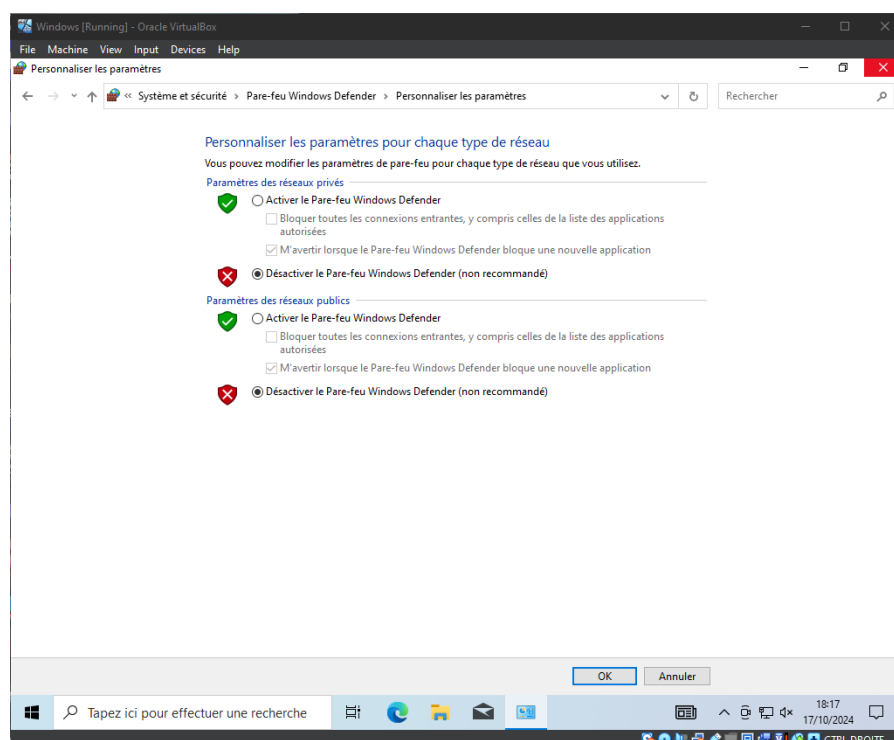
## Faire “ discuter ” deux machines entre-elles

Pour faire discuter deux machines entre-elles, nous allons devoir configurer un réseau local ainsi que leurs propres IP pour que les deux machines puissent se voir et s’entendre.

Nous avons donc mis en place à travers VirtualBox un réseau NAT, celui-ci permet de connecter plusieurs VM à Internet ainsi que la possibilité de les faire communiquer ensemble. Nous leur avons attribué l’adresse IPv4 suivante : 198.180.27.x avec un id propre allant de 0 à 24.



Une fois fais, nous devons désactiver les pare-feu de Windows afin que celui-ci ne nous refuse pas la tentative de discussion venant de notre machine Debian.



Pour récupérer l'IPv4 attribué à nos VM nous devons rentrer les commandes suivantes :  
"ipconfig" pour Windows et "ip a" pour Debian.

Enfin, pour vérifier que la communication est bien possible entre nos deux machines, nous allons utiliser la commande ping dans le terminal de chaque machine suivis de l'adresse IPv4 de la seconde machine pour avoir un retour de connexion établie.

Elles ont ainsi discuté entre-elles.

```

Debian [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
18 oct. 18:48
user@user: ~
user@user:~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:45:21:b9 brd ff:ff:ff:ff:ff:ff
    inet 198.180.27.5/24 brd 198.180.27.255 scope global dynamic noprefixroute enp0s3
        valid_lft 578sec preferred_lft 578sec
    inet6 fe80::a00:27ff:fe45:21b9/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
user@user:~$ ping 198.180.27.4
PING 198.180.27.4 (198.180.27.4) 56(84) bytes of data.
64 bytes from 198.180.27.4: icmp_seq=1 ttl=128 time=0.244 ms
64 bytes from 198.180.27.4: icmp_seq=2 ttl=128 time=0.216 ms
64 bytes from 198.180.27.4: icmp_seq=3 ttl=128 time=0.267 ms
^C
--- 198.180.27.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.216/0.242/0.267/0.020 ms
user@user:~$

Windows [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
C:\Windows\system32\cmd.exe
Microsoft Windows [version 10.0.19045.2965]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\user>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . . : lan
    Adresse IPv6 de liaison locale. . . . . : fe80::34e9:2021:63a7:28cd%9
    Adresse IPv4. . . . . : 198.180.27.4
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 198.180.27.1

C:\Users\user>ping 198.180.27.5

Envoi d'une requête 'Ping' 198.180.27.5 avec 32 octets de données :
Réponse de 198.180.27.5 : octets=32 temps<1ms TTL=64
Réponse de 198.180.27.5 : octets=32 temps<1ms TTL=64
Réponse de 198.180.27.5 : octets=32 temps<1ms TTL=64
Réponse de 198.180.27.5 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 198.180.27.5:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms

C:\Users\user>
  
```

## Utilisation d'Apache pour transférer des fichiers

Nous avons décidé d'installer Apache sous Debian afin de mettre en place le partage de fichier à travers celui-ci.

Nous devons ainsi vérifier si Apache est déjà présent sous Debian et est en fonctionnement.

Pour cela, nous pouvons faire la commande suivante :

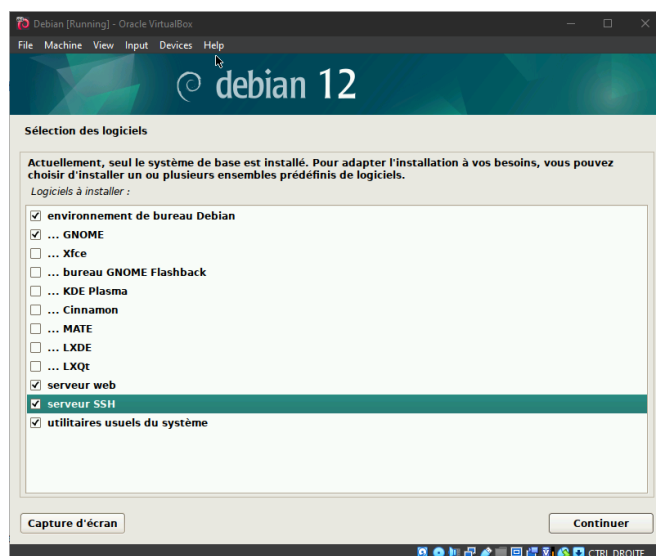
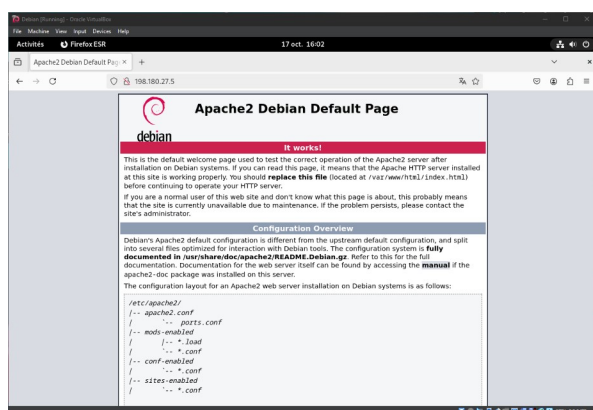
“sudo systemctl status apache2 ”

Si ce n'est pas le cas, nous pouvons l'installer avec la commande suivante :

“sudo apt install apache2 ”

Pour notre cas, lors de l'installation de Debian sur la VM, nous avons coché l'option “serveur web” afin de l'installer directement.

Ainsi avec Apache installé et en fonctionnement, nous pouvons tester en lançant notre navigateur et en tapant “localhost” ou alors directement notre adresse IPv4.



Ensuite, pour mettre en place notre transfert de fichier, il suffit de se rendre au niveau du répertoire principal de notre serveur apache, qui est dans notre cas situé :

“var\www\html”

Par ailleurs, nous pouvons retrouver le fichier “index.html” qui est la page d'accueil par défaut d'Apache.

Nous allons donc créer deux dossiers dans ce répertoire et y déposer deux fichiers.

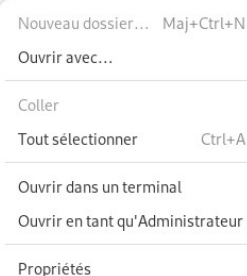
Nous avons installé Nautilus-Admin qui permet directement avec l'explorateur de fichier d'ouvrir le dossier en tant qu'Admin.



Dossier01



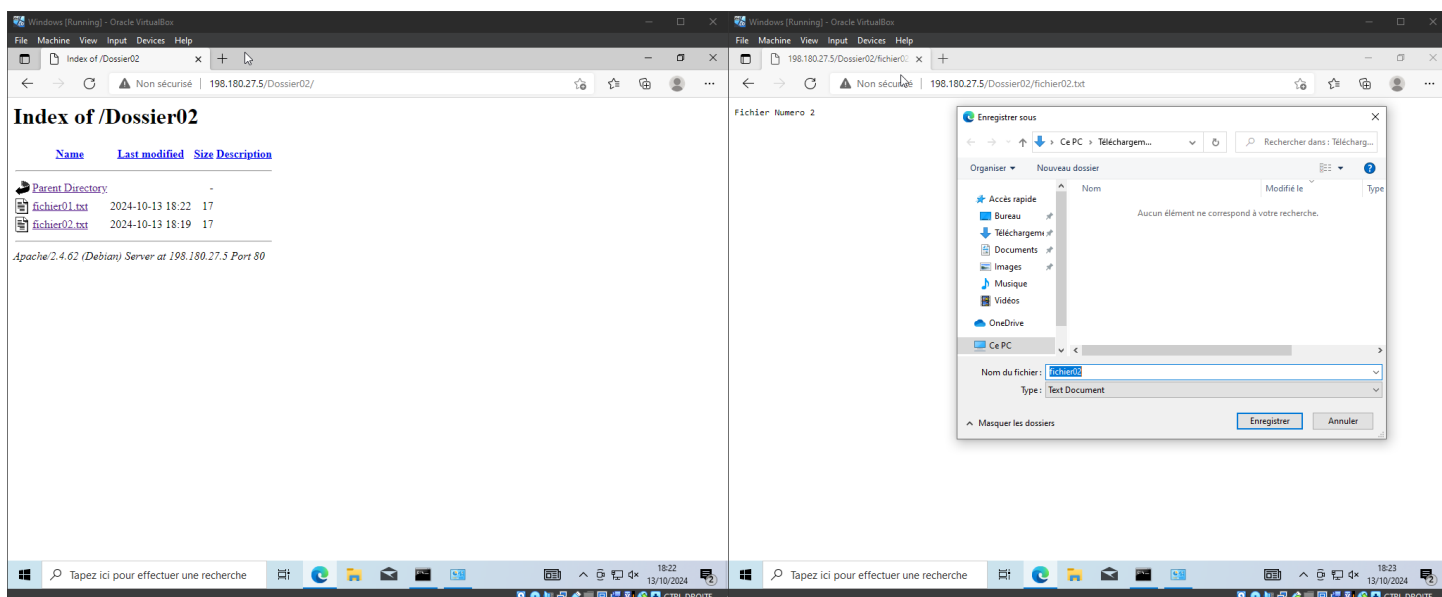
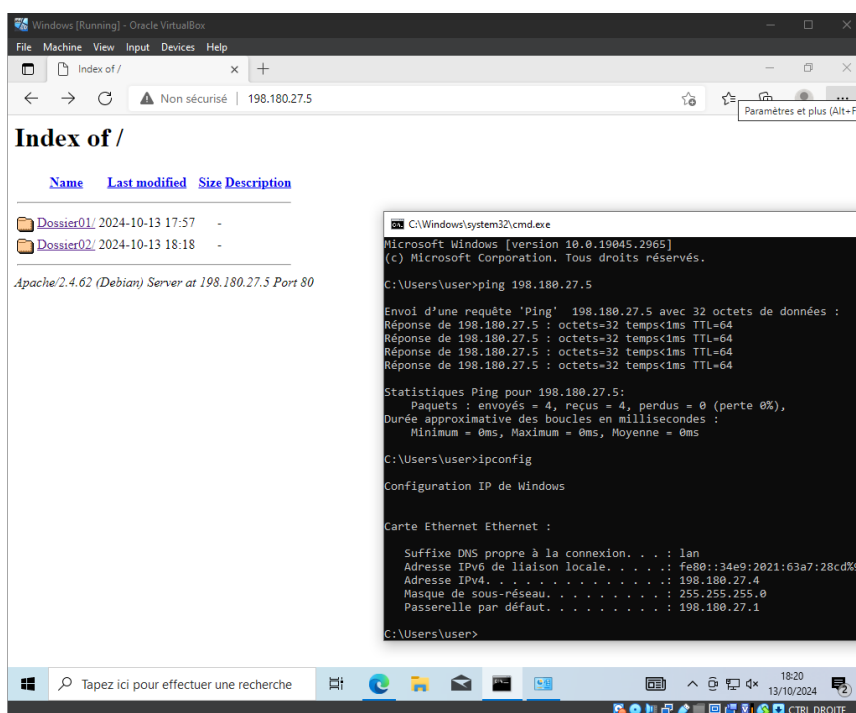
Dossier02



Précédemment, nous avons connecté nos deux VM en utilisant un réseau NAT. Nous gardons cette mise en place réseau pour pouvoir continuer à faire communiquer nos deux ordinateurs.

Nous venons donc de mettre tout en place pour le bon fonctionnement d'un transfert de fichier entre Windows et Debian à travers Apache.

Pour cela, nous pouvons lancer notre navigateur sur Windows et se connecter à notre machine Debian avec l'IPv4. Nous allons ainsi tomber sur le répertoire de notre serveur Apache avec les deux dossiers. Il nous suffit d'ouvrir le dossier que nous souhaitons et d'enregistrer le fichier voulu.



## Partage de fichier en dehors d'Apache

### Linux vers Windows

Nous avons utilisé le package Samba, son installation est faite via le terminal en tant que super utilisateur. Samba permet de transférer et de modifier des fichiers entre Linux et Windows, ce package est l'un des plus populaires pour pouvoir utiliser des fichiers partagés.

```

debian@debian: ~$ su
Mot de passe :
root@debian:/home/debian# apt-get samba
E: L'opération samba n'est pas valable
root@debian:/home/debian# apt-get install samba
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 :
  attr ibverbs-providers libcephfs2 libfmt9 libgfs2 libgfrpc0 libgfsxdr0
  libglusterfs0 libibverbs1 librados2 librdmacm1 liburing2 python3-anyio
  python3-click python3-colorama python3-dnspython python3-gpg python3-h11
  python3-h2 python3-hpack python3-httpcore python3-httpx python3-hyperframe
  python3-ldb python3-markdown python3-markdown-it python3-mdurl
  python3-pygments python3-requests-toolbelt python3-rfc3986 python3-rich
  python3-samba python3-sniffio python3-talloc python3-tdb python3-yaml
  samba-ad-provision samba-common samba-common-bin samba-dsdb-modules
  samba-vfs-modules tdb-tools
Paquets suggérés :
  python3-trio python3-aioquic python-markdown-doc python-pygments-doc
  ttf-bitstream-vera bind9 bind9utils ctdb ldb-tools ntp | chrony ufw winbind
  heimdal-clients
Les NOUVEAUX paquets suivants seront installés :
  attr ibverbs-providers libcephfs2 libfmt9 libgfs2 libgfrpc0 libgfsxdr0

```

Modification du fichier de configuration samba pour pouvoir mettre en place le 'dossier partagé'. Celui-ci se trouvant dans /etc/samba/smb.conf.

```

root@debian:/home/debian# rm /etc/samba/smb.conf
root@debian:/home/debian# nano /etc/samba/smb.conf

```

```

GNU nano 7.2 /etc/samba/smb.conf
[Processus]
  path=/home/debian/
  browseable=yes
  readonly=off

```

Définition du chemin d'accès du répertoire de partage et définition des droits de l'utilisateur, c'est-à-dire qu'on puisse consulter et modifier les fichiers depuis une autre machine.

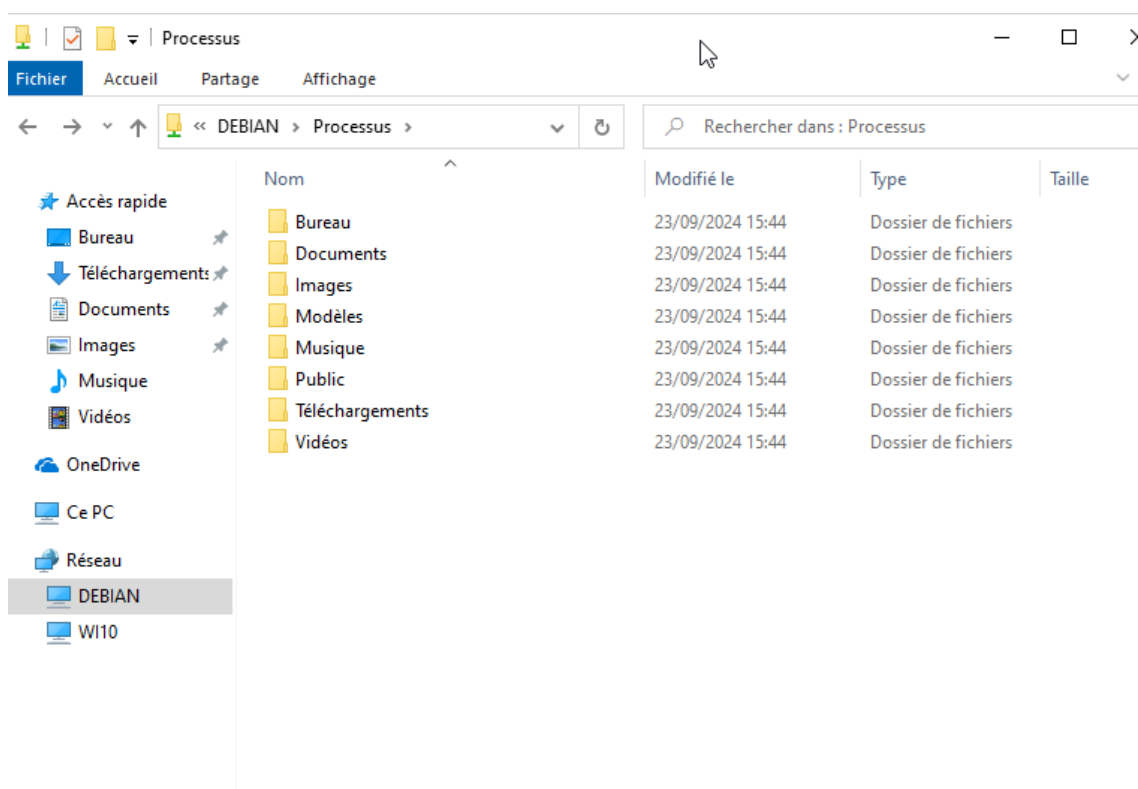
Création d'un utilisateur pour pouvoir se connecter via Windows.

Dans notre cas, le nom d'utilisateur est « root » et le mot de passe est aussi « root ». Pour que cela prenne effet, on redémarre le service Samba.

```
root@debian:/home/debian# smbpasswd -a root
New SMB password:
Retype new SMB password:
Added user root.
root@debian:/home/debian# /etc/init.d/smbd restart
Restarting smbd (via systemctl): smbd.service.
```

Notre partage de fichier est donc maintenant mis en place sur Linux, nous pouvons ainsi y accéder depuis Windows.

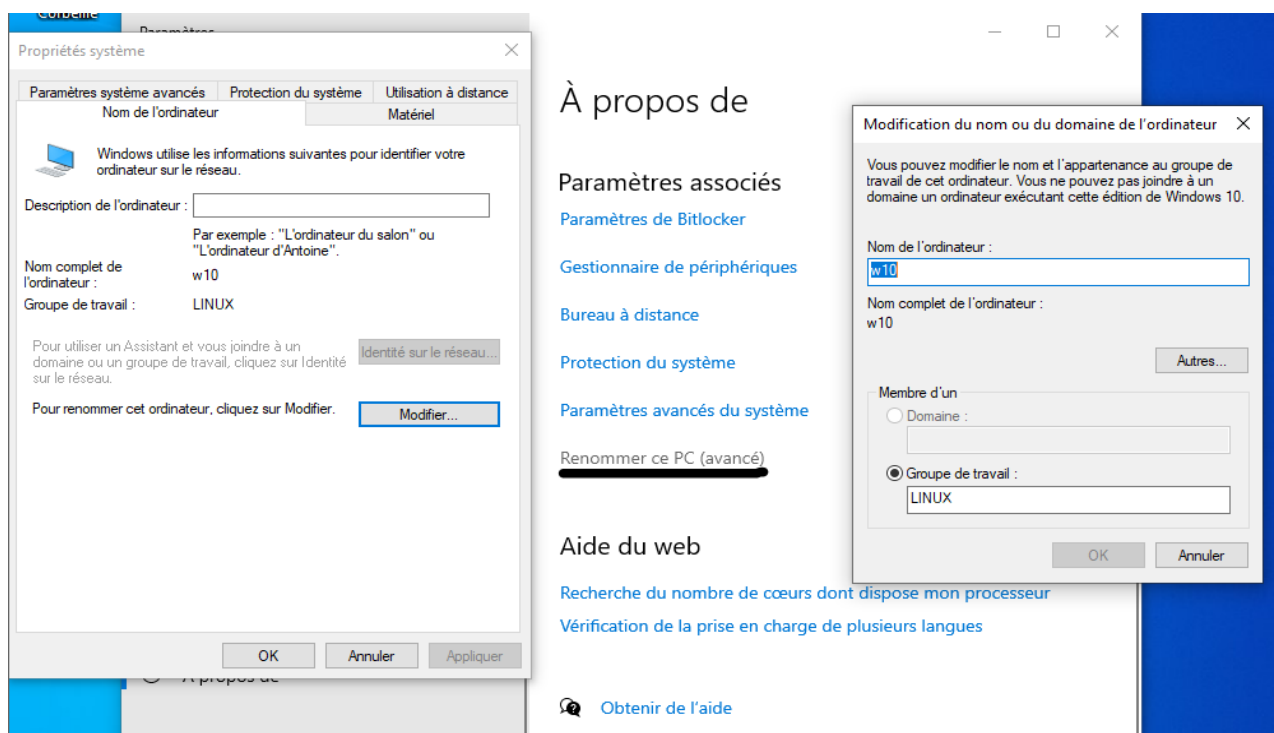
Pour cela dans la catégorie réseaux de notre explorateur de fichier, nous cliquons sur la machine Linux suivis des logins que nous avons définis précédemment. On a ainsi directement accès au dossier /home/ de la machine Debian.



## Windows vers Linux

Pour transférer des fichiers de Windows vers Linux, on doit d'abord créer un groupe de travail. Cette action est située dans la section pour modifier le nom de la machine sous Window. En cochant la case « Groupe de travail » et en lui attribuant un nom nous venons de mettre en place le groupe de travail.

Bien penser à retirer le pare-feu pour éviter tout problème de connexion lors des tests.



Nous en profitons pour récupérer l'adresse IPv4 de la machine.

```
Carte Ethernet Ethernet1 :

Suffixe DNS propre à la connexion. . . : localdomain
Adresse IPv6 de liaison locale. . . . : fe80::4c1f:144c:5ca0:91ec%3
Adresse IPv4. . . . . : 192.168.139.128
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . :

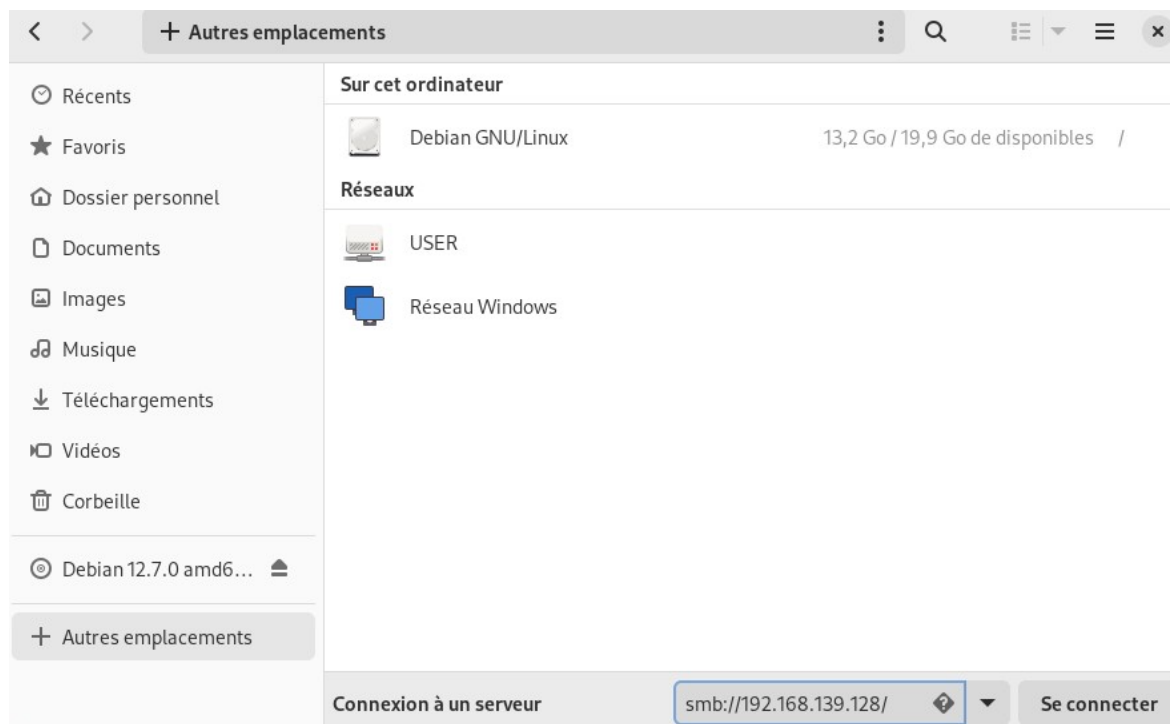
C:\Users\user>
```



Il ne nous reste plus qu'à ajouter le nouveau dossier partagé sous Linux.

À l'aide de l'explorateur de fichier Linux, on accède à 'autre emplacement' et on rentre alors l'adresse IPv4 de Windows avec l'aide de Samba pour s'y connecter.

La machine va alors demander un user et un mot de passe, c'est celui que nous avons défini sur la machine Windows lors de sa configuration.



## Controlling Linux from a Windows VM using SSH

We would like to be able to access Linux from Windows to carry out Linux commands. The connection must be secure, so we will be using an SSH connection.

To do so, we need to set up an SSH link on Debian.

First, we go to the terminal to install OpenSSH with the following command :

« `sudo apt-get install openssh-server` »

```
mainuser@vbox: ~
mainuser@vbox:~$ mainuser
bash: mainuser : commande introuvable
mainuser@vbox:~$ sudo apt-get install openssh-server
[sudo] Mot de passe de mainuser :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
openssh-server est déjà la version la plus récente (1:9.2p1-2+deb12u3).
openssh-server passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
mainuser@vbox:~$
```

Then to check if the SSH server is up and running, we will use another command to get the status of the service.

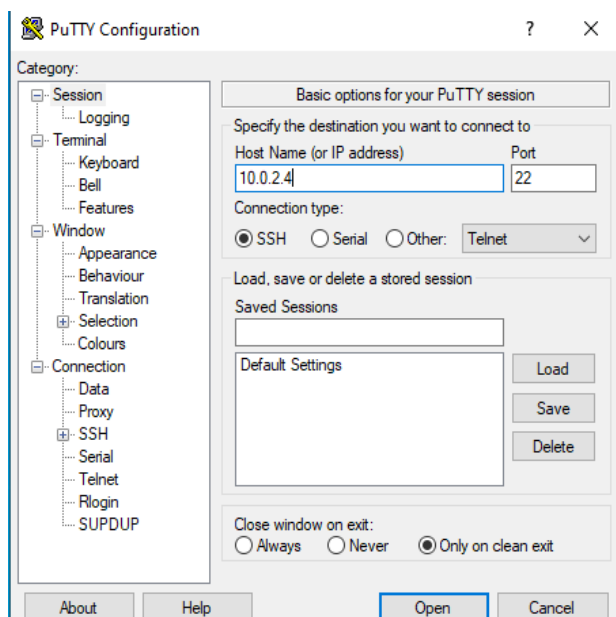
« `sudo systemctl status sshd` »

```
user@user: ~
user@user:~$ sudo systemctl status sshd
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-10-17 21:23:37 CEST; 4min 5s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 579 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 609 (sshd)
    Tasks: 1 (limit: 2293)
   Memory: 5.4M
      CPU: 31ms
  CGroup: /system.slice/ssh.service
          └─609 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

oct. 17 21:23:37 user systemd[1]: Starting ssh.service - OpenBSD Secure Shell s>
oct. 17 21:23:37 user sshd[609]: Server listening on 0.0.0.0 port 22.
oct. 17 21:23:37 user sshd[609]: Server listening on :: port 22.
oct. 17 21:23:37 user systemd[1]: Started ssh.service - OpenBSD Secure Shell se>
lines 1-17/17 (END)
```

We see here that the SSH connection is active, furthermore we can also get the port the SSH server is listening to, which is port 22.

Now that the SSH server is setup on the Debian VM, we can get on our Windows counterpart to establish the connection through the SSH server using PuTTY.



Now that the SSH server is set up on the Debian VM, we can get on our Windows counterpart to establish the connection through the SSH server using PuTTY.

The software is represented as followed, we can start our link between Debian and Windows with SSH.

To establish the connection we need to put the IP in our Debian VM which was set up as 10.02.2.4

The correct port is the default one, 22.

```
login as: mainuser
mainuser@10.0.2.4's password:
Linux vbox 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) 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: Thu Oct 17 20:00:49 2024 from 10.0.2.15
mainuser@vbox:~$
```

Succeeding the opening of the connection, we need to log in as our Debian user. In our VM the login is « mainuser » and the password is « motdepasse ».

We are now successfully connected to Debian with an SSH connection from a Windows machine.

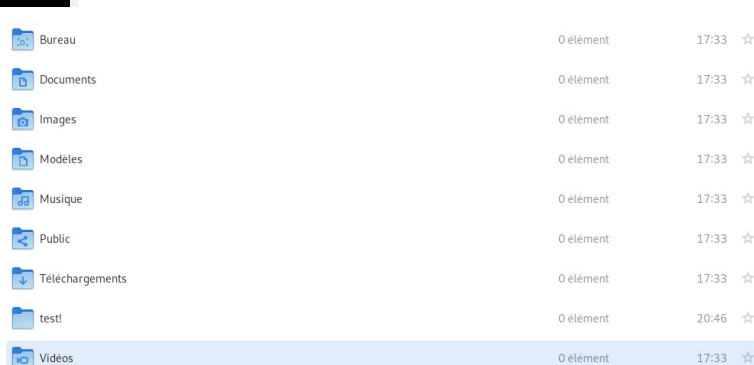
We can now dig into the usage of commands to interact with Debian.

```
mainuser@vbox: ~
login as: mainuser
mainuser@10.0.2.4's password:
Linux vbox 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

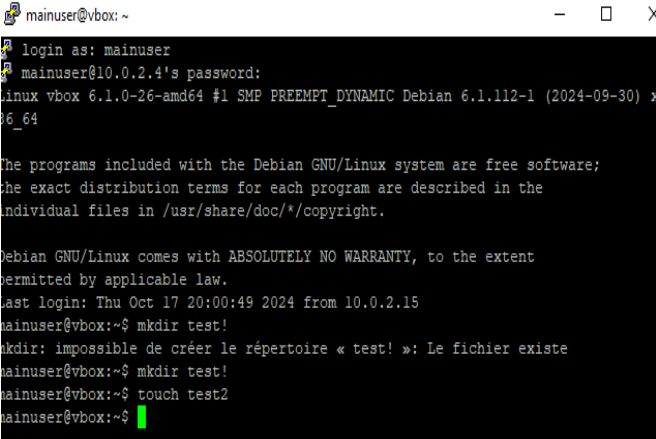
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct 17 20:00:49 2024 from 10.0.2.15
mainuser@vbox:~$ mkdir test!
mkdir: impossible de créer le répertoire « test! »: Le fichier existe
mainuser@vbox:~$ mkdir test!
mainuser@vbox:~$
```

By typing on Windows the command :  
mkdir folder\_name, it will create a folder in the Debian Virtual machine. By default, in the home directory.



We will proceed by creating a text file with the « touch file\_name» command.



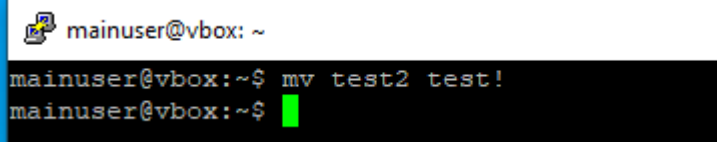
```
mainuser@vbox: ~
login as: mainuser
mainuser@10.0.2.4's password:
Linux vbox 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

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.

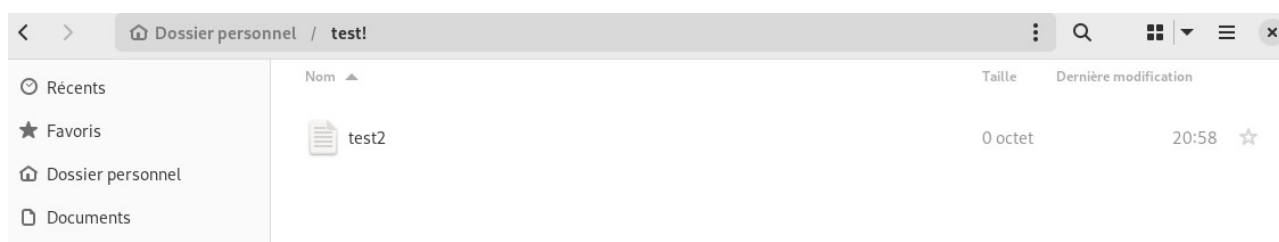
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct 17 20:00:49 2024 from 10.0.2.15
mainuser@vbox:~$ mkdir test!
mkdir: impossible de créer le répertoire « test! »: Le fichier existe
mainuser@vbox:~$ mkdir test!
mainuser@vbox:~$ touch test2
mainuser@vbox:~$
```

To finish our little experiment with SSH we will move the text file we just created into the folder we made.

We will do so using « mv files\_path folder\_path », in our example it will be « mv test2 test! »



```
mainuser@vbox: ~
mainuser@vbox:~$ mv test2 test!
mainuser@vbox:~$
```



## Problème rencontré

Au cours de ce projet, nous avons pu rencontrer divers problèmes. Lors de la première séance, durant l'installation de Debian avec le paquet nécessitant une connexion Internet, le premier problème est apparu. Impossible de se connecter à Internet. Nous allons apprendre que c'était dû au Proxy de l'IUT, mais nous avons quand même essayé de trouver une solution.

Ainsi, en recherchant des causes possibles, une solution viendrait d'un problème de miroir des packages de Debian qui n'aurait pas été bien inscrit dans le fichier correspondant.

C'est ainsi que nous sommes allés modifier sans interface graphique le fichier "sources.list" se situant dans "/etc/apt".

Logiquement, cette modification, n'a pas résolu le problème rencontré.

Il suffisait de mettre le proxy suivant lors de l'installation : proxy1-rech.uphf.fr:3128

## Conclusion

Ce projet a pu nous apporter les connaissances rudimentaires sur le fonctionnement d'un système réseau. À travers nos différents problèmes rencontrés, nous avons pu trouver des solutions et mettre en place les divers moyens de communication nécessaire entre nos deux machines virtuelle.

En tant potentiel futur étudiant en parcours B, cette SAE est un passage nécessaire et important pour la suite de nos études.

Nous vous remercions du temps que vous avez pris pour nous lire.

Burgnies Axel,  
Collot Grégoire,  
Marecaille-Henaut Mattieu,  
Macrez Mathis.