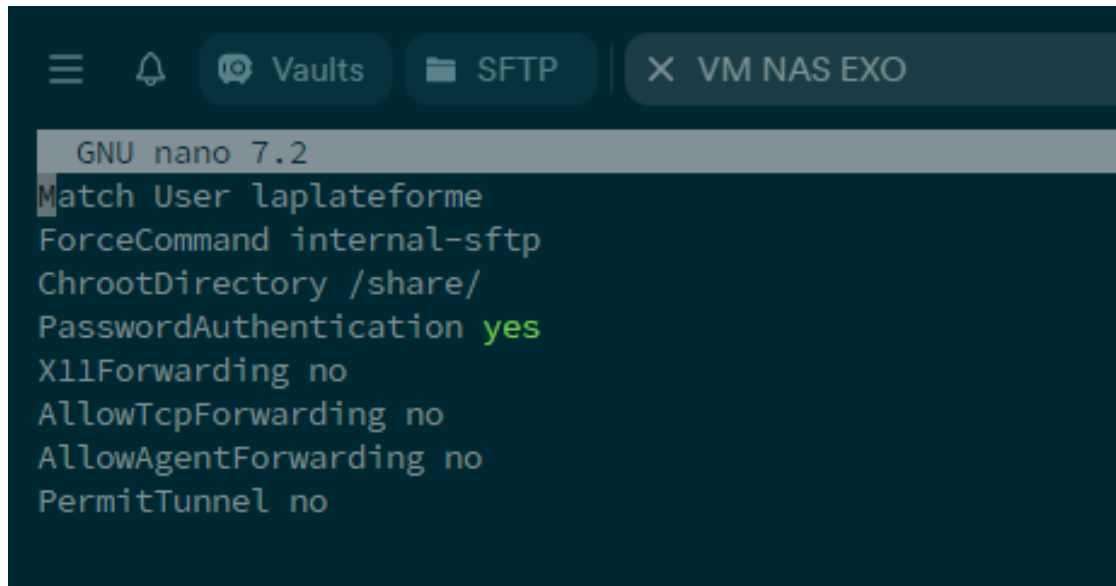


Documentation Serveur NAS

Création d'utilisateur en SFTP :

(L'utilisateur doit au préalable être créé dans le système nous allons juste parler de son fichier de configuration)

A screenshot of a terminal window with a dark background. At the top, there is a navigation bar with icons for a menu, a bell, a chat bubble labeled 'Vaults', a folder icon labeled 'SFTP', and a close button labeled 'VM NAS EXO'. Below the navigation bar, the terminal shows the prompt 'GNU nano 7.2' followed by the configuration for a user named 'laplateforme'. The configuration lines are: 'Match User laplateforme', 'ForceCommand internal-sftp', 'ChrootDirectory /share/', 'PasswordAuthentication yes' (where 'yes' is highlighted in green), 'X11Forwarding no', 'AllowTcpForwarding no', 'AllowAgentForwarding no', and 'PermitTunnel no'.

```
GNU nano 7.2
Match User laplateforme
ForceCommand internal-sftp
ChrootDirectory /share/
PasswordAuthentication yes
X11Forwarding no
AllowTcpForwarding no
AllowAgentForwarding no
PermitTunnel no
```

Le fichier de configuration doit être dans `/etc/ssh/sshd_config.d/<nomdutilisateur>.conf` pour plus de lisibilité, il est possible de tout mettre dans le fichier `/etc/ssh/sshd.conf` mais cela

n'est pas conseillé et peut poser problèmes en cas d'automatisation à la suppression d'un utilisateur par exemple.

Nous allons détailler ligne par ligne ce fichier de configuration:

Match User laplateforme → Cela détermine à quel utilisateur nous voulons appliquer cette configuration, dans le cas ici à l'utilisateur "laplateforme"

ForceCommand internal-sftp → Nous forçons l'utilisation de SFTP et non de SSH, car l'utilisateur ne doit pas avoir accès au shell mais uniquement au SFTP.

ChrootDirectory /share/ → On met la racine du SFTP au repertoire “/share/” qui contient le dossier Public et laplateforme qui appartient à l'utilisateur. Cela aussi nous permet de contenir l'utilisateur uniquement à ce dossier et au donnée qui sont dans ce dossier et non au reste du système.

PasswordAuthentication yes → On lui autorise la connexion par mot de passe, le SFTP pourrais supporter un système de clé mais pour des raison de facilité d'utilisation nous allons rester au mot de passe.

X11Forwarding no → On interdit l'exécution d'un serveur X et donc de l'affichage d'une interface graphique venant du serveur de la part de l'utilisateur.

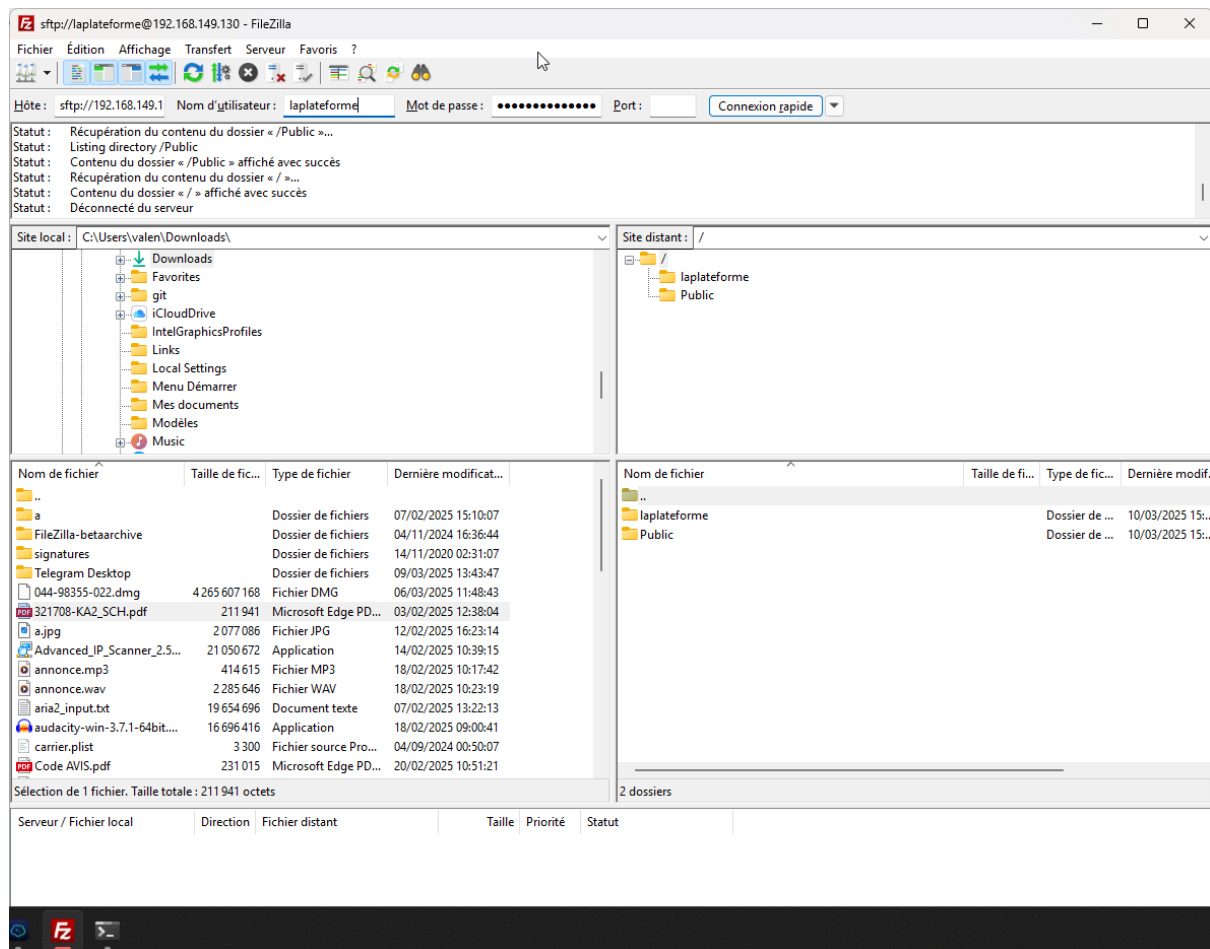
AllowTcpForwarding no → On interdit le transfert de port vers la machine hôte, l'utilisateur n'en a pas besoin pour le SFTP.

AllowAgentForwarding no → On interdit le transfert d'agent SSH

PermitTunnel no → On interdit la création de tunnel entre le serveur et la machine hôte.

Cela nous bloque bien la connection via SSH mais nous laisse accès via la SFTP

```
PS C:\Users\valen> ssh laplateforme@192.168.149.130
The authenticity of host '192.168.149.130 (192.168.149.130)' can't be established.
ED25519 key fingerprint is SHA256:2ldoKFYybfgVqCJoJXzJzLWeIC0moNOdrTGEhr2tUXE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.149.130' (ED25519) to the list of known hosts.
laplateforme@192.168.149.130's password:
This service allows sftp connections only.
Connection to 192.168.149.130 closed.
```



Création d'un utilisateur Samba:

Nous allons reprendre les informations de connexion utilisées pour le SFTP, mais cela permet de monter comme un disque réseau le partage de fichier sur les différents périphériques disponible dans le parc informatique avec plus de facilité pour l'utilisateur que le SFTP qui lui demande de passer par un logiciel tiers pour transférer ses fichiers et ne lui permettant pas d'utiliser ses fichiers directement sur le disque réseau.

Pour le samba, nous avons créé un groupe commun à tous les utilisateurs pour le dossier Public, puis nous avons créé les partages avec un partage pour le dossier publique uniquement accessible à tous les utilisateurs, et un partage pour l'utilisateur uniquement avec son dossier utilisateur.

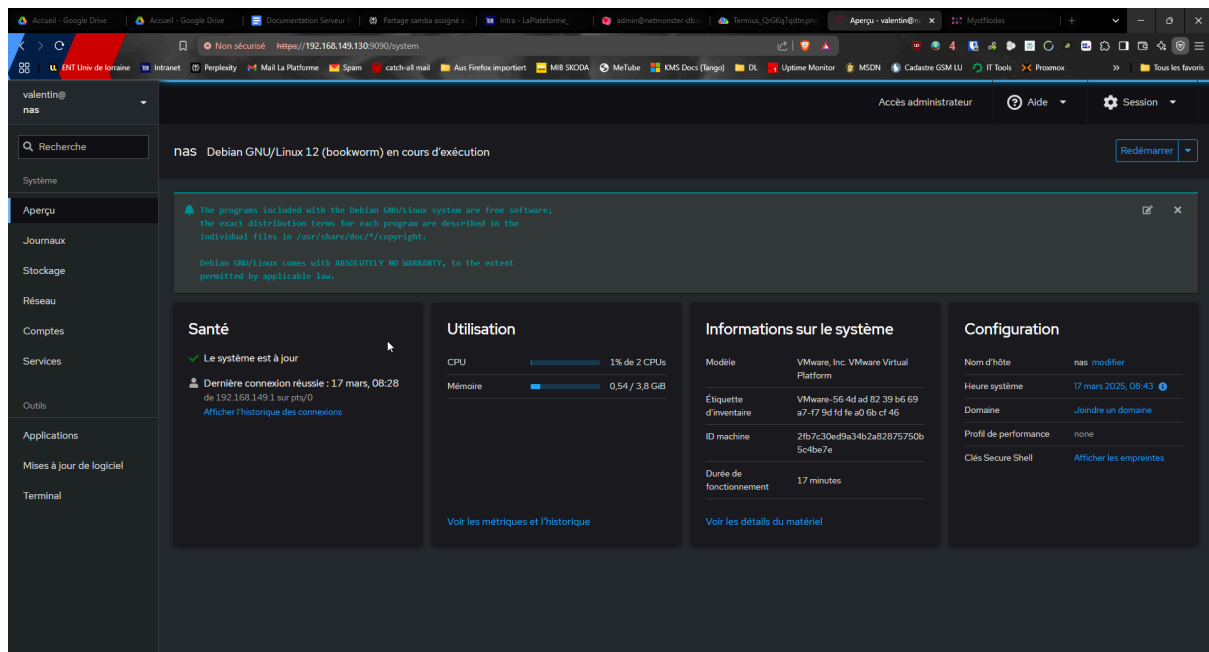
```
[laplateforme]  
path = /share/laplateforme  
browseable = no  
writable = yes  
read only = no  
valid user = laplateforme
```

```
[Public]  
path = /share/Public  
browseable = no  
writable = yes  
valid users = @smbshare  
force group = smbshare
```

Cockpit

Nous pouvons gérer depuis le web notre serveur grâce à cockpit.

Nous pouvons nous connecter à cette interface avec les utilisateurs “non-root” uniquement, pour éviter tout détournement. L'utilisateur root n'a pas accès au SSH non plus.



Création d'une interface WEB à l'aide de webdav;

Pour l'installer nous devons d'abord installer serveur http pour cela nous choisissons Apache2 qui est gratuit et polyvalent avec beaucoup de ressource en ligne.

```
sudo apt install apache2
```

On va ensuite l'activer et le lancer au démarrage de la VM.

```
sudo systemctl enable apache2  
sudo systemctl start apache2
```

On active les modules essentiels à Webdav inclus dans Apache2.

```
sudo a2enmod dav  
sudo a2enmod dav_fs
```

Bien sûr pour appliquer les changements, on redémarrera le service.

```
sudo systemctl restart apache2
```

Pour afficher les dossiers que l'on veut, on va donner l'accès à ce fichier au client web

```
sudo chown www-data:www-data /mnt/md0/share
```

Maintenant on configure le dossier de configuration du client

```
sudo nano /etc/apache2/sites-available/webdav.conf
```

On configure aussi le aussi un compte pour plus de sécurité

```
sudo htpasswd -c /etc/apache2/.htpasswd username
```

voici ce que l'on met à l'intérieur;

```
<VirtualHost *80>  
    ServerName 192.168.213.131  
    ServerAdmin webmaster@192.168.213.131  
    DocumentRoot /mnt/md0/share  
    Alias /share /mnt/md0/share
```

```
<Directory "/mnt/md0/share">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
    Dav On
    AuthType Basic
    AuthName "WebDav Restricted"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>

DavLockDB /var/www/DavLock
</VirtualHost>
```

Options Indexes FollowSymLinks :

Si aucun fichier index.htm n'est présent dans ce dossier, Apache affichera une liste des fichiers et dossiers qu'il contient.

AllowOverride None :

Cela signifie qu'aucun fichier .htaccess (un fichier de configuration local) dans ce dossier ne sera pris en compte. Les paramètres de configuration sont uniquement ceux définis ici.

Require all granted :

Avant la configuration de l'authentification webdav, cette ligne permet de donner l'accès à tout le monde.

Dav On :

Active WebDAV (Web-based Distributed Authoring and Versioning) pour ce dossier.

WebDAV permet aux utilisateurs de modifier et de gérer des fichiers directement sur le serveur via le web.

AuthType Basic :

Définit le type d'authentification sur "Basic". Cela signifie que les utilisateurs devront saisir un nom d'utilisateur et un mot de passe.

AuthName "WebDav Restricted" :

Définit le message affiché dans la boîte de dialogue d'authentification. Ici, ce sera "WebDav Restricted".

AuthUserFile /etc/apache2/.htpasswd :

Indique à Apache où trouver le fichier contenant les noms d'utilisateur et les mots de passe. /etc/apache2/.htpasswd est l'emplacement de ce fichier.

Require valid-user :


Seuls les utilisateurs dont les informations d'identification sont présentes dans le fichier .htpasswd seront autorisés à accéder à ce dossier.

DavLockDB /var/www/DavLock :

Indique à Apache où stocker les informations de verrouillage WebDAV. Cela évite que plusieurs utilisateurs ne modifient le même fichier en même temps. /var/www/DavLock est le fichier où ces informations sont stockées.

Index of /

Name	Last modified	Size	Description
----------------------	-------------------------------	----------------------	-----------------------------

 files/	2025-03-17 10:34	-	
--	------------------	---	--

Apache/2.4.62 (Debian) Server at 192.168.213.131 Port 80

On autorise la page et on relance

```
sudo a2ensite webdav.conf  
sudo systemctl reload apache2
```

On peut maintenant grâce à l'adresse IP de la VM se connecter à la page.

Installation d'un RAID5

Pour cela on utilise le tutoriel de [DigitalOcean](#) ;

Pour l'installation d'un RAID de niveau 5 on a besoin de minimum 3 disques durs supplémentaires en plus du disque principal (celui de l'OS)

On va d'abord vérifier si les disques dure sont bien détecter;

`lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT`

On devrait voir les 3 disques apparaitre

On crée le tableau avec la commande

`sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sda /dev/sdb /dev/sdc`

Bien sure on change sda, sdb, sdc avec les noms de nos propres disques.

cela va prendre un moment à se configurer, pour suivre l'avancer on utilise la commande

`cat /proc/mdstat`

maintenant que le RAID 5 est créé on va maintenant monter les fichiers système indispensable pour utiliser le RAID.

`sudo mkfs.ext4 -F /dev/md0`

`sudo mkdir -p /mnt/md0`

`sudo mount /dev/md0 /mnt/md0`

Etape par étape, on va créer les fichiers système, créer un point d'attache pour monter , et monter le RAID.

`df -h -x devtmpfs -x tmpfs`

On vérifie que le nouveau RAID est bien créer.

`lsblk -o NAME,SIZE,FSTYPE,TYPE,MOUNTPOINT`

initialisation a chaque étape dans /etc/fstab

```
UUID=07b4973d-d8a0-4b88-b6e6-4b94c7a6b1cd / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=d7fc817c-2743-4804-b39f-ae9a72ebde65 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
UUID=0baa582a-4dc4-4276-994e-1ed1f5cb43ee /mnt/md0 ext4 defaults 0 0
```