

NAS

Avant-projet NAS Debian :

Projet

- Serveur NAS et OS Debian sans interface graphique
- Fonctionnalité du NAS :
 1. Transfert de fichier via SFTP
 2. Accès sécurisé via WebDAV
 3. Espace partagé avec un dossier public
 4. Espace privé avec un dossier privé personnel par session
 5. Plusieurs comptes peuvent se connecter en simultanée
 6. Pour la confidentialité et la sécurité des données : Sessions attribuées distinctement
 7. Session administrateur :
 - . Privilège étendus (Supervision du système)
 - . Gestion des sessions utilisateur
 - . Modification des autorisations d'accès dossiers
- Inclure une sauvegarde (rsync et création d'un deuxième serveur)
- Utiliser du RAID

Phases de test pendant le déploiement :

- . Test de connectivité
- . Tester la stabilité des transferts de fichiers
 - . Tester la gestion des sessions
 - . Tester la modification des autorisations

C'est quoi un NAS :

Le NAS (*Network Attached Storage*), est un appareil de stockage autonome qui peut se connecter à votre réseau privé ou professionnel via internet. Il permet de stocker, partager, sécuriser mais aussi de faciliter l'accès à nos fichiers depuis plusieurs appareils.

Il fonctionne comme un *disque dur externe*, mais offrant davantage de sécurité, le serveur NAS se compose d'un boîtier comprenant différents emplacements appelés des baies, ainsi qu'un ou plusieurs disques durs. Le nombre de baies dépend de vos besoins d'espace de stockage mais aussi de la configuration souhaitée en termes de sécurisation des données :

Qui sera déterminés par une technologie qu'on appelle **RAID**, sont disponibles. Un câble d'alimentation, un ventilateur, un processeur, de la RAM et une carte mère viennent compléter le tout.

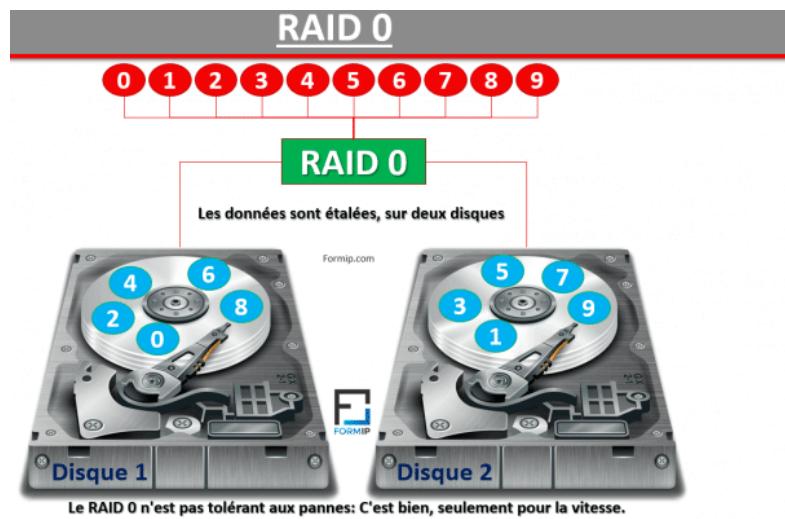
C'est quoi un RAID :

Un RAID (*Redundant Array of Independent Disks en français Regroupement Redondant de disques indépendants*) est une technologie de stockage qui combine plusieurs disques durs ou SSD en un seul ensemble pour améliorer les performances, la fiabilité et/ou la capacité de stockage.

Il existe plusieurs niveaux de RAID, chacun ayant ses avantages et ses inconvénients :

RAID 0 :

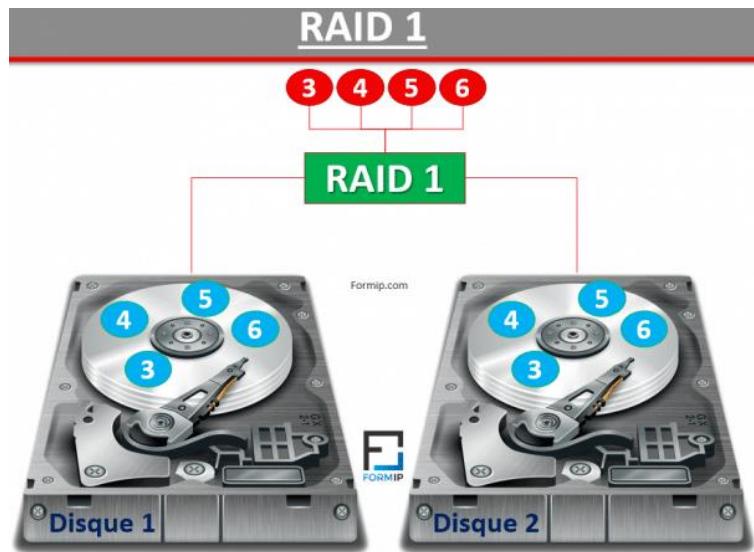
Le RAID 0 est une méthode de stockage qui utilise plusieurs disques pour améliorer la vitesse de lecture et d'écriture des données. Il divise les données entre ces disques, ce qui accélère leur traitement. Cependant, si l'un des disques tombe en panne, toutes les données stockées sur ce RAID sont perdues. En résumé, bien que le RAID 0 soit rapide, il n'offre aucune protection contre la perte de données en cas de panne de disque



RAID 1 :

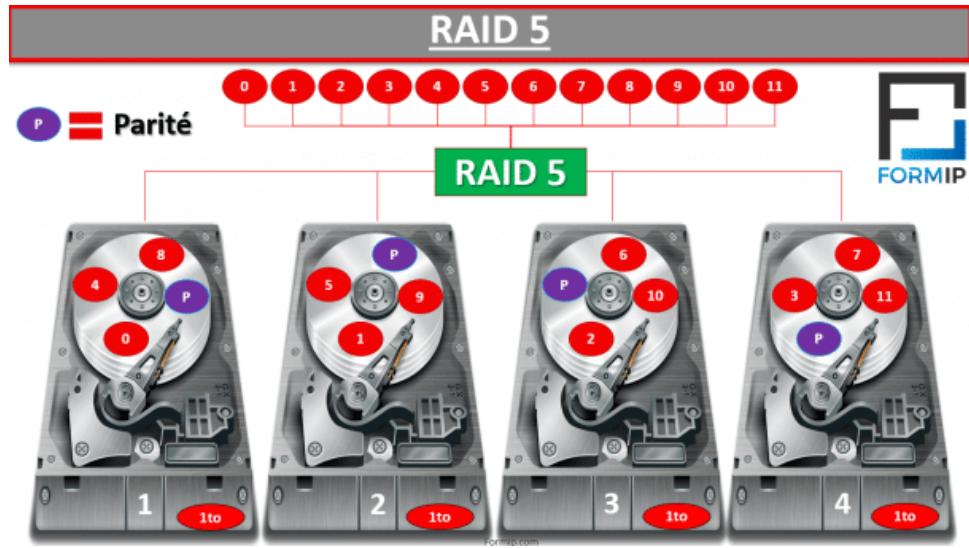
Le RAID 1 est une méthode de stockage qui utilise deux disques ou plus pour copier exactement les mêmes données sur chaque disque. Cela signifie que si l'un des disques tombe en panne, les données peuvent toujours être récupérées à partir du ou des autres disques. Il offre une redondance des données, ce qui signifie que même en cas de panne d'un disque, vos données sont en sécurité. Cependant, le RAID 1 n'offre pas d'amélioration

significative des performances car les données doivent être écrites sur chaque disque
Il s'agit d'une architecture de type « **Miroir** ».



RAID 5 : (le plus utilisé)

Le RAID 5 est une technologie de stockage qui utilise au moins 3 disques, mais souvent plus. Il est populaire car il offre à la fois vitesse et capacité de stockage élevée. Contrairement au RAID 1, il ne duplique pas les données sur tous les disques, mais les répartit avec une donnée spéciale appelée parité, qui est répartie uniformément sur chaque disque. Cette parité est utilisée pour reconstruire les données en cas de panne d'un disque. Cependant, une limitation du RAID 5 est que l'équivalent d'un disque entier est utilisé pour stocker cette parité, ce qui réduit la quantité totale de données disponibles. Par exemple, dans un système de 4 disques de 1 To chacun, le stockage total disponible serait de 3 To, car 1 To est réservé pour la parité.



RAID 10 :

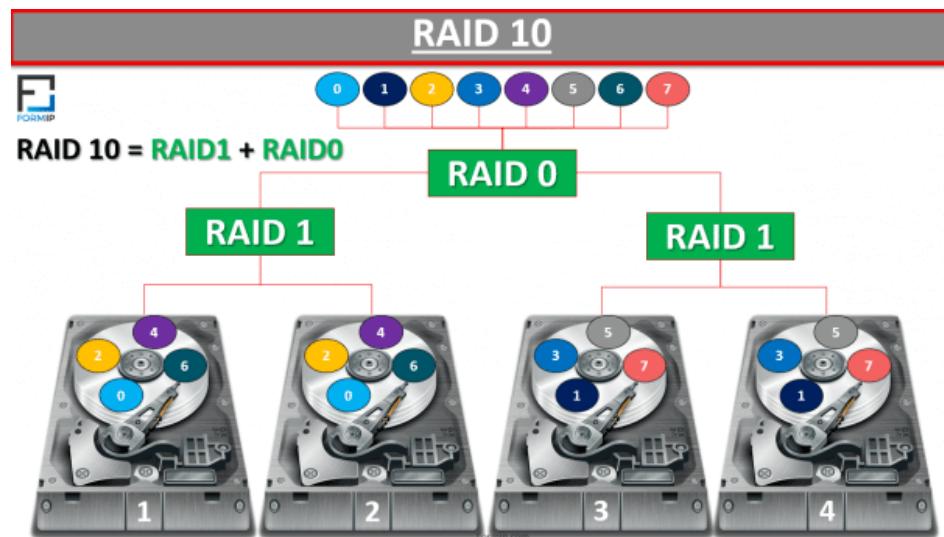
Le RAID 10 combine les avantages du RAID 1 et du RAID 0.

Il utilise au moins quatre disques, organisés en paires de disques miroirs (RAID 1), puis configurés en RAID 0 pour améliorer les performances.

Les données sont écrites simultanément sur deux disques miroirs pour la redondance et sur plusieurs paires pour améliorer les performances.

Cette configuration offre à la fois une excellente performance et une redondance élevée des données.

Cependant, elle nécessite plus de disques et utilise la moitié de l'espace de stockage total disponible pour la redondance des données.



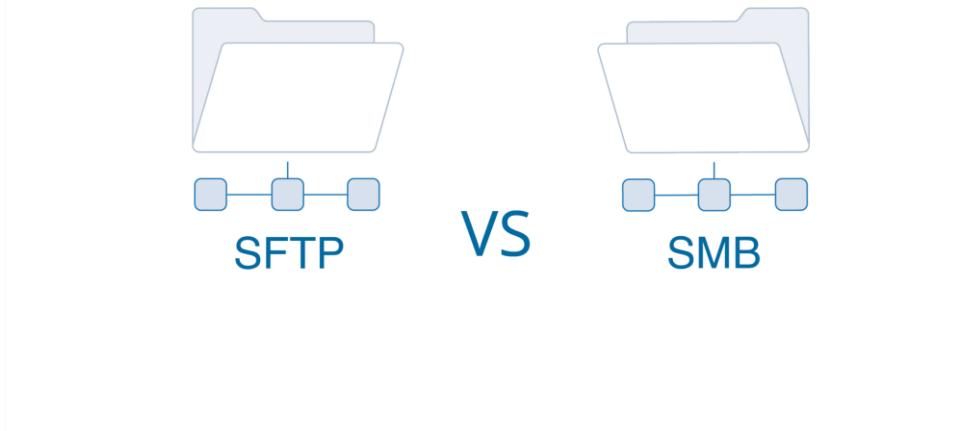
A quoi sert un NAS ? Quelle utilisation ?

Il offre plusieurs utilisations et avantages :

Stockage centralisé : Un NAS permet de centraliser et de stocker de grandes quantités de données en un seul endroit. Cela permet un accès facile et partagé aux fichiers pour tous les appareils connectés au réseau, tels que les ordinateurs, les smartphones, les tablettes, etc.

- **Sauvegarde automatique** : Les NAS offrent souvent des fonctionnalités de sauvegarde automatique, permettant de protéger les données importantes contre la perte en cas de défaillance d'un appareil.
- **Partage de fichiers** : Avec un NAS, vous pouvez facilement partager des fichiers et des dossiers avec d'autres utilisateurs ou appareils sur le même réseau. Cela facilite la collaboration sur des projets ou le partage de médias avec la famille ou les collègues.
- **Serveur multimédia** : Beaucoup de NAS peuvent également servir de serveur multimédia, permettant de diffuser des films, de la musique et des photos sur des appareils compatibles, tels que les téléviseurs intelligents, les consoles de jeux, etc.
- **Accès à distance** : Certains NAS offrent la possibilité d'accéder aux fichiers à distance via Internet, ce qui permet de récupérer des fichiers ou de les partager même lorsque vous n'êtes pas chez vous.

QUELLE EST LA DIFFERENCE ENTRE SFTP ET SMB ?



QU'EST-CE QUE SFTP

SFTP, ou Secure File Transfer Protocol, permet le transfert sécurisé de fichiers entre un hôte local et distant via un flux de données crypté SSH. Voici quelques points clés sur SFTP :

- **Connexion cryptée** : Toutes les données sont cryptées pendant le transit, assurant la sécurité des fichiers et des transferts via un tunnel SSH.
- **Port SSH standard** : Utilise le port SSH 22 pour les connexions, garantissant des communications sécurisées standardisées.
- **Accès au niveau des fichiers** : Permet un accès direct, la modification, la suppression, le renommage, le téléchargement et le téléversement de fichiers distants.
- **Autorisations granulaires** : Des contrôles d'accès utilisateur détaillés peuvent être configurés sur le serveur pour une sécurité renforcée.
- **Compatibilité multiplateforme** : Pris en charge sur Linux, Unix, Windows (avec clients) et macOS, grâce à son protocole basé sur SSH.
- **Utilisations courantes** : Hébergement web, administration système à distance, automatisation des transferts de fichiers entre serveurs.

En raison de son cryptage puissant et de ses contrôles au niveau des fichiers, SFTP est idéal pour accéder en toute sécurité aux fichiers sur un serveur distant, que ce soit pour le développement web, la gestion informatique ou tout autre scénario nécessitant une sécurité stricte des fichiers.

QU'EST CE QUE SMB

SMB (Server Message Block) est un protocole de partage de fichiers réseau qui permet aux utilisateurs d'accéder à des fichiers stockés sur un serveur distant comme s'ils étaient sur leur système local. Voici quelques points clés sur SMB :

1. **Accès au niveau partage** : Contrairement à SFTP, SMB partage des répertoires entiers, permettant de les monter en tant que lecteurs réseau.
2. **Basé sur TCP** : Fonctionne sur TCP pour les transferts de fichiers, sans cryptage intégré.
3. **Natif de Windows** : Développé par Microsoft, inclus dans toutes les versions de Windows, utilisant les ports TCP 139 et 445.
4. **Verrouillage de fichiers** : Gère efficacement l'accès aux fichiers entre plusieurs utilisateurs.

Conçu initialement pour une utilisation en réseau local, SMB facilite le partage transparent de fichiers entre les ordinateurs Windows, mais ne dispose pas du cryptage robuste trouvé dans SFTP.

Dans le domaine informatique, les développeurs web privilégient souvent l'utilisation de SFTP pour gérer les fichiers sur les serveurs web, tandis que les entreprises préfèrent SMB pour partager des fichiers et des imprimantes au sein de leur réseau Windows interne. Les équipes informatiques optent généralement pour SFTP pour automatiser l'administration sécurisée des serveurs Linux, tandis que les créateurs choisissent SMB pour collaborer sur des fichiers d'illustrations dans un environnement Windows.

- SFTP offre des transferts cryptés basés sur SSH et un accès au niveau des fichiers, idéal pour la gestion sécurisée des serveurs web et Linux.
- SMB est conçu pour un partage de répertoire transparent dans les environnements Windows, ce qui en fait le choix privilégié pour le partage de fichiers au sein d'un réseau Windows interne.
- Bien que SMB ait évolué avec les versions 3.0 et supérieures pour inclure le cryptage, SFTP reste la référence en matière de transferts de fichiers sécurisés et multiplateformes.

Le choix entre SFTP et SMB dépend des exigences spécifiques en matière de sécurité, de configuration réseau et de systèmes d'exploitation utilisés.

C'est quoi WebDAV

WebDAV, en entier : *Web-based Distributed Authoring and Versioning*, est un protocole déjà ancien (1996) et curieusement peu connu. Il permet pourtant une chose essentielle : **écrire** sur [le Web](#), au lieu de seulement surfer (c'est-à-dire seulement lire).

C'est un **protocole ouvert**, le W3C (organisme qui "normalise le web") en a confié le développement à l'IETF qui avait déjà normalisé HTTP.

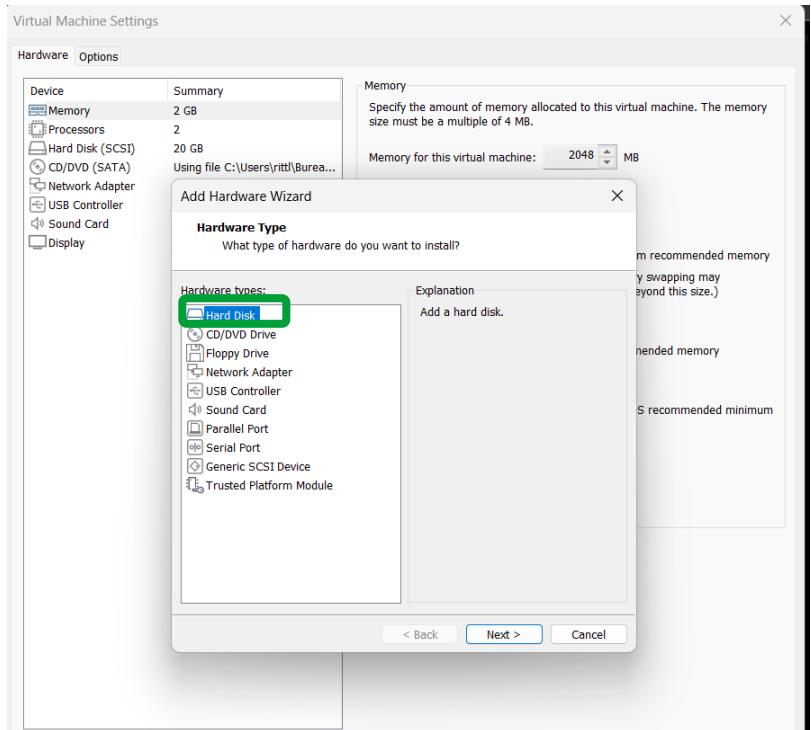
Pour résumer, WebDAV est une extension du HTTP. Au delà des GET et POST bien connus, WebDAV ajoute les méthodes PUT, DELETE, COPY, PROPFIND, etc. Pour les curieux, la norme est là : <http://tools.ietf.org/html/rfc2518>.

Étant une simple extension au protocole HTTP, WebDAV fonctionne dans à peu près toutes les situations où la navigation n'est pas bloquée.

Créer un Nas sur vm :

On commence par la configuration de notre vm

on ajoute 3 disque de 2gb qui nous serviront le RAID 5.



The image contains two vertically stacked windows from the 'Add Hardware Wizard'.
The top window is titled 'Select a Disk Type' and asks 'What kind of disk do you want to create?'. It shows four options: IDE, SCSI (which is selected and highlighted with a blue border), SATA, and NVMe. Below these are two checkboxes: 'IDE disks can be added only while the VM is powered off.' and 'NVMe disks can be added only while the VM is powered off.'
The bottom window is titled 'Specify Disk Capacity' and asks 'How large do you want this disk to be?'. It has a 'Maximum disk size (GB)' field set to '2.0' with a spin button. Below it, it says 'Recommended size for Debian 12.x 64-bit: 20 GB'. There are two radio button options: 'Allocate all disk space now.' (unchecked) and 'Split virtual disk into multiple files' (which is selected and highlighted with a blue border). A note below explains that splitting the disk makes it easier to move the virtual machine but may reduce performance with very large disks.
Both windows have standard navigation buttons at the bottom: '< Back', 'Next >', and 'Cancel'.

On passe notre adresse ip en static

```
GNU nano 7.2                               /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 ens33
iface ens33 inet static
    address 192.168.27.136/24
    gateway 192.168.27.2
    dns-nameservers 192.168.27.2
```

Configuration de la table de partionnement :

On commence par installer l'utilitaire **gdisk**

```
[root@Debian: ~]# apt update && apt install mdadm gdisk
```

- Nettoyer la **table de partitionnement** du disque **/dev/sdb** si nécessaires

⚠ Bien vérifier la lettre d'attribution des disques avec **fdisk -l pour éviter toute suppression malencontreuse. ⚠**

```
wipefs -a /dev/sdb[1-9]*
```

```
wipefs -a /dev/sdb
```

Maintenant on crée une nouvelle table de partitionnement pour **/dev/sdb**



```

root@debianNas:~# gdisk /dev/sdb
GPT fdisk (gdisk) version 1.0.9

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help):

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-4194270, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-4194270, default = 4192255) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): FD00
Changed type of partition to 'Linux RAID'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sdb.
The operation has completed successfully.
root@debianNas:~#

```

Definition

GPT fdisk" (gdisk) est un outil informatique en ligne de commande utilisé pour gérer les partitions de disque selon le style de partitionnement GPT (GUID Partition Table). Il permet de créer, modifier, supprimer et analyser des partitions sur un disque dur, offrant des fonctionnalités avancées pour travailler avec ce style de partitionnement moderne.

Commande pour afficher les disques

```

root@debianNas:~# lblk
bash: lblk: commande introuvable
root@debianNas:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda     8:0    0  20G  0 disk
└─sda1  8:1    0  19G  0 part /
  └─sda2  8:2    0    1K  0 part
  └─sda5  8:5    0  975M 0 part [SWAP]
sdb     8:16   0   2G  0 disk
└─sdb1  8:17   0   2G  0 part
sdc     8:32   0   2G  0 disk
sdd     8:48   0   2G  0 disk
sr0    11:0    1  627M 0 rom

```

```

Disque /dev/sdd : 2 GiB, 2147483648 octets, 4194304 secteurs
Modèle de disque : VMware Virtual S
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets

Disque /dev/sda : 20 GiB, 21474836480 octets, 41943040 secteurs
Modèle de disque : VMware Virtual S
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x329b347b

Périphérique Amorçage     Début      Fin Secteurs Taille Id Type
/dev/sda1      *          2048 39942143 39940096   19G 83 Linux
/dev/sda2            39944190 41940991 1996802   975M 5 Étendue
/dev/sda5            39944192 41940991 1996800   975M 82 partition d'échange Linux / Solaris

Disque /dev/sdb : 2 GiB, 2147483648 octets, 4194304 secteurs
Modèle de disque : VMware Virtual S
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : gpt
Identifiant de disque : 9C6898EF-CE5E-4520-AE7D-62938E3C7231

Périphérique Début      Fin Secteurs Taille Type
/dev/sdh1    2048 4192255 4190208    2G RAID Linux
root@debianMas:~# fdisk -l

```

COPIER LA TABLE DE PARTITIONNEMENT :

Copier la table de partitionnement d'un disque vers d'autres disques est essentiel pour assurer la cohérence et la fiabilité du système de stockage. Cela garantit une uniformité dans la configuration des disques, notamment lors de la configuration de systèmes RAID, de la sauvegarde et de la restauration de données, du clonage de disques, de la migration de données, et permet de réduire les risques d'erreurs humaines. En résumé, cette opération simplifie la gestion du stockage et contribue à maintenir l'intégrité des données

```

root@debianMas:~# sfdisk -d /dev/sdb | sfdisk --force /dev/sdc
Vérification que personne n'utilise le disque en ce moment... OK

Disque /dev/sdc : 2 GiB, 2147483648 octets, 4194304 secteurs
Modèle de disque : VMware Virtual S
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets

>>> Script d'en-tête accepté.
>>> Une nouvelle étiquette de disque GPT a été créée (GUID : 9C6898EF-CE5E-4520-AE7D-62938E3C7231)
/dev/sdc1: Une nouvelle partition 1 de type « Linux RAID » et de taille 2 GiB a été créée.
/dev/sdc2: Terminé.

Nouvelle situation :
Type d'étiquette de disque : gpt
Identifiant de disque : 9C6898EF-CE5E-4520-AE7D-62938E3C7231

Périphérique Début      Fin Secteurs Taille Type
/dev/sdc1    2048 4192255 4190208    2G RAID Linux

La table de partitions a été altérée.
Appel d'ioctl() pour relire la table de partitions.
Synchronisation des disques.

```

```

root@debianNas:~# sfdisk -d /dev/sdb | sfdisk --force /dev/sdd
Vérification que personne n'utilise le disque en ce moment... OK

Disque /dev/sdd : 2 GiB, 2147483648 octets, 4194304 secteurs
Modèle de disque : VMware Virtual S
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets

>>> Script d'en-tête accepté.
>>> Une nouvelle étiquette de disque GPT a été créée (GUID : 9C6898EF-CE5E-4520-AE7D-62938E3C7231)
/dev/sdd1: Une nouvelle partition 1 de type « Linux RAID » et de taille 2 GiB a été créée.
/dev/sdd2: Terminé.

Nouvelle situation :
Type d'étiquette de disque : gpt
Identifiant de disque : 9C6898EF-CE5E-4520-AE7D-62938E3C7231

Périphérique Début      Fin Secteurs Taille Type
/dev/sdd1      2048 4192255 4190208     2G RAID Linux

La table de partitions a été altérée.
Appel d'iocctl() pour relire la table de partitions.
Synchronisation des disques.

```

On vérifie que la table de partitionnement soit bien copier dans chaque disque que l'on va utiliser pour le raid

```

root@debianNas:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda    8:0    0  20G  0 disk
└─sda1  8:1    0  19G  0 part /
└─sda2  8:2    0   1K  0 part
└─sda5  8:5    0  975M 0 part [SWAP]
sdb    8:16   0   2G  0 disk
└─sdb1  8:17   0   2G  0 part
sdc    8:32   0   2G  0 disk
└─sdc1  8:33   0   2G  0 part
sdd    8:48   0   2G  0 disk
└─sdd1  8:49   0   2G  0 part
sr0    11:0   1  627M 0 rom

```

Installer mdadm

```

root@debianNas:~# apt install mdadm
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
mdadm est déjà la version la plus récente (4.2-5).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.

```

On crée notre raid5 avec cette commande :

```

root@debianNas:~# mdadm --create --level=5 --raid-devices=3 /dev/md0 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
root@debianNas:~# 

```

On vérifie que le **raid5** soit bien crée :

```
root@debianNas:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
      4186112 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

La commande **mkfs.ext4 /dev/md0** sert à créer un système de fichiers de type ext4 sur le périphérique **/dev/md0**. Cela formate le périphérique en effaçant toutes les données existantes et en créant une nouvelle structure de système de fichiers ext4, prête à être utilisée pour stocker des fichiers et des répertoires sur le système.

```
root@debianNas:~# mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 1046528 4k blocks and 261632 inodes
Filesystem UUID: a1411de4-0d7f-4535-a59c-6267664a2d7c
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Créer le dossier de montage :

La commande **mkdir /data** crée un nouveau répertoire nommé "data" à la racine du système de fichiers. Ce répertoire peut être utilisé pour monter des périphériques de stockage ou pour organiser des données selon les besoins du système. Une fois créé, vous pouvez utiliser ce répertoire pour stocker des fichiers et des répertoires ou pour monter d'autres périphériques de stockage.

```
root@debianNas:~# mkdir /data
root@debianNas:~# mount /dev/md0 /data
```

La commande `mount /dev/md0 /data` permet de monter le système de fichiers situé sur le périphérique `/dev/md0` sur le répertoire `/data`. Cela rend le contenu du système de fichiers accessible à partir du point de montage spécifié. Monter un système de fichiers permet d'accéder aux fichiers et répertoires qu'il contient à partir de l'emplacement de montage dans l'arborescence du système de fichiers global.

Affiché les points de montage :

```
root@debianNas:~# df -h
Sys. de fichiers Taille Utilisé Dispo Utile% Monté sur
udev          944M      0  944M   0% /dev
tmpfs         194M    792K  193M   1% /run
/dev/sda1      19G     1,9G   16G  11% /
tmpfs         967M      0  967M   0% /dev/shm
tmpfs         5,0M      0  5,0M   0% /run/lock
tmpfs         194M      0  194M   0% /run/user/1000
/dev/md0       3,9G    24K  3,7G  1% /data
```

Configurer mdadm pour assembler notre **RAID** au démarrage :

```
root@debianNas:~# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

La commande **mdadm --detail --scan >> /etc/mdadm/mdadm.conf** configure **mdadm** pour assembler automatiquement les ensembles RAID au démarrage. Elle scanne les ensembles RAID existants, génère leurs détails de configuration et les ajoute au fichier **/etc/mdadm/mdadm.conf**. Cela assure que les ensembles RAID seront assemblés correctement avec les bons paramètres dès le démarrage du système, sans nécessiter d'intervention manuelle.

Mettre à jour initramfs pour prise en compte :

```
root@debianNas:~# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-6.1.0-21-amd64
```

Ajouter le point de montage dans fstab pour un montage automatique au démarrage :

```
root@debianNas:~# echo "/dev/md0 /data ext4 rw,nofail,relatime,x-systemd.device-timeout=20s,defaults
0 2" >> /etc/fstab
```

La commande permet d'ajouter une ligne au fichier **/etc/fstab**, permettant un montage automatique du système de fichiers situé sur **/dev/md0** sur le répertoire **/data** au démarrage. Cette ligne de configuration spécifie le type de système de fichiers, les options de montage, et assure que le montage ne provoquera pas l'échec du démarrage en cas de problème

INSTALLATION ET CONFIGURATION DE SAMBA

Nous allons maintenant installer et configurer le service samba qui va permettre aux machines Windows d'accéder au partage de fichiers

```
root@debianNas:~# apt install samba
```

Créer un utilisateur sur samba qui fera office d'admin

```
root@debianNas:~# adduser --home /data --system manon
adduser : Attention ! Le répertoire personnel que vous avez indiqué (/data) existe déjà.
Ajout de l'utilisateur système « manon » (UID 103) ...
Ajout du nouvel utilisateur « manon » (UID 103) avec pour groupe d'appartenance « nogroup » ...
adduser : Le répertoire personnel « /data » existe déjà. Pas de modification de ce répertoire.
adduser : Attention ! Le répertoire personnel « /data » n'appartient pas à l'utilisateur que vous êtes en train de créer.
```

La commande permet de créer un nouvel utilisateur système nommé "manon" avec comme répertoire de base "/data". Les utilisateurs système sont souvent utilisés pour exécuter des services système ou des tâches système spécifiques, et dans ce cas, cet utilisateur pourrait être utilisé pour exécuter un service ou une tâche spécifique nécessitant un accès limité au système. Le répertoire "/data" pourrait être utilisé pour stocker les données associées à cet utilisateur.

On donne les droits :

```
root@debianNas:~# chown manon: /data
```

La commande permet de changer le propriétaire du répertoire `/data` pour l'utilisateur "manon", en lui attribuant également le groupe par défaut de cet utilisateur. Cela signifie que l'utilisateur "manon" devient le propriétaire du répertoire `/data` et peut donc accéder à ce répertoire et y effectuer des opérations telles que la lecture, l'écriture et la suppression de fichiers.

On créer un mdp de protection pour l'accès au partage :

```
root@debianNas:/data# smbpasswd -a manon
New SMB password:
Retype new SMB password:
Added user manon.
```

```

GNU nano 7.2                               /etc/samba/smb.conf
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
# - When such options are commented with ";", the proposed setting
#   differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the default
#   behaviour of Samba but the option is considered important
#   enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

===== Global Settings =====

[global]

## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = WORKGROUP
server string= nas

[share]
path = /data
read only = no
valid users= manon

```

Ce fragment de configuration Samba définit les paramètres globaux et spécifiques à une ressource partagée :

- **[global]**: *Paramètres globaux pour le service Samba.*
- **workgroup**: *Groupe de travail du serveur.*
- **server string**: *Nom du serveur Samba.*

- **[share]**: *Paramètres spécifiques à une ressource partagée.*
 - **path**: *Chemin d'accès du répertoire partagé.*
 - **read only**: *Accès en lecture/écriture activé.*
 - **valid users**: *Utilisateurs autorisés à accéder au partage.*

On redémarre les services :

```

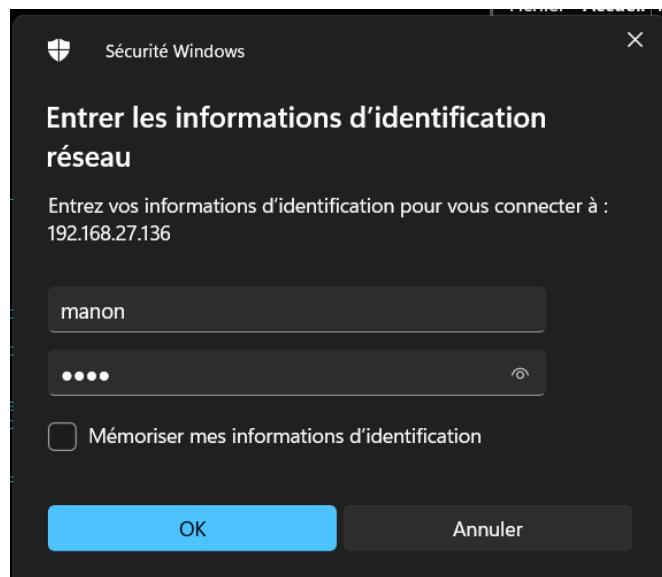
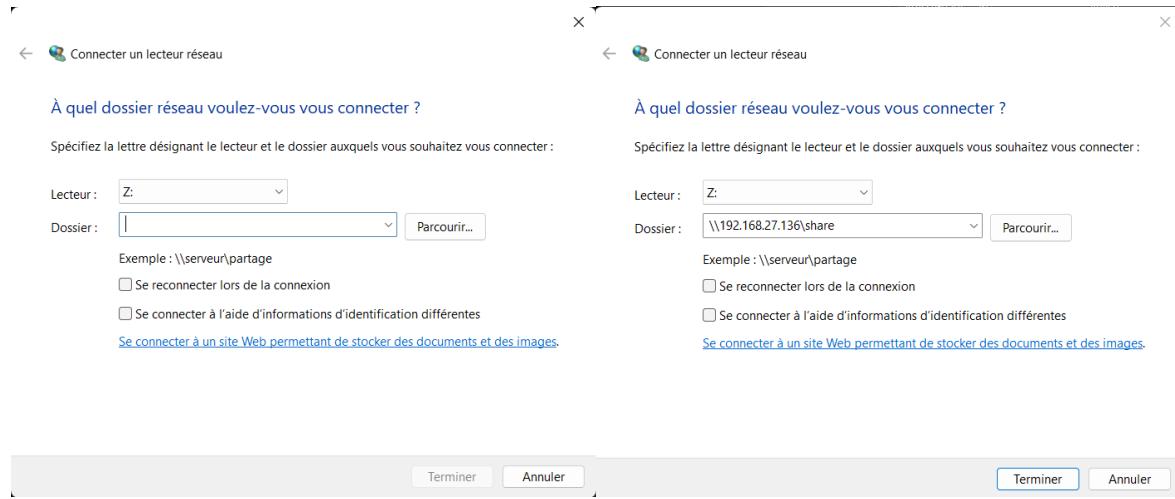
root@debianNas:/data# systemctl restart smbd; systemctl restart nmbd

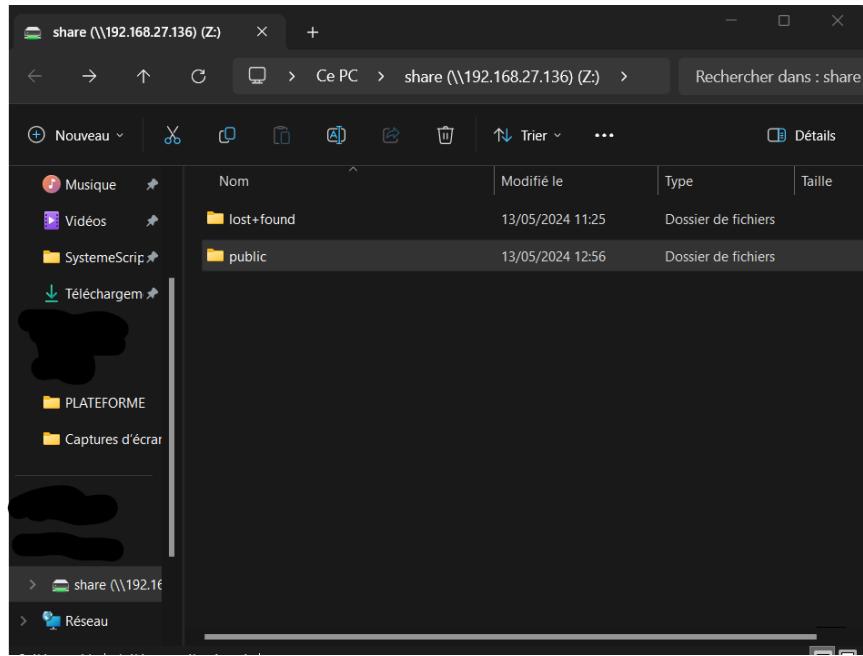
```

Maintenant on va créer notre dossier partage « PUBLIC »

```
root@debianNas:/data# mkdir public  
root@debianNas:/data# ls  
lost+found  public
```

On se connecte au NAS :





Maintenant on crée notre **groupe système** pour notre dossier « **public** »

```
root@debianNas:/data# groupadd -r public-users
```

La commande **sudo groupadd -r public-users** crée un groupe système nommé "public_users" sur votre système. Ce groupe peut être utilisé dans la configuration de Samba pour définir les utilisateurs autorisés à accéder au dossier partagé "public".

Les groupes système sont des groupes réservés aux tâches système sur les systèmes Unix/Linux. Ils sont utilisés pour gérer les autorisations d'accès aux fichiers système et pour exécuter des services système. Les groupes système ont des identifiants de groupe (GID) inférieurs à 1000 et sont distincts des groupes d'utilisateurs ordinaires.

On ajoute les utilisateurs que l'on souhaite autoriser au dossier « **public** »

```
root@debianNas:/data# usermod -aG public-users manon_
```

Commande pour voir les groupes:

```
public-users:x:995:manon,rija
root@debianNas:/data# getent group
```

Pour l'héritage des droits sur le dossier « **Public** » :

1. Je définie le groupe « **partage** » en tant que propriétaire :

```
root@debianNas:/data# chown :public-users /data/public
root@debianNas:/data# ls -l
total 20
drwx----- 2 root root      16384 13 mai   11:25 lost+found
drwxr-xr-x  2 root public-users  4096 13 mai   12:56 public
root@debianNas:/data# _
root@debianNAS:/etc# cd
root@debianNAS:~# sudo chown :partage /data/Public
```

2. J'active le bit setgid sur le dossier Public afin que tous les nouveaux fichiers et sous-dossiers créés dans Public héritent du groupe « partage » :

Le bit SetGID (Set Group ID) est un attribut de fichier spécial dans les systèmes Unix/Linux qui affecte le comportement de l'exécution des fichiers exécutables et des répertoires. Lorsqu'il est activé sur un fichier exécutable, il permet à celui-ci d'être exécuté avec les mêmes droits que le groupe propriétaire du fichier, au lieu des droits du groupe d'exécution de l'utilisateur qui lance le fichier. Sur un répertoire, le bit SetGID force tous les fichiers créés dans ce répertoire à hériter du groupe du répertoire, au lieu du groupe de l'utilisateur qui crée les fichiers. Cela garantit une cohérence des autorisations sur les fichiers créés dans ce répertoire.

```
root@debianNAS:/etc# cd
root@debianNAS:~# sudo chmod g+s /data/Public
```

3. Je modifie les permissions du dossier Public pour accorder des droits de lecture, d'écriture et d'exécution (rwx) au groupe partage :

```
root@debianNAs:/etc# cd
root@debianNAS:~# sudo chmod 2775 /data/Public
```

chmod u=rwx,g=rwx,o=rx, g+s nom_du_fichier

Cette commande définira les mêmes autorisations que **2775**, où :

- **u=rwx** : Le propriétaire du fichier aura les autorisations de lecture, écriture et exécution.
- **g=rwx** : Le groupe propriétaire du fichier aura les autorisations de lecture, écriture et exécution.
- **o=rx** : Les autres utilisateurs auront la permission de lecture et d'exécution.
- **g+s** : Le bit SetGID sera activé, ce qui signifie que les fichiers créés dans ce répertoire hériteront du groupe du répertoire.

4. Je configure le masque de création de fichiers pour le système. Cela assure que tous les

utilisateurs créeront des fichiers avec des permissions qui permettent au groupe partage de modifier les fichiers. J'ajoute « umask 002 » tout à la fin du fichier bash.bashrc situé dans le répertoire ./etc.

```
GNU nano 7.2                                bash.bashrc
#      PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
#      ;;
#*)
#      ;;
#esac

# enable bash completion in interactive shells
#if ! shopt -oq posix; then
#  if [ -f /usr/share/bash-completion/bash_completion ]; then
#    . /usr/share/bash-completion/bash_completion
#  elif [ -f /etc/bash_completion ]; then
#    . /etc/bash_completion
#  fi
#fi

# if the command-not-found package is installed, use it
if [ -x /usr/lib/command-not-found -o -x /usr/share/command-not-found/command-not-found ]; then
    function command_not_found_handle {
        # check because c-n-f could've been removed in the meantime
        if [ -x /usr/lib/command-not-found ]; then
            /usr/lib/command-not-found -- "$1"
            return $?
        elif [ -x /usr/share/command-not-found/command-not-found ]; then
            /usr/share/command-not-found/command-not-found -- "$1"
            return $?
        else
            printf "%s: command not found\n" "$1" >&2
            return 127
        fi
    }
fi
umask 002
```

Pour le test je change d'utilisateur, je vais créer un fichier text avec l'utilisateur jordan.

```
laplateforme@debianNAS:/data/Public$ su jordan
Mot de passe :
jordan@debianNAS:/data/Public$ _
```

Je créer un fichier texte et regarde les droits :

```
jordan@debianNAS:/data/Public$ ls
hello.txt
jordan@debianNAS:/data/Public$ echo "hey c'est jordan" > test.txt
```

Nous pouvons bien voir que le document créé par jordan a bien hérité des droits, contrairement au fichier « hello.txt » qui lui a été créé avant la règle d'héritage.

```
jordan@debianNAS:/data/Public$ ls -l
total 8
-rw-r--r-- 1 laplateforme partage 6 10 mai 15:21 hello.txt
-rw-rw-r-- 1 jordan      partage 17 10 mai 15:49 test.txt
```

Je change d'utilisateur pour tester si mes droits fonctionne bien en ajoutant « Bonjour Jordan » dans le fichier test.txt:

```
mot de passe :
laplateforme@debianNAS:/data/Public$ nano test.txt _
```

```
|laplateforme@debianNAS:/data/Public$ cat test.txt  
hey c'est jordan  
Bonjour Jordan
```

Les droits ont bien fonctionné

VERIFIER SI UN UTILISATEUR EST ACTIVE OU DESACTIVER

Le message que vous voyez indique que l'utilisateur "rija" est configuré pour se connecter avec **/usr/sbin/nologin**, ce qui signifie que la connexion de cet utilisateur est intentionnellement désactivée. Lorsqu'un utilisateur est configuré pour utiliser **nologin** comme shell de connexion, cela signifie qu'il ne peut pas se connecter au système via une interface interactive comme un terminal

```
|root@debianNas:~# cat /etc/passwd  
root:x:0:0:root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin  
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin  
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin  
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin  
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin  
laplateforme:x:1000:1000:laplateforme,,,:/home/laplateforme:/bin/bash  
Debian-exim:v:102:110::/var/spool/exim4:/usr/sbin/nologin  
manon:x:103:65534::/data:/usr/sbin/nologin  
rija:x:104:65534::/data:/bin/bash
```

```
|root@debianNas:~# chsh -s /bin/bash manon
```

Cette commande change le shell de connexion de l'utilisateur "rija" pour qu'il utilise Bash comme shell par défaut lorsqu'il se connecte au système

```
manon:x:103:65534::/data:/bin/bash  
rija:x:104:65534::/data:/bin/bash
```

Utiliser le WebDAV :

WebDAV est un protocole (extension http) qui permet de gérer les fichiers avec des serveurs distants. Il permet la synchronisation de dossiers, publication, etc..

WebDAV permet de rendre possible l'écriture de données à travers le web, et notamment d'éditer du contenu web simultanément.

Gestion en auto des droits en verrouillant momentanément les dossiers édités.

1. Il faut au préalable avoir notre adresse IP en statique, ici mon IP est 192.168.197.100

```
root@debianNAS:~# ip a | grep ens33
    inet 192.168.197.100/24 brd 192.168.197.255 scope global ens33
```

2. Installer WebDAV

- Si Apache n'est pas déjà installé, veuillez l'installer avec la commande :

```
root@debianNAS:~# apt install apache2
```

- Activer les modules WebDAV dav_fs et dav :

```
root@debianNAS:~# a2enmod dav_fs_
root@debianNAS:~# a2enmod dav_
root@debianNas:~# a2enmod dav_fs
Considering dependency dav for dav_fs:
Enabling module dav.
Enabling module dav_fs.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debianNas:~# a2enmod dav
Module dav already enabled
```

-Créer un repertoire (dossier) web dans lequel un fichier mot de passe « passwddav »

```
root@debianNas:/var/www# ls
html
root@debianNas:/var/www# mkdir -p /var/www/web1
```

- Redémarrer Apache :

```
root@debianNAS:~# /etc/init.d/apache2 reload_
```

```
root@debianNas:~# systemctl restart apache2
root@debianNas:~# /etc/init.d/apache2 reload
Reloading apache2 configuration (via systemctl): apache2.service.
```

3. Création de notre hôte virtuel. En d'autres termes, on va mettre à disposition notre dossier « /data/Public/ »

- Pour mettre par défaut notre répertoire /data/Public, il faut modifier le fichier de configuration « 000-default.conf » situé dans le répertoire /etc/apache2/sites-available.

Je vais d'abord créer un backup de celui-ci :

```
root@debianNAS:~# mv /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/000-default.conf.backup
```

Je peux maintenant créer mon nouveau fichier de configuration :

```
root@debianNAS:~# nano /etc/apache2/sites-available/000-default.conf
```

Je configure ce fichier comme suit :

```
GNU nano 7.2          /etc/apache2/sites-available/000-default.conf *
NameVirtualHost *
<VirtualHost *>
    ServerAdmin webmaster@localhost

    DocumentRoot /data/Public/
    <Directory /data/Public/>
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

</VirtualHost>
```

NameVirtualHost *: Cette directive indique à Apache d'écouter sur toutes les interfaces réseau disponibles. Cela signifie que ce VirtualHost répondra aux requêtes HTTP entrantes sur n'importe quelle adresse IP configurée sur le serveur.

<VirtualHost *> : Début de la configuration pour un hôte virtuel. L'astérisque (*) spécifie que cet hôte virtuel répondra à toutes les requêtes entrantes sur toutes les adresses IP configurées sur le serveur.

ServerAdmin webmaster@localhost : L'adresse e-mail du webmaster pour ce site.

DocumentRoot /data/Public/ : C'est le répertoire racine où les fichiers du site sont stockés sur le serveur. Dans ce cas, il s'agit de /data/Public/.

<Directory /data/Public/> : Définit les paramètres spécifiques au répertoire /data/Public/.

Options Indexes Multiviews : Cette directive indique les options activées pour ce répertoire. "Indexes" permet à Apache de générer une liste de fichiers dans un répertoire s'il n'y a pas d'index (comme un fichier index.html). "Multiviews" permet à Apache de servir plusieurs versions d'un fichier basées sur les préférences de l'utilisateur.

AllowOverride None : Cette directive spécifie que les fichiers de configuration .htaccess dans ce répertoire ne seront pas autorisés à modifier les configurations Apache. Cela peut être une mesure de sécurité pour éviter que des utilisateurs non autorisés ne modifient la configuration du serveur à partir du contenu du site.

Order allow,deny et allow from all : Ces directives permettent l'accès à ce répertoire. "Order allow,deny" spécifie l'ordre dans lequel les directives d'autorisation sont appliquées. "allow from all" autorise toutes les demandes d'accès à ce répertoire.

</Directory> : Fin de la configuration spécifique au répertoire.

</VirtualHost> : Fin de la configuration de l'hôte virtuel.

Je redémarre Apache :

```
root@debianNAS:~# /etc/init.d/apache2 reload
```

5. Configurer l'hôte virtuel pour WebDAV

Je vais maintenant créer un fichier mot de passe « passwd.dav » dans le répertoire /var/www/web1 pour l'utilisateur « laplateforme » :

```
root@debianNas:/var/www# htpasswd -c /var/www/web1/passwd.dav laplateforme
New password:
Re-type new password:
```

htpasswd : C'est le nom de la commande elle-même, qui est utilisée pour créer et gérer les fichiers de hachage des mots de passe.

-c : C'est une option qui indique à « htpasswd » de créer un nouveau fichier de mot de passe ou de le remplacer s'il existe déjà. Cette option est suivie du chemin vers le fichier de mot de passe que vous souhaitez créer ou modifier.

/var/www/web1/passwd.dav : C'est le chemin complet vers le fichier de mot de passe. Dans cet exemple, le fichier de mot de passe sera situé dans /var/www/web1/ et aura pour nom passwd.dav. Si le fichier existe déjà, cette commande le remplacera. Si le fichier n'existe pas, htpasswd le créera.

laplateforme : C'est le nom d'utilisateur pour lequel vous allez définir un mot de passe dans le fichier de mot de passe. Lorsque vous exécutez cette commande, elle vous demandera d'entrer un mot de passe pour cet utilisateur. Une fois que vous avez saisi le mot de passe, htpasswd le crypte et l'ajoute au fichier de mot de passe.

Une fois créé, voilà à quoi ressemble notre fichier passwd.dav :

```
root@debianNas:/var/www/web1# cat passwd.dav
laplateforme:$apr1$2fcmlz02$TqT/lQwan34Brq.cmMk0/.
```

```
GNU nano 7.2                               /var/www/web1/passwd.dav
laplateforme:$apr1$TuF.97fu$tkpixlQX7.SS0V1D1Tr7U0
```

Nous utiliserons plus tard l'URL **http://192.168.197.100/webdav** pour se connecter à WebDAV. Lorsque je le fais sur un client Windows et que je saisie le nom d'utilisateur 'laplateforme', Windows le traduit en '192.168.0.100\laplateforme'. Par conséquent, nous créons maintenant un deuxième compte utilisateur (sans l'option -c car le fichier de mot de passe existe déjà) :

```
root@debianNas:/var/www/web1# htpasswd /var/www/web1/passwd.dav 192.168.27.136\\laplateforme
New password:
Re-type new password:
Adding password for user 192.168.27.136\laplateforme
```

192.168.167.100\\laplateforme : C'est la syntaxe utilisée pour spécifier le nom d'utilisateur. Dans ce cas, 192.168.167.100 est utilisé comme préfixe du nom d'utilisateur, suivi de \\ pour échapper le caractère de séparation \\", puis le nom d'utilisateur réel, qui est laplateforme.

Je change les permissions sur le fichier **/var/www/web1/passwd.dav** pour que seul l'utilisateur root et les utilisateurs du groupe www-data peuvent y accéder

```
root@debianNAS:~# chown root:www-data /var/www/web1/passwd.dav
root@debianNAS:~# chmod 640 /var/www/web1/passwd.dav
```

640 : C'est l'argument qui spécifie les nouvelles autorisations pour le fichier **/var/www/web1/passwd.dav**. Les trois chiffres représentent les autorisations pour le propriétaire, le groupe et les autres utilisateurs respectivement.

- Le premier chiffre, **6**, représente les autorisations pour le propriétaire du fichier. Dans ce cas, **6** correspond à la somme des autorisations de lecture (**4**) et d'écriture (**2**), ce qui signifie que le propriétaire a le droit de lire et de modifier le fichier.
- Le deuxième chiffre, **4**, représente les autorisations pour le groupe auquel appartient le fichier. Dans ce cas, **4** correspond uniquement à l'autorisation de lecture, ce qui signifie que les membres du groupe auquel appartient le fichier peuvent le lire mais pas le modifier.
- Le troisième chiffre, **0**, représente les autorisations pour les autres utilisateurs (ceux qui ne sont ni le propriétaire du fichier ni membres du groupe). Dans ce cas, **0** signifie qu'ils n'ont aucune autorisation.

J'ajoute les lignes suivantes au fichier '`/etc/apache2/sites-available/000-default.conf`' :

```
GNU nano 7.2                               /etc/apache2/sites-available/000-default.conf
NameVirtualHost *
<VirtualHost *>
    ServerAdmin webmaster@localhost

    DocumentRoot /data/Public/
    <Directory /data/Public/>
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    Alias /webdav /data/Public/

    <Location /webdav>
        DAV On
        AuthType Basic
        AuthName "webdav"
        AuthUserFile /var/www/web1/passwd.dav
        Require valid-user
    </Location>
</VirtualHost>
```

Alias /webdav /var/www/web1/web :

- Cette directive crée un alias pour le chemin **/webdav**, qui pointe vers le répertoire **/var/www/web1/web**.
- Cela signifie que lorsque vous accédez à l'URL **http://votresite/webdav**, Apache servira les fichiers à partir du répertoire **/var/www/web1/web**.

<Location /webdav> :

- Définit la configuration pour l'emplacement /webdav, qui correspond à l'alias précédemment défini.
- Cela signifie que toutes les directives à l'intérieur de cette balise s'appliquent uniquement aux requêtes qui correspondent à l'URL avec le chemin /webdav.

DAV On :

- Active le module WebDAV pour ce chemin. Cela indique à Apache que le répertoire /var/www/web1/web peut être utilisé avec WebDAV pour effectuer des opérations de manipulation de fichiers à distance (comme la lecture, l'écriture, etc.).

AuthType Basic :

- Spécifie le type d'authentification utilisé, qui est l'authentification de base. Cela signifie que les utilisateurs doivent fournir un nom d'utilisateur et un mot de passe pour accéder à ce répertoire.

AuthName "webdav" :

- Définit le nom d'authentification qui sera affiché dans la boîte de dialogue d'authentification du navigateur lorsque les utilisateurs essaient d'accéder à ce répertoire.

AuthUserFile /var/www/web1/passwd.dav :

- Spécifie le chemin vers le fichier de hachage des mots de passe utilisé pour l'authentification. Dans cet exemple, le fichier /var/www/web1/passwd.dav est utilisé pour stocker les noms d'utilisateur et les mots de passe.

Require valid-user :

- Indique qu'un utilisateur valide doit fournir des informations d'identification pour accéder à ce répertoire. Cela signifie que tout utilisateur présent dans le fichier de hachage des mots de passe peut accéder au répertoire.

Je redémarre apache :

```
root@debianNAS:~# /etc/init.d/apache2 reload_
```

5. Je test si WebDAV fonctionne :

Test avec Cadaver :

```
root@debianNAS:~# apt install cadaver_
```

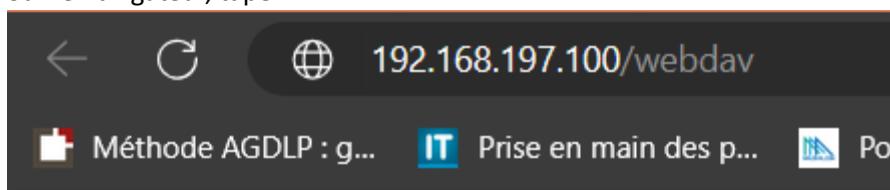
```
root@debianNAS:~# cadaver http://localhost/webdav/_
```

Si vous avez ça affiché, alors c'est fonctionnel :

```
Authentication required for webdav on server `localhost':  
Username: laplateforme  
Password:  
dav:/webdav/> _
```

Test sur Windows :

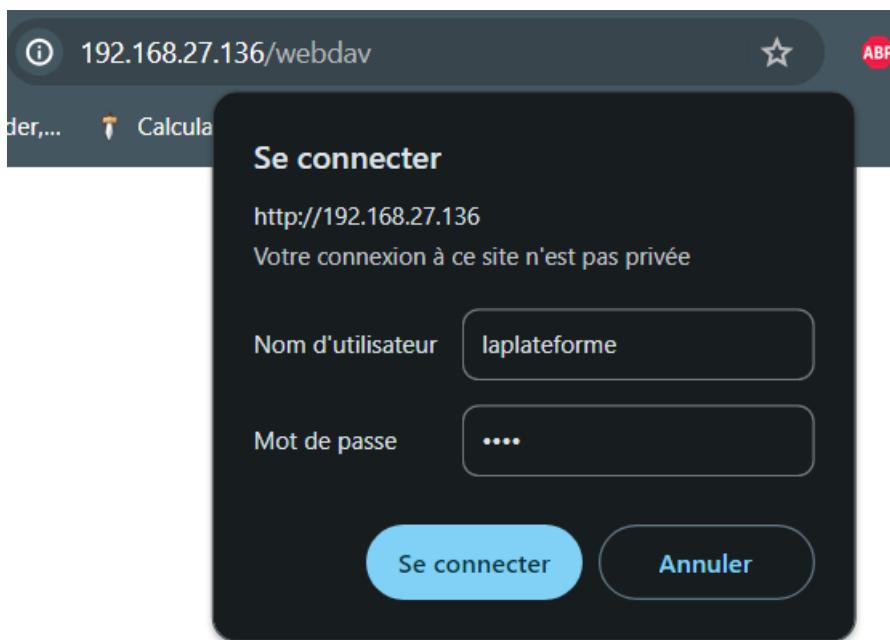
Sur le navigateur, taper :



Il va ensuite vous demander de taper l'utilisateur et le mot de passe :

Pour moi c'est ;

- utilisateur : laplateforme
- Mdp : LaPlateforme13



Une fois l'identifiant entré, je peux maintenant accéder à WebDAV :

Index of /webdav

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
-------------	----------------------	-------------	--------------------

 Parent Directory	-	-	-
 public/	2024-05-14 11:29	-	-

Apache/2.4.59 (Debian) Server at 192.168.27.136 Port 80

Se connecter au webDAV avec une connection HTTPS

Étape 1 : Activer le module SSL

D'abord faire une mise un check de mise à jour de OpenSSL :

```
root@debianNAS:~# apt update  
root@debianNAS:~# apt upgrade openssl
```

Activer le module SSL par défaut dans Apache :

```
root@debianNAS:~# a2enmod ssl
```

Ensuite, nous activons le site web ssl par défaut (après avoir taper la commande, si une erreur apparait, veuillez ne pas le prendre en compte):

```
root@debianNAS:~# a2enmod default-ssl
```

Redémarrez le service Apache pour que la modification soit prise en compte :

```
root@debianNAS:~# service apache2 reload
```

Étape 2 : Créer un certificat SSL auto-signé

Créer un nouveau dossier dans le répertoire Apache où nous pouvons stocker le certificat et la clé privée :

```
root@debianNAS:~# mkdir /etc/apache2/ssl
```

Nous allons générer un nouveau certificat et une clé privée à l'aide d'Openssl en exécutant la commande ci-dessous :

```
root@debianNAS:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/cyrus.key -out /etc/apache2/ssl/cyrus.crt
```

openssl :

L'outil de ligne de commande OpenSSL, utilisé pour divers types de cryptographie et de gestion des certificats.

req :

Cette sous-commande de openssl est utilisée pour la gestion des demandes de signature de certificat (Certificate Signing Requests - CSR).

-x509 :

Cette option spécifie que nous voulons créer un certificat auto-signé au lieu d'une CSR. X.509 est le standard pour les certificats de clé publique.

-nodes :

Cette option indique que la clé privée ne doit pas être chiffrée avec une phrase de passe (no DES encryption).

-days 365 :

Spécifie la durée de validité du certificat en jours. Dans ce cas, le certificat sera valable pendant 365 jours (1 an).

-newkey rsa:2048 :

Cette option génère une nouvelle clé privée et une demande de certificat en utilisant l'algorithme RSA avec une taille de clé de 2048 bits.

-keyout /etc/apache2/ssl/cyrus.key :

Spécifie le chemin et le nom de fichier où la clé privée générée sera sauvegardée. Dans ce cas, la clé privée sera sauvegardée dans /etc/apache2/ssl/cyrus.key.

-out /etc/apache2/ssl/cyrus.crt :

Spécifie le chemin et le nom de fichier où le certificat auto-signé sera sauvegardé. Dans ce cas, le certificat sera sauvegardé dans /etc/apache2/ssl/cyrus.crt.

Entrer les informations demandé ci-dessous, adapter l'IP à selon la vôtre :

La clé privée et le certificat nouvellement générés doivent être protégés. Par conséquent, nous allons attribuer une autorisation de fichier de sorte que seul le propriétaire puisse lire et écrire le fichier et que les autres utilisateurs n'y aient pas accès.

Nous allons définir l'autorisation de fichier à 600 :

```
root@debianNAS:/# chmod 600 /etc/apache2/ssl/*_
```

Nous allons configurer l'hôte virtuel Apache par défaut pour qu'il utilise le certificat et la clé privée :

```
rroot@dehjanNAS:~# nano /etc/apache2/sites-available/
```

For more information, visit www.fcc.gov.

Ajouter ces deux lignes là à la fin du fichier :

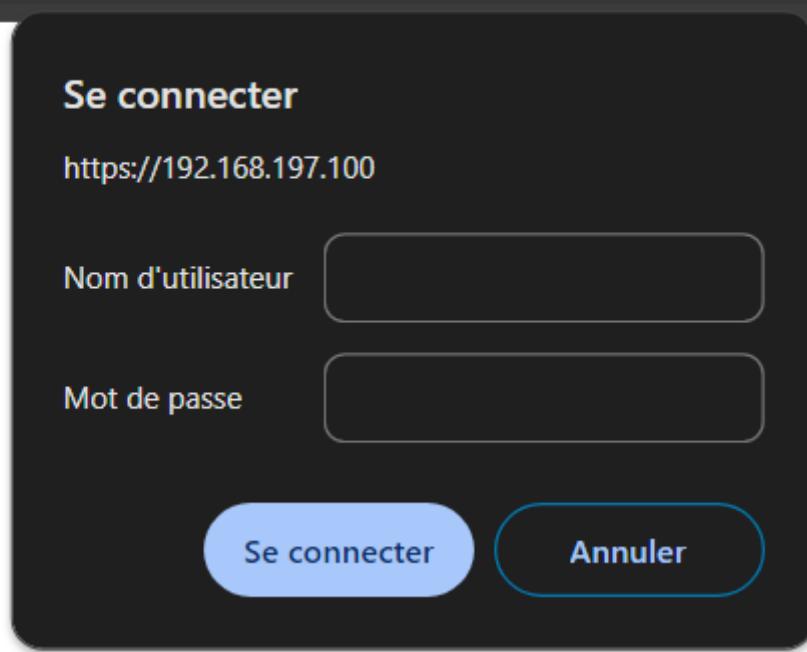
```
SSLCertificateFile /etc/apache2/ssl/cyrus.crt  
SSLCertificateKeyFile /etc/apache2/ssl/cyrus.key
```

Redémarrer apache2 :

```
root@debianNAS:~# systemctl restart apache2
```

Nous pouvons maintenant nous connecter en https :

(i) 192.168.197.100/webdav



UTILISATION SFTP

Lorsque vous installez Debian et que vous sélectionnez l'option "serveur SSH" lors de l'installation, cela va généralement installer le paquet OpenSSH, qui est l'implémentation open-source du protocole SSH (Secure Shell) sur Debian.

OpenSSH est une suite d'outils comprenant notamment le démon sshd pour le serveur SSH et les outils en ligne de commande tels que ssh pour se connecter à des serveurs distants de manière sécurisée, scp pour transférer des fichiers, et sftp pour le transfert de fichiers via le protocole SFTP.

En sélectionnant l'option "serveur SSH" lors de l'installation de Debian, vous installez le serveur SSH et les outils associés, ce qui vous permet de vous connecter à distance à votre serveur Debian via SSH dès que l'installation est terminée. Cela vous donne un accès sécurisé à votre système via une interface en ligne de commande, ce qui est essentiel pour l'administration à distance et la gestion des serveurs.

Commande pour se connecter, on ouvre l'invite de commande sur notre pc

```
C:\Users\ritt1l>sftp manon@192.168.27.136
manon@192.168.27.136's password:
Connected to 192.168.27.136.
```

Commande **GET** pour afficher le chemin d'un fichier et télécharger un fichier

```
sftp> get trucmuch.txt
Fetching /data/public/trucmuch.txt to trucmuch.txt
100% 39 8.4KB/s 00:00
sftp> get trucmuch.txt C:/Users/ritt1\Downloads
Fetching /data/public/trucmuch.txt to C:/Users/ritt1/Downloads/trucmuch.txt
100% 39 15.7KB/s 00:00
```

Commande **PUT** uploader un fichier vers le serveur

```
sftp> put C:/Users/ritt1/Downloads/NASV2.docx /data/public/
uploading C:/Users/ritt1/Downloads/NASV2.docx to /data/public/NASV2.docx
C:/Users/ritt1/Downloads/NASV2.docx
100% 464KB 17.7MB/s 00:00
sftp> ls
NASV2.docx dd.txt ffdd.txt tesrt.txt testheritage.txt trucmuch.txt
```

1. Voici quelques-unes des commandes les plus couramment utilisées :

- **ls** : Listez les fichiers et répertoires dans le répertoire distant actuel.
- **cd** : Changez de répertoire sur le serveur distant.
- **get** : Téléchargez un fichier du serveur distant vers votre système local.
- **put** : Téléversez un fichier de votre système local vers le serveur distant.
- **pwd** : Affichez le répertoire de travail distant actuel.
- **exit** : Déconnectez-vous du serveur SFTP et quittez la session SFTP.

Créer Espace privé avec un dossier privé personnel par session

On commence par se connecter avec **ssh** avec l'utilisateur pour lequel on veut faire l'espace privée.

Voici la commande :

```
root@debianNas:~# ssh rija@192.168.27.136
ssh: Could not resolve hostname rija@192.168.27.136: Name or service not known
root@debianNas:~# ssh rija@192.168.27.136
rij@192.168.27.136's password:
Linux debianNas 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) 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
```

On vérifie que l'on soit dans le **NAS** (notre répertoire « **data** »)

```
permitted by applicable law.  
rija@debianNas:~$ ls  
lost+found manon public
```

Maintenant on crée notre dossier privé pour notre utilisateur :

```
rija@debianNas:~$ sudo mkdir rija  
[sudo] Mot de passe de rija :
```

On configure les permissions ;

D'abord on met l'utilisateur choisi en propriétaire du répertoire que l'on veut priver

```
rija@debianNas:~$ sudo chown rija: rija
```

On accorde à l'utilisateur (nous rija) des droits complets sur son dossier personnel (lecture, écriture, exécution) tout en refusant l'accès à tout autre utilisateur. Cela garantit la confidentialité et la sécurité des données dans le dossier utilisateur1.

```
rija@debianNas:~$ sudo chmod 700 rija
```

Voici ce que signifient les chiffres "700" dans la commande chmod :

- Le premier chiffre (7) spécifie les permissions pour l'utilisateur propriétaire du fichier/dossier.
- Le deuxième chiffre (0) spécifie les permissions pour le groupe propriétaire du fichier/dossier.
- Le troisième chiffre (0) spécifie les permissions pour les autres utilisateurs (ceux qui ne sont ni le propriétaire ni dans le groupe propriétaire).

Dans ce cas :

- Le chiffre "7" pour l'utilisateur propriétaire accorde les permissions de lecture, écriture et exécution.
- Les chiffres "0" pour le groupe propriétaire et les autres utilisateurs signifient qu'ils n'ont aucun droit d'accès.

Configuration de Rsync

Qu'est ce que c'est ?

Rsync, abréviation de "remote sync", est un **utilitaire de logiciel libre** pour Unix et Linux, permettant la **synchronisation de fichiers** et de répertoires entre deux emplacements de manière efficace et rapide. Il est souvent utilisé pour les **sauvegardes**, les copies de fichiers et la gestion de fichiers sur des réseaux.

Installation rsync :

```
root@debian:/home/master# apt install rsync
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  python3-braceexpand
Les NOUVEAUX paquets suivants seront installés :
  rsync
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 417 ko dans les archives.
Après cette opération, 795 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 rsync amd64 3.2.7-1 [417 kB]
417 ko réceptionnés en 0s (861 ko/s)
Sélection du paquet rsync précédemment désélectionné.
(Lecture de la base de données... 45080 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../rsync_3.2.7-1_amd64.deb ...
Dépaquetage de rsync (3.2.7-1) ...
Paramétrage de rsync (3.2.7-1) ...
rsync.service is a disabled or a static unit, not starting it.
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
root@debian:/home/master# _
```

Autoriser rsync à se lancer : nano /etc/default/rsync et on passe le enable en true :

```
RSYNC_ENABLE=true
```

On crée l'utilisateur et le groupe rsync :

```
root@debian:/etc# useradd rsync
root@debian:/etc# passwd rsync
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
root@debian:/etc#
```

```
root@debian:/etc# groupadd rsync
groupadd : le groupe 'rsync' existe déjà
root@debian:/etc# gpasswd -a rsync rsync
Ajout de l'utilisateur rsync au groupe rsync
root@debian:/etc#
```

On attribue les droits au groupe et à l'user rsync :

```
root@debian:/etc# chown -R rsync:rsync /data
drwxr-xr-x   6 rsync rsync  4096 16 mai   09:32 data
root@debian:/# chmod -R 775 /data
```

Dans la commande chown -R, l'option -R signifie "récursevement". Cela indique que la commande doit s'appliquer non seulement au répertoire spécifié, mais aussi à tous les sous-répertoires et fichiers qu'il contient.

Pour configurer rsync, on crée le fichier de configuration de rsync en éditant le fichier /etc/rsyncd.conf :

```
GNU nano 7.2                               /etc/rsyncd.conf
uid = rsync
gid = rsync

[share_rsync]
path = /data/
comment = Synchro fichiers internes
read only = false
```

Ces lignes de configuration définissent un module rsync appelé share_rsync, qui partage le répertoire /srv/intern avec des permissions de lecture et d'écriture.

On attribue une adresse ip fixe à notre deuxième machine :

```
GNU nano 7.2                                     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 ens33
iface ens33 inet static
    address 192.168.197.101
    netmask 255.255.255.0
    gateway 192.168.197.2
    dns-nameservers 192.168.197.2
```

On lance manuellement le rsync :

```
root@debian:/# /etc/init.d/rsync start
Starting rsync (via systemctl): rsync.service.
```

```
c.service.
root@debian:/etc/default# cd ..
root@debian:/etc# cd ..
root@debian:# cd ...
bash: cd: ...: Aucun fichier ou dossier de ce type
root@debian:# cd ..
root@debian:# cd /data
root@debian:/data# ls
lost+found master public test.txt
root@debian:/data# ;ls
bash: erreur de syntaxe près du symbole inattendu « ; »
root@debian:/data# ls
lost+found master public test.txt
root@debian:/data# rsync -av --stats --delete --force --ignore-errors /data/public /data/master 192.168.76.101::share_rsync/
sending incremental file list
master/
master/coucou/

Number of files: 12 (reg: 8, dir: 4)
Number of created files: 0
Number of deleted files: 0
Number of regular files transferred: 0
Total file size: 69 bytes
Total transferred file size: 0 bytes
Literal data: 0 bytes
Matched data: 0 bytes
File list size: 0
File list generation time: 0,004 seconds
File list transfer time: 0,000 seconds
Total bytes sent: 484
Total bytes received: 46

sent 484 bytes received 46 bytes  1.060,00 bytes/sec
total size is 69 speedup is 0,12
root@debian:/data# rsync -av --stats --delete --force --ignore-errors /data/public /data/master 192.168.76.101::share_rsync/_
```

Automatisation avec crontab pour que le rsync s'execute toutes les dix minutes :

```
root@debianNAS:/data/jordan# nano /etc/crontab
```

```
GNU nano 7.2                                     /etc/crontab *
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .---- day of month (1 - 31)
# | | | .--- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .-- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | |
# * * * * * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }
47 6    * * 7   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }
52 6    1 * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }
#
10 * * * * root rsync -av --stats --delete --force --ignore-errors /data/Public /data/jordan /data/laPlateforme 192.168.197.101::share_rsync/ ]
```

-a: Archive mode. Cela signifie que rsync copie les fichiers de manière récursive et préserve les permissions, les timestamps, les groupes, les propriétaires, etc.

-v:Verbose. Cela rend rsync plus bavard, affichant des informations sur les fichiers transférés.

--stats: Affiche les statistiques de transfert à la fin de la synchronisation.

--delete: Supprime les fichiers du répertoire de destination qui ne sont plus présents dans le répertoire source.

--force: Force la suppression des répertoires non vides pour les rendre identiques au répertoire source.

--ignore-errors: Continue le processus même si des erreurs sont rencontrées.

Les chemins suivants sont les répertoires source que rsync va synchroniser :

/data/public

/data/jordan

/data/laplateforme

Et la destination est :

192.168.197.101::share_rsync/

```
root@debianNAS:/data# /etc/init.d/cron reload
```

Monitoring avec cockpit

Qu'est ce que c'est ?

Cockpit est un outil d'administration graphique accessible via une interface web qui facilite la gestion des serveurs Linux, y compris les NAS (Network Attached Storage). Voici comment il peut être utilisé pour gérer un NAS :

Gestion des Disques et des Partitions : Visualiser, formater et partitionner les disques ; gérer les systèmes de fichiers (ext4, Btrfs, XFS, etc.).

Volumes Logiques et RAID : Créer et gérer des volumes logiques avec LVM ; configurer et surveiller des arrays RAID.

Partages de Réseau : Configurer et gérer des partages de fichiers via Samba et NFS ; définir les permissions et surveiller l'utilisation.

Surveillance des Ressources : Voir l'utilisation du disque en temps réel ; configurer des alertes et consulter les journaux du système.

Sauvegardes et Restauration : Gérer les outils de sauvegarde ; utiliser des snapshots avec Btrs ou LVM pour des points de restauration.

Extensions et Intégrations : Ajouter des plugins spécifiques (comme Docker) ; intégrer avec d'autres outils d'administration de NAS.

Sécurité et Accès : Utiliser HTTPS pour des connexions sécurisées ; gérer les utilisateurs et les permissions d'accès.

Cockpit simplifie et centralise la gestion des ressources de stockage et des services réseau sur un NAS, offrant une interface intuitive pour les tâches d'administration courantes.

Installation

On installe cockpit :

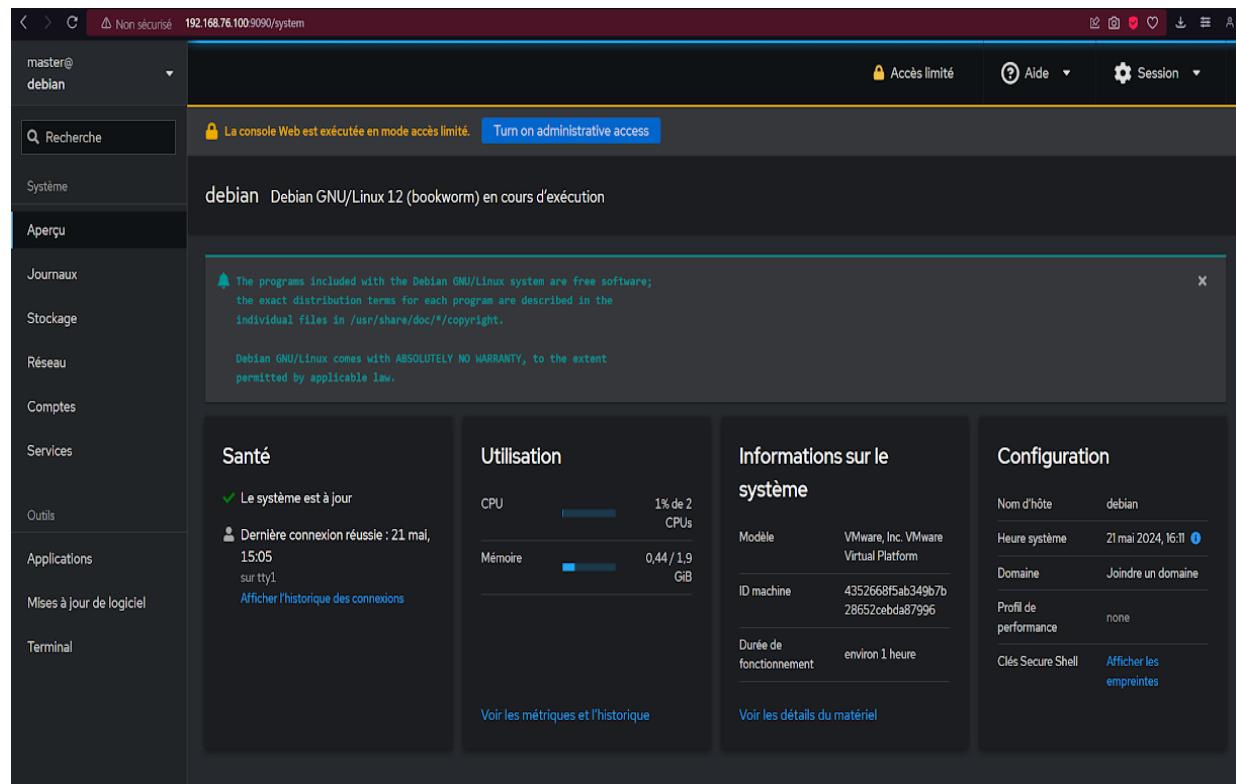
```
root@debian:/home/master# apt install cockpit_
```

```
root@debian:/home/master# systemctl enable --now cockpit.socket
```

cockpit.socket : cockpit est un outil d'administration système à distance, et il utilise des sockets pour établir des connexions. Le fichier d'unité cockpit.socket gère les connexions entrantes sur le port utilisé par Cockpit (en général, le port 9090)

On va sur notre navigateur et on rentre notre adresse ip avec le port 9090 par exemple :
<https://192.168.197.100:9090/>

On obtient le résultat suivant :



The screenshot shows the Cockpit web interface running on a Debian system. The top navigation bar includes tabs for 'Système', 'Aperçu', 'Journaux', 'Stockage', 'Réseau', 'Comptes', 'Services', 'Outils', 'Applications', 'Mises à jour de logiciel', and 'Terminal'. The main content area displays system status and configuration details:

- Santé:** Shows the system is up-to-date and last connected on May 21, 15:05.
- Utilisation:** CPU usage is at 1% of 2 CPUs, and memory usage is at 0.44/1.9 GiB.
- Informations sur le système:** Model is VMware, Inc. VMware Virtual Platform, ID machine is 4352668f5ab34967b28652cebd87996, and duration of operation is about 1 hour.
- Configuration:** Host name is debian, system time is May 21 2024, 16:11, domain is joined, performance profile is none, and SSH key fingerprint is available.

A message at the top indicates the console is in restricted mode and provides a link to turn on administrative access.

On peut également passer en administrateur le mieux pour cela est d'aller modifier le fichier sudoers pour avoir des droits permissifs sans pour autant être équivalent à des droits roots sur tous les fichiers :

```
GNU nano 7.2                               /etc/sudoers
# Ditto for GPG agent
Defaults:Defaults:+= "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

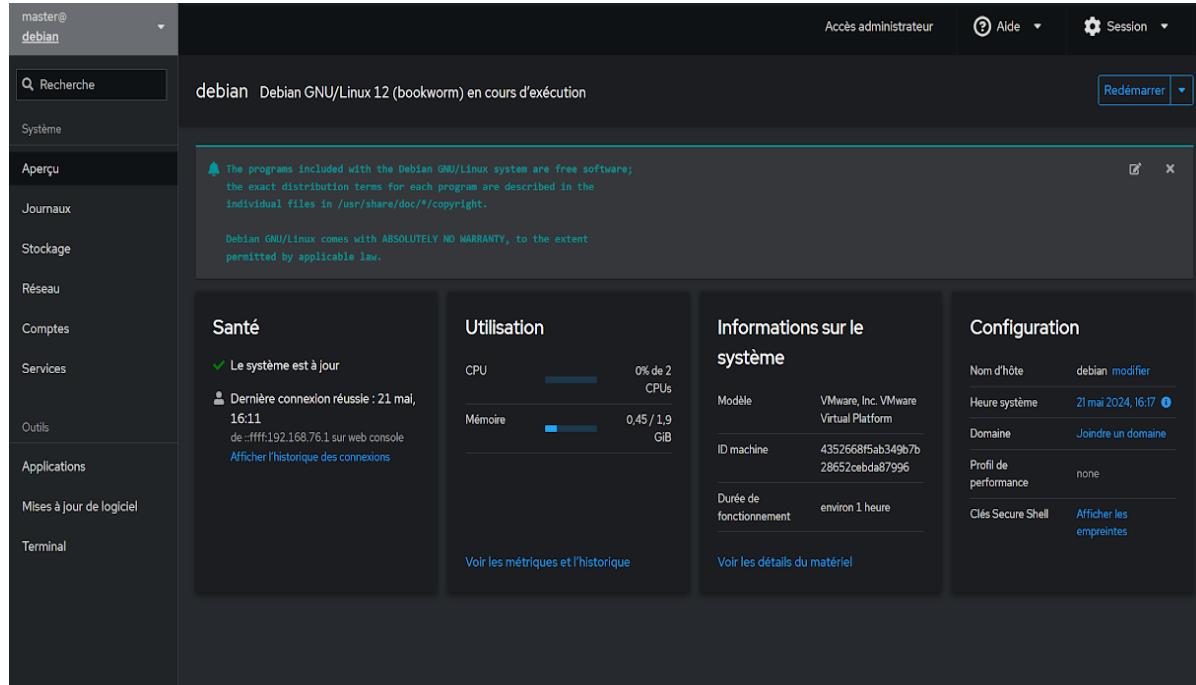
# User privilege specification
root    ALL=(ALL:ALL) ALL

#-----
# User 'laplateforme' privilege specification
#-----
# Supervision du système:
laplateforme ALL=(ALL) NOPASSWD: /usr/bin/top, /usr/bin/htop, /usr/bin/vmstat, /usr/bin/iostat
# Gestion/création/suppression d'utilisateur
laplateforme ALL=(ALL) NOPASSWD: /usr/sbin/useradd, /usr/sbin/userdel, /usr/sbin/usermod
# Gestion/création/suppression d'un groupe
laplateforme ALL=(ALL) NOPASSWD: /usr/sbin/groupadd, /usr/sbin/groupdel, /usr/sbin/groupmod
# Gestion des sessions utilisateur:
laplateforme ALL=(ALL) NOPASSWD: /usr/bin/kill, /usr/bin/pkill, /usr/bin/who, /usr/bin/w
# Modification des autorisations d'accès au dossier
laplateforme ALL=(ALL) NOPASSWD: /bin/chmod, /bin/chown, /bin/chgrp, /bin/mkdir, /bin/rmdir

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:
@includedir /etc/sudoers.d
```

Affichage en administrateur :



Avoir des droits administrateurs sur Cockpit permet de tirer pleinement parti de ses capacités pour gérer un NAS de manière efficace, sécurisée et personnalisée. Les administrateurs disposent d'un contrôle total sur le système, ce qui leur permet de configurer, surveiller, et maintenir le NAS en fonction des besoins spécifiques de l'organisation, tout en assurant une sécurité robuste et une gestion optimisée des ressources.