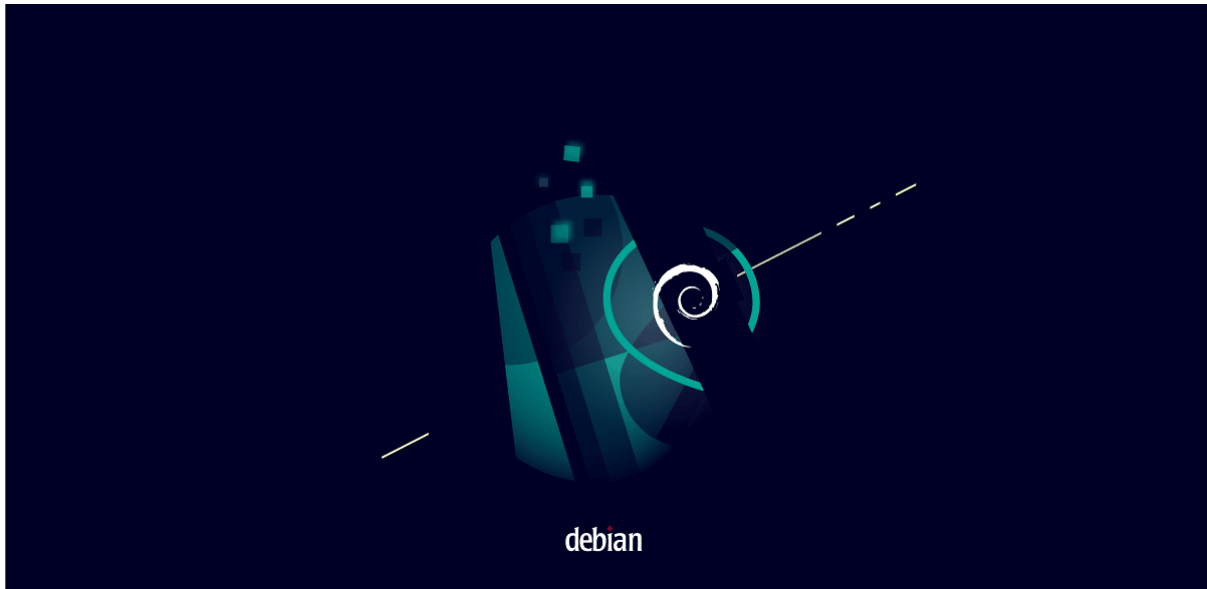


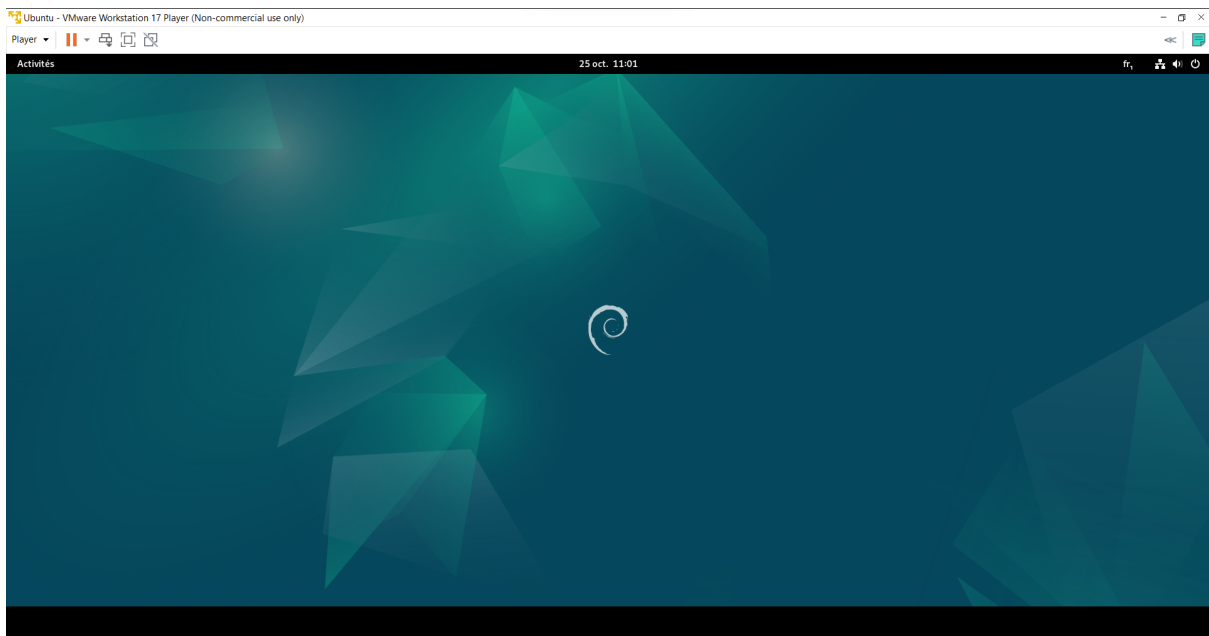
DDWS

Job 01

Installation de debian 12 avec l'interface graphique:



Une fois toutes les étapes finalisé, on apparaît sur le bureau de la VM:



Job 02

Pour commencer, installer un serveur Web Apache2.

À la fin de votre installation, vous devriez avoir sur votre navigateur une page comme suit :

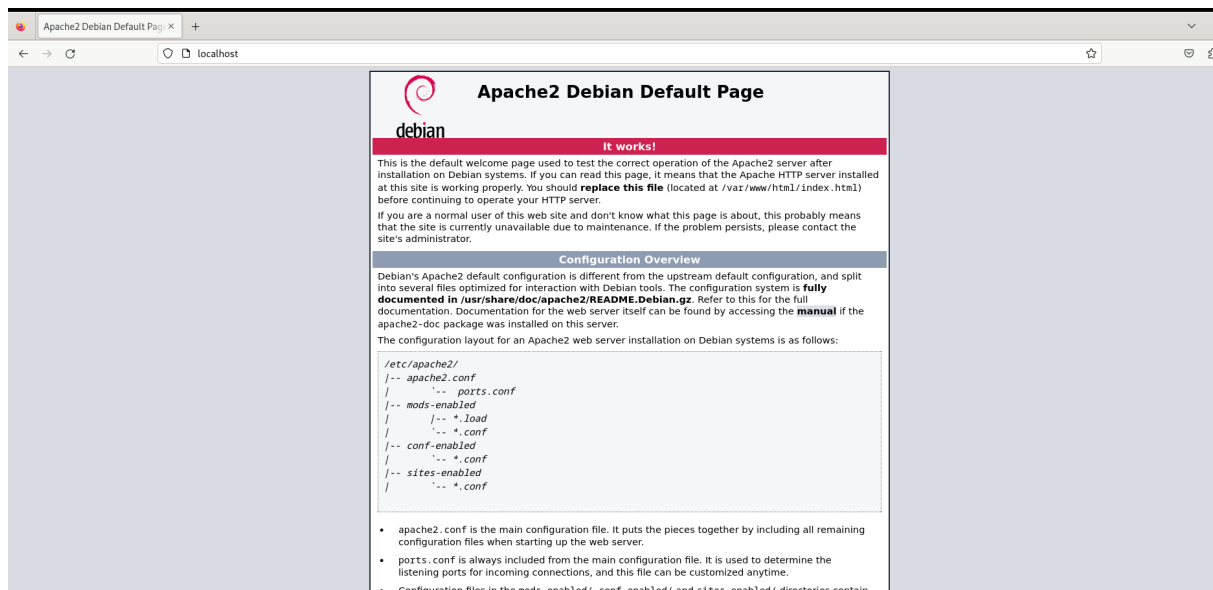
Pour l'installation on va utiliser plusieurs commande:

On va tout d'abord mettre à jour les paquets actuelles = `sudo apt update`

Ensuite on installe apache2 avec = `sudo apt install apache2`

Pour vérifier son installation et son activité nous faisons = `sudo systemctl status apache2`

Pour finaliser on ouvre votre navigateur avec l'URL de l'hôte et on voit s'il s'affiche.



Job 03

Apache HTTP Server (Apache)

- Avantages:

- Très stable et fiable.
- Modulaire et extensible avec de nombreux modules tiers.
- Bonne documentation et grande communauté d'utilisateurs.

- Inconvénients:

- Peut être un peu plus lourd en termes de ressources comparé à certains serveurs plus récents.

Nginx

-Avantages:

- Excellente performance, particulièrement pour servir des contenus statiques.

- Faible utilisation des ressources.
- Conçu pour être scalable (peut gérer un grand nombre de connexions simultanées).

-Inconvénients:

- Moins adapté pour des scénarios où de nombreux modules sont nécessaires (comme Apache).

LiteSpeed

- Avantages:

- Performances élevées, comparable à Nginx.
- Interface web conviviale pour la gestion.

- Inconvénients:

- La version gratuite a des fonctionnalités limitées.

Caddy

- Avantages:

- Configuration automatique avec Let's Encrypt pour le support HTTPS.
- Facile à configurer.

- Inconvénients:

- Peut être moins flexible pour des configurations avancées.

Cherokee

- Avantages:

- Interface web conviviale.
- Bonne performance.

-Inconvénients:

- Communauté plus petite que certains autres serveurs.

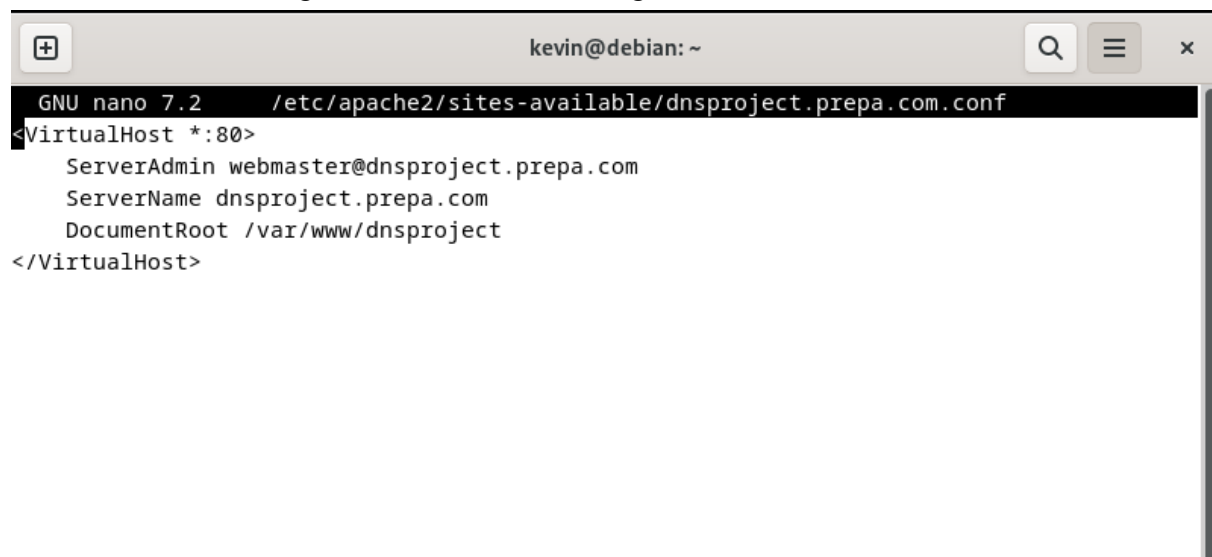
Chacun de ces serveurs a ses propres cas d'utilisation privilégiés en fonction des besoins spécifiques. Le choix dépend souvent des exigences de performance, de la complexité de la configuration, et des préférences personnelles. Si tu as des besoins spécifiques ou des critères particuliers, n'hésite pas à les mentionner pour obtenir des recommandations plus ciblées.

Job 04

*Mettez en place un DNS sur votre serveur Linux qui fera correspondre l'adresse IP de votre serveur au nom de domaine local suivant : "dnsproject.prepa.com"
Votre serveur devra donc pouvoir se ping via ce nom de domaine.*

Pour arriver à ce résultat nous devons dans un premier temps configurer le serveur apache pour notre site web dnsproject.prepa.com.

Nous faisons = `sudo nano /etc/apache2/sites-available/dnsproject.prepa.com`
Puis nous le configurons avec la configuration de base:



```
kevin@debian: ~  
GNU nano 7.2 /etc/apache2/sites-available/dnsproject.prepa.com.conf  
<VirtualHost *:80>  
    ServerAdmin webmaster@dnsproject.prepa.com  
    ServerName dnsproject.prepa.com  
    DocumentRoot /var/www/dnsproject  
</VirtualHost>
```

On enregistre ceci, on active le serveur et on le redémarre pour effectuer les modifications avec les commandes suivantes:

Activé = `sudo a2ensite dnsproject.prepa.com`
Pour activer la nouvelle configuration avant le redémarrage = `sudo systemctl reload apache2`
Le redémarrage = `sudo systemctl restart apache2`

Pour en finir avec apache2 (notre serveur HTTP) on ouvre le fichier host avec la commande = `sudo nano /etc/hosts` et on ajoute notre nom de domaine liée à l'ip de la machine car nous voulons afficher sur le web.

```
kevin@debian: ~  
GNU nano 7.2 /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    debian  
192.168.8.130 dnsproject.prepa.com  
  
# The following lines are desirable for IPv6 capable hosts  
::1    localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

On peut vérifier tout ce que l'on vient de faire en faisant un ping du domaine et voir si ça répond, c'est un petit test à faire pour l'avancement et être sûr que ça marche.

```
kevin@debian: ~  
To activate the new configuration, you need to run:  
systemctl reload apache2  
root@debian:/home/kevin# systemctl reload apache2  
root@debian:/home/kevin# sudo a2ensite dnsproject.prepa.com.conf  
Site dnsproject.prepa.com already enabled  
root@debian:/home/kevin# sudo systemctl restart apache2  
root@debian:/home/kevin# sudo nano /etc/hosts  
root@debian:/home/kevin# ping dnsproject.prepa.com  
PING dnsproject.prepa.com (127.0.0.1) 56(84) bytes of data.  
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.025 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.030 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.032 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.023 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.029 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.026 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.026 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.063 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.026 ms  
^C  
--- dnsproject.prepa.com ping statistics ---  
9 packets transmitted, 9 received, 0% packet loss, time 8196ms  
rtt min/avg/max/mdev = 0.023/0.031/0.063/0.011 ms  
root@debian:/home/kevin#
```

Maintenant que le serveur HTTP apache2 est prêt, on va s'occuper d'installer le serveur DNS pour lire et traduire le nom de domaine et l'adresse IP.

On installe donc bind9 avec toujours pareil = `sudo apt install bind9`

On vérifie son fonctionnement avec = `sudo sytemctl status bind9`

On commence la configuration du serveur a commencant par se rentre dans le fichier Option = `sudo nano /etc/bind/named.conf.options` pour une configuration classique d'un serveur DNS comme ceci:

```
GNU nano 7.2 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    recursion yes;
    allow-recursion { localhost; };
    allow-transfer { none; };
    allow-query { any; };

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};
```

Puis l'étape essentiel pour le fonctionnement du DNS, crée un zone comme ceci:

```
GNU nano 7.2 /etc/bind/named.conf.local
zone "dnsproject.prepa.com" {
    type master;
    file "/etc/bind/db.dnsproject.prepa.com";
};
```

Cette va contenir le configuration principal du DNS, c'est la qu'on va lier des nom de domaine a l'ip du serveur pour qu'au moment de la recherche le nom de domaine affichera la page voulu. On fait comme ceci:

`sudo nano /etc/bind/db.tondomaine.com`

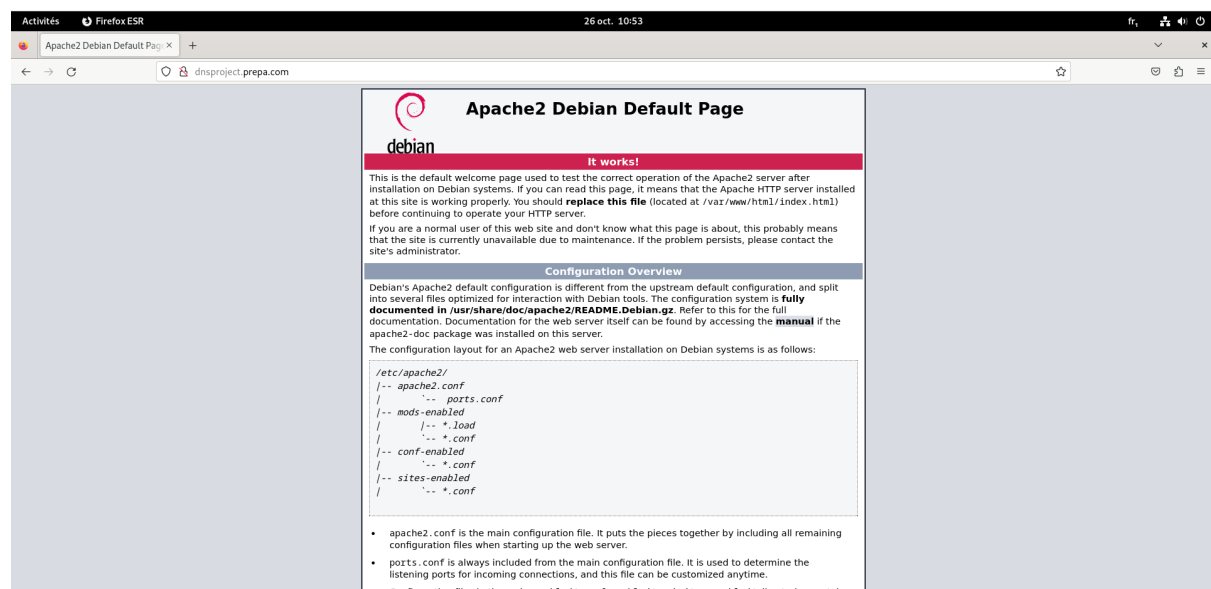
```
GNU nano 7.2 /etc/bind/db.dnsproject.prepa.com

; BIND data file for dnsproject.prepa.com
;
$TTL      604800
@         IN      SOA      dnsproject.prepa.com  admin.dnsproject.prepa.com (
                                2023102501 ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200     ; Expire
                                604800 )    ; Negative Cache TTL
;
@         IN      NS       dnsproject.prepa.com
ns        IN      A        <192.168.8.130>

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^/ Aller ligne
```

Après tout ces configuration, on redémarre le serveur pour les appliquer = `sudo systemctl restart bind9`

Nous avons terminé, un ping du nom de domaine devrait également fonctionner et si on l'écrit le nom de domaine sur le web, la page s'affiche.



Job 05

Faites des recherches sur comment obtient-on un nom de domaine public ?

Pour réserver un nom de domaine, vous devez vous adresser à l'un des bureaux d'enregistrement de nom de domaine, aussi appelés « registrars ». Le bureau compétent varie selon l'extension du nom de domaine (ex : .fr, .com, .eu).

Quelles sont les spécificités que l'on peut avoir sur certaines extensions de nom de domaine ?

L'extension du nom de domaine peut être géographique ou générique. Elle se choisit en fonction de la disponibilité du nom de domaine, mais également en fonction des objectifs visés par le site Internet. Le choix d'une extension de nom de domaine n'est pas limité. Par exemple: .net, .fr, .com, .org

Job 06

Connectez votre hôte au nom de domaine local de votre serveur, pour que votre page apache soit accessible via ce même nom de domaine.

Nous l'avons fait précédemment dans le fichier host , où on n'a liée l'ip du domaine à celle de la machine et donc du serveur:

```
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    debian
192.168.8.130 dnsproject.prepa.com

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

Job 07

Mettez en place un pare-feu en utilisant ufw sur votre serveur principale de manière que votre hôte puisse accéder à la page apache par défaut, mais qu'il ne puisse plus ping votre serveur.

Tout d'abord installons un Par-feu tel que ufw:

```
sudo apt install ufw
```

Vérifie son statut pour savoir si il est bien installé:

```
sudo ufw status
```

Le but étant je rappelle de paramétrer le Par-feu de tel sorte que ce qu'on puisse accéder au site web "dnsproject.prepa.com" mais en bloquant le ping vers celui-ci.

Pour réaliser cela on va autoriser le port 80 qui est le port par défaut de communication du web non chiffré.

```
sudo ufw allow 80
```

On interdit le port 22 de fonctionner car il permet de faire communication distant comme le processus icmp (les ping).

```
sudo ufw deny 22
```

```
root@debian:/etc/ufw# sudo ufw status
```

```
Status: active
```

To	Action	From
--	-----	----
80	ALLOW	Anywhere
22	DENY	Anywhere
80 (v6)	ALLOW	Anywhere (v6)
22 (v6)	DENY	Anywhere (v6)

Maintenant on va rentrer dans le fichier become.rules pour mettre en DROP les configuration concernant l'ICMP.

```
sudo nano /etc/ufw/before.rules
```

```

GNU nano 7.2 /etc/ufw/before.rules
# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines

# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT

# quickly process packets for which we already have a connection
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

# allow dhcp client to work
-A ufw-before-input -p udp --sport 67 --dport 68 -j ACCEPT

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C EmplacementM-U Annuler
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligneM-E Refaire

```

En faisant tout ça, ça devrait bloquer les pings. Comme ci-dessous:

Job 08

Mettez en place sur votre serveur un dossier partagé avec les autres membres de votre réseau (soit là où les autres machines virtuelles). Ils pourront partager des fichiers dans ce dossier, ainsi que récupérer des fichiers depuis ce dossier.

Ce dossier doit être accessible dans votre gestionnaire de fichier en interface graphique.

Pour cela nous allons installer samba qui est utilisée pour créer des dossiers de partage. Pour commencer l'installation:

```
sudo apt update
sudo apt install samba
```

Une fois le serveur de partage installé, on va le configurer pour créer une liaison sur le réseau local, on va dans un fichier de configuration spécifique et on met les paramètres que j'ai mis:

```
sudo nano /etc/samba/smb.conf
```

On ajoute que la dernière partie on c'est écrit "partage"

```
GNU nano 7.2 /etc/samba/smb.conf
# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
    comment = Printer Drivers
    path = /var/lib/samba/printers
    browseable = yes
    read only = yes
    guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[partage]
    path = /home/kevin/partage
    browseable = yes
    read only = no
```

Maintenant que nous avons bien configuré ceci, on crée le dossier de partage en respectant le même chemin que dans la configuration: `sudo mkdir /home/kevin/partage`

Un fois les deux étapes assez simple terminé , nous redémarrant le service samba avec = `sudo service smb restart`

Pour vérifier que ceci fonctionne, on va sur une autre machine (VM ou la machine physique) on tape : `\\ip_du_notre_VM\partage` dans la barre de recherche de l'explorateur de fichier.

On va vous demander normalement l'identifiant et le mot de passe de votre machine virtuelle puis vous serez dans le dossier comme ici:

