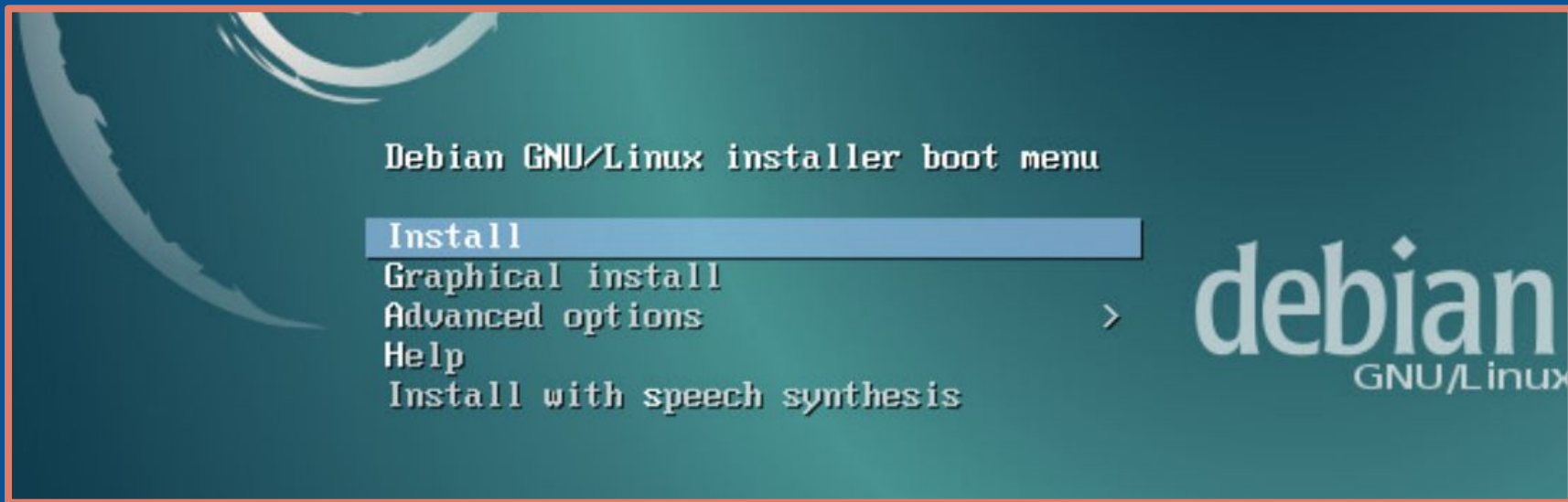




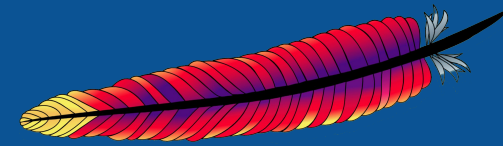
DDWS

Job 01



Pour installer Debian, on récupère une iso de debian que l'on charge avec VMWARE. On choisit l'installtion graphique. Une fois configuré, Debian est prête à être utilisé.

Job 02



Pour installer Apache2 sur Debian, on ouvre le terminal et on met à jour le gestionnaire de paquet en utilisant :

sudo apt-get update

(Puis on en profite pour mettre à jour les paquets non à jour)

sudo apt-get upgrade

```
lucas@lucas:~$ sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
  linux-image-amd64
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.
```

Une fois les paquets à jour, on utilise :

sudo apt-get install apache2

```
Les paquets supplémentaires suivants seront installés :
  apache2-data apache2-utils
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-data apache2-utils
0 mis à jour, 3 nouvellement installés, 0 à enlever et 1 non mis à jour.
```

Job 03

Un serveur Web est un programme qui utilise le protocole HTTP pour fournir les fichiers qui constituent les pages Web en répondant aux demandes des clients (navigateur) en leur fournissant les fichiers nécessaires pour afficher les pages. Les utilisateurs accéderont à ces pages en entrant l'adresse IP (Ou le DNS) dans la barre de recherche du navigateur.

Il existe de nombreux serveur web comme **Apache** (le serveur Web le plus répandu), **IIS** (Internet Information Server) de Microsoft ou encore **Nginx**.

Exemple de serveur Web	Avantage	Inconvénient
Apache	<ul style="list-style-type: none">- Stable et éprouvé.- Grande communauté d'utilisateurs et de développeurs.- Supporte de nombreux modules et extensions.	<ul style="list-style-type: none">- Moins performant que d'autres serveurs web pour les charges très élevées.- Configuration peut sembler complexe pour les débutants.
IIS	<ul style="list-style-type: none">-Intégration transparente avec d'autres produits Microsoft.- Prise en charge d'ASP.NET et d'autres technologies Microsoft.	<ul style="list-style-type: none">- Principalement destiné aux systèmes Windows, ce qui limite la portabilité.- Moins couramment utilisé pour l'hébergement web non-Microsoft.
Nginx	<ul style="list-style-type: none">- Excellente performance, notamment pour les charges élevées et les connexions simultanées.- Gestion efficace des serveurs proxy et des répartiteurs de charge.	<ul style="list-style-type: none">- Configuration moins intuitive pour les débutants.- Moins de modules que ceux disponibles pour Apache.

Job 04



Afin de créer notre DNS, on installe le logiciel Bind9 qui nous permet de l'utiliser en tant que serveur DNS. On utilise donc la commande :

sudo apt-get install bind9

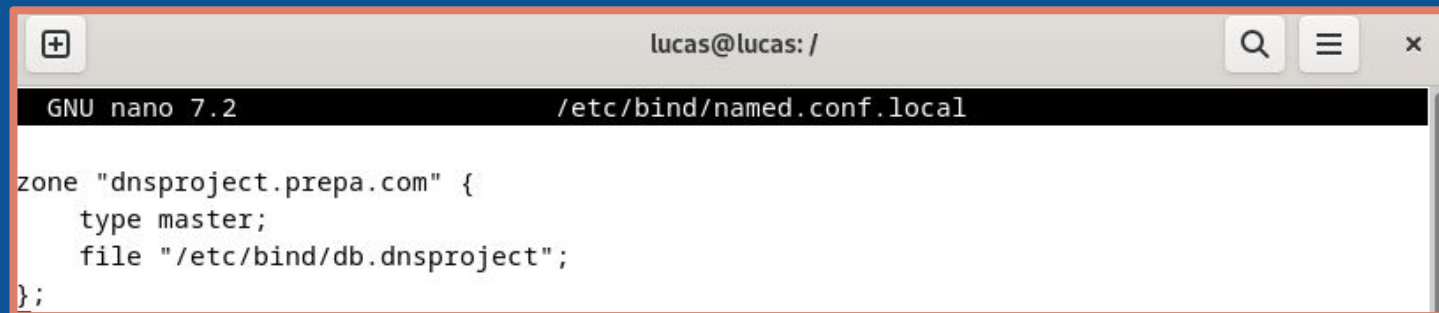
On crée ensuite un fichier dans **/etc/bind/** nommé **db.dnsproject**

On renseigne dans les dernières lignes le nom de domaine souhaité et l'adresse IP de notre serveur.

```
GNU nano 7.2 /etc/bind/db.dnsproject *
## /etc/bind/db.dnsproject pour dnsproject.prepa.com
$TTL      604800
@         IN      SOA      dnsproject.prepa.com. admin.dnsproject.prepa.com. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
@         IN      NS       dnsproject.prepa.com
@         IN      A        192.168.154.128
```

Job 04

Une fois cette configuration terminée, on la rajoute dans les listes de configuration de bind9 en editant le **fichier named.conf.local**



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

Job 05

Afin d'obtenir un nom de domaine public, on doit d'abord s'adresser à un registrar (Bureau d'enregistrement gérant la reservation de nom de domaine). Il faut ensuite vérifier la disponibilité du nom de domaine souhaité. Si ce n'est pas le cas, il faudra modifier le nom de domaine ou l'extension. On choisit ensuite l'extension comme .com, .org, .net, .io, .fr ... Enfin, on choisit notre nom de domaine.

Les extension de domaine possèdent différentes spécificités. Il en existe 2 types :

Les extensions “génériques”

.com par exemple est l'une des plus populaires et était associée à l'origine à des entreprises commerciales.

.org Traditionnellement associée à des organisations à but non lucratif, ou encore

.net qui était destinée aux entreprises de réseau

.io Bien qu'historiquement considéré comme une extension locale (Océan Indien), elle est principalement considéré comme une extension générique par sa large utilisation (surtout dans les domaines informatiques)

Aujourd'hui ces extensions peuvent être enregistrées par n'importe qui.

Les extensions “locales”

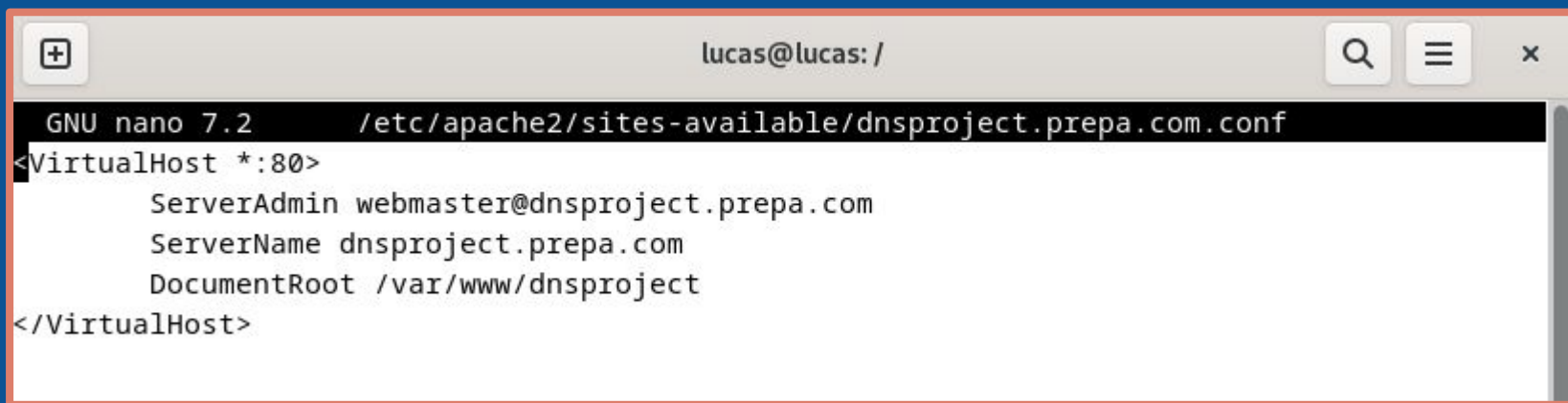
.fr / .co.uk / .ca / .eu / etc. Ce sont des extensions nationales qui sont spécifiques à une zone (France, Royaume-Uni, Canada, Europe etc.).

.edu et .gov sont réservées à l'enseignement supérieur et aux gouvernements aux Etats-unis

Job 06

Pour pouvoir accéder à notre site, je commence par configurer un hôte virtuel.

On commence par créer un document du nom de notre site dans le dossier **/etc/apache2/sites-available/** dans lequel on renseigne le nom de notre site ainsi que son fichier "root" où se trouvera son contenu.

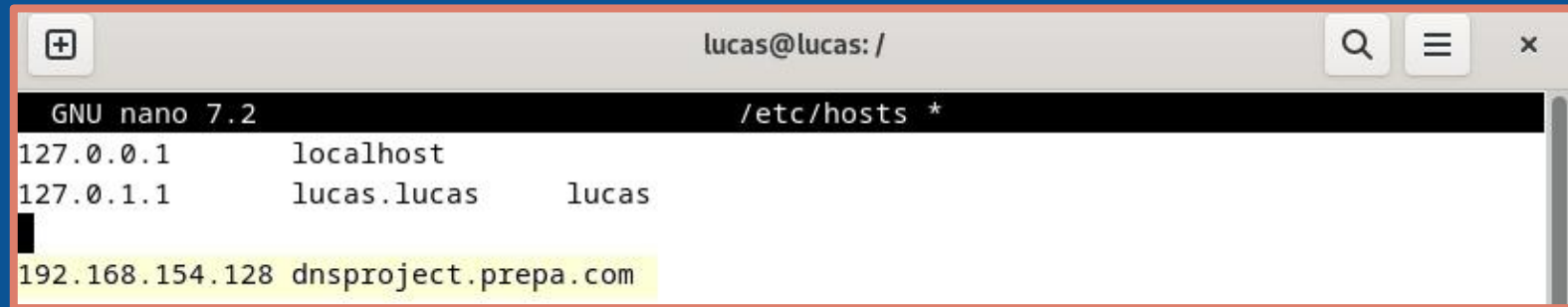


```
lucas@lucas: /  
GNU nano 7.2 /etc/apache2/sites-available/dnsproject.prepa.com.conf  
<VirtualHost *:80>  
    ServerAdmin webmaster@dnsproject.prepa.com  
    ServerName dnsproject.prepa.com  
    DocumentRoot /var/www/dnsproject  
</VirtualHost>
```

Une fois configuré, on active notre site en utilisant la commande **sudo a2ensite dnsproject.prepa.com.conf** *a2ensite (pour Apache2 Enable Site) suivi du nom du fichier de configuration*

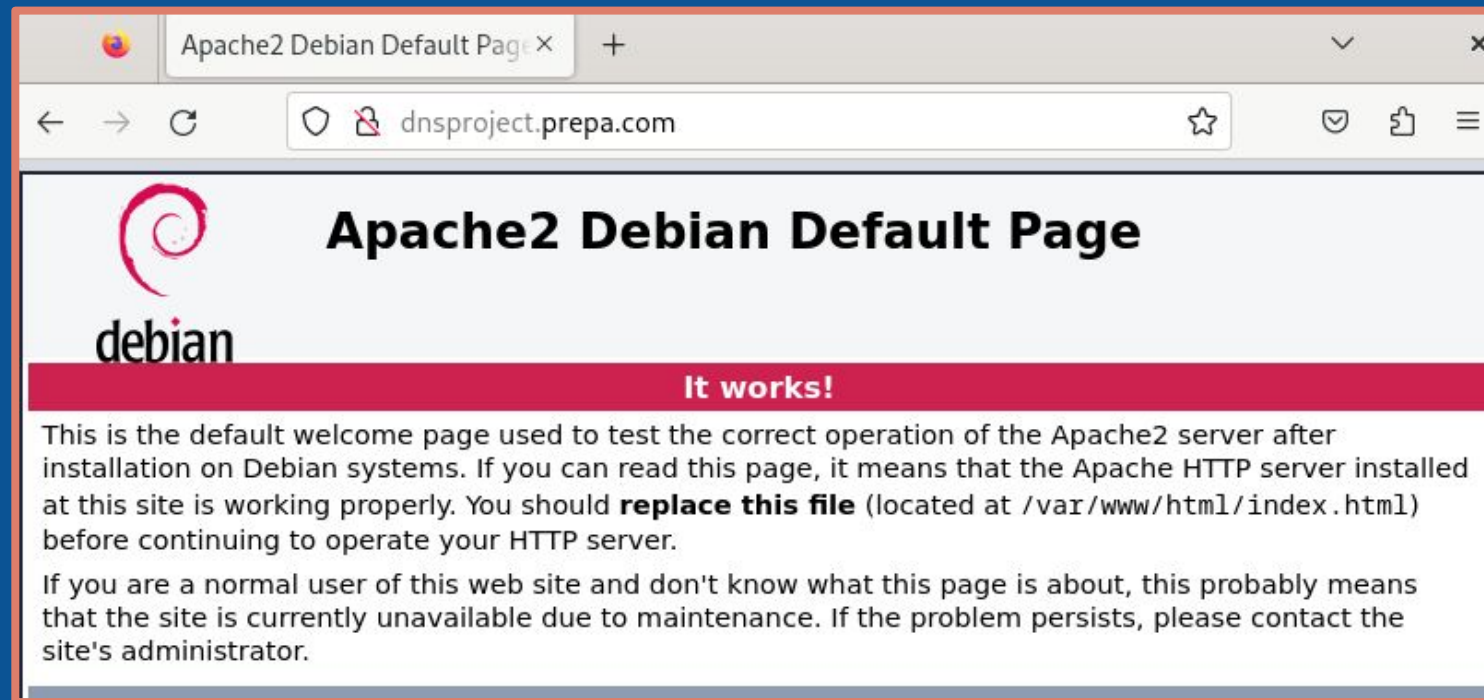
On redémarre ensuite notre logiciel apache2 en utilisant la commande: **sudo systemctl reload apache2**

Pour que le serveur puisse accéder à notre nouveau nom de domaine "dnsproject.prepa.com", on le renseigne dans notre fichier **/etc/hosts** précédé de l'IP de notre site.



```
lucas@lucas: /  
GNU nano 7.2 /etc/hosts *  
127.0.0.1 localhost  
127.0.1.1 lucas.lucas lucas  
192.168.154.128 dnsproject.prepa.com
```

On vérifie le bon fonctionnement de nos manipulations en entrant le nom de domaine dans notre navigateur.



Job 07



Pour installer le parefeu UFW, on utilise les commandes :

sudo apt-get update (si elle n'avait pas été effectué récemment) puis
sudo apt-get install ufw

Une fois le parefeu installé, on doit accorder certaines autorisations.
On utilise les commandes :

`sudo ufw allow 80` (Permet d'autoriser le trafic HTTP)
`sudo ufw allow 53` (Permet d'autoriser le trafic DNS)

```
lucas@lucas:~$ sudo ufw status
Status: active

To Action From
--
80 ALLOW Anywhere
53 ALLOW Anywhere
80 (v6) ALLOW Anywhere (v6)
53 (v6) ALLOW Anywhere (v6)
```



Enfin, pour bloquer les ping :

On doit modifier notre fichier de configuration du pare-feu “before.rules” et localiser les lignes de commandes qui ciblent les **icmp** particulièrement la section INPUT en modifiant la ligne de la manière suivante suivante:

- A ufw-before-input -p icmp --icmp-type echo-request -j DROP

On redémarre ensuite notre parefeu avec la ligne

sudo ufw disable && sudo ufw enable

```
lucas@lucas:~$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (192.168.154.128) 56(84) bytes of data.
^C
--- dnsproject.prepa.com ping statistics ---
20 packets transmitted, 0 received, 100% packet loss, time 19473ms
```

Job 08

SAMBA

On commence par installer samba avec la commande:
`sudo apt-get update && sudo apt-get install samba`

On créer ensuite notre dossier de partage en lui donnant les autorisations nécessaire
`mkdir /home/lucas/partage`
`sudo chmod -R 777 /home/lucas/partage`

On édite le fichier de configuration samba pour lui indiquer le chemin du dossier partager et en lui accordant les autorisations avec la commande :
`sudo nano /etc/samba/smb.conf`

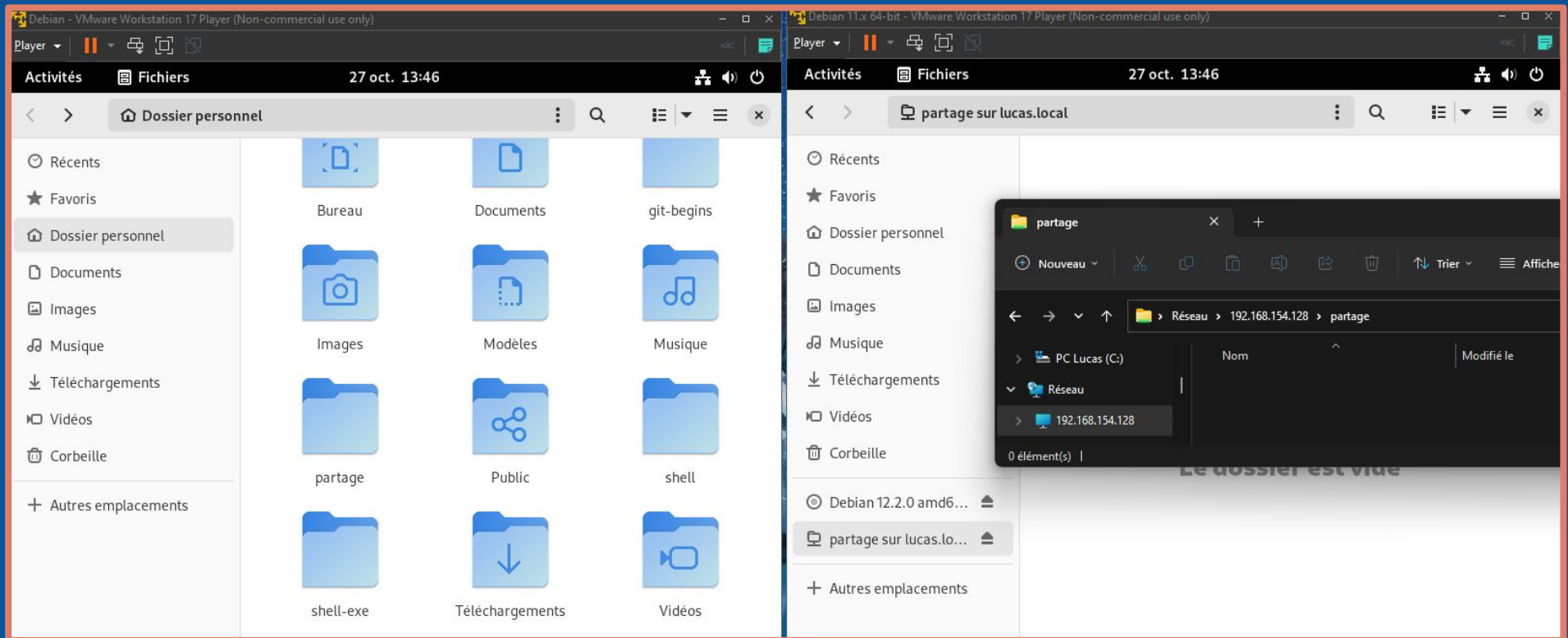
```
[partage]
  path = /home/lucas/partage
  browseable = yes
  read only = no
```

On crée un utilisateur en utilisant la commande :
`sudo smbpasswd -a lucas`
Puis on lui donne un mot de passe

Enfin, on redémarre samba en utilisant la commande :
`sudo /etc/init.d/smbd restart`

```
lucas@lucas:/$ sudo /etc/init.d/smbd restart
Restarting smbd (via systemctl): smbd.service.
```

Pour vérifier que cela fonctionne, on entre l'adresse ip de notre serveur dans la barre de recherche du gestionnaire de fichier suivi de /partage (sur Debian) ou alors en écrivant \\<notre adresse ip\partage (sur windows)
On entre ensuite le nom d'utilisateur et le mot de passe.



Notre dossier est bien accessible sur notre 2ème VM (à droite) ainsi que sur windows.