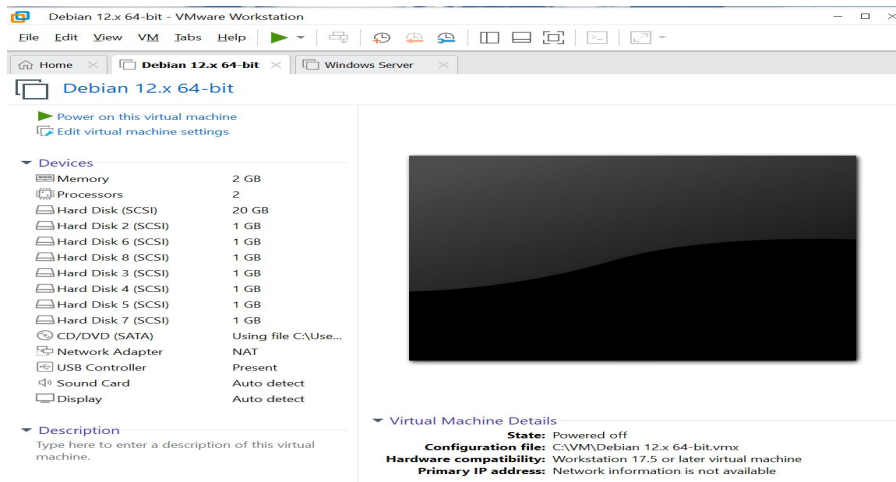


SUPPPORT ET MISE À DISPOSITION

PROF : M. Vincent KRAHENBUHL

Compte Rendu - Création de Raids sur Linux

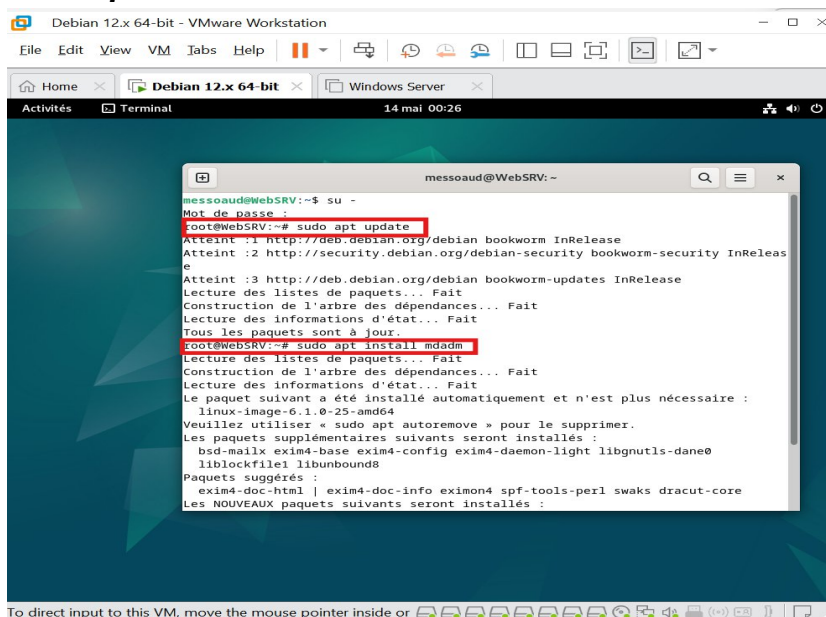


1. Installation de mdadm

Debian ne vient pas avec l'outil **mdadm** préinstallé. Je l'ai donc installé avec la commande suivante :

sudo apt update

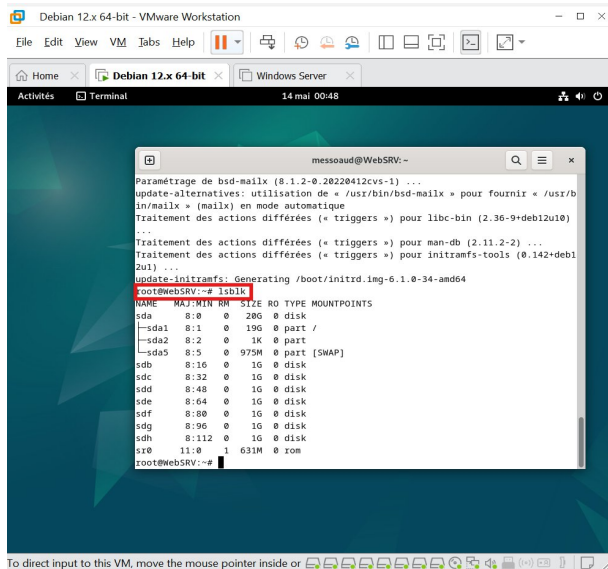
sudo apt install mdadm



To direct input to this VM, move the mouse pointer inside or

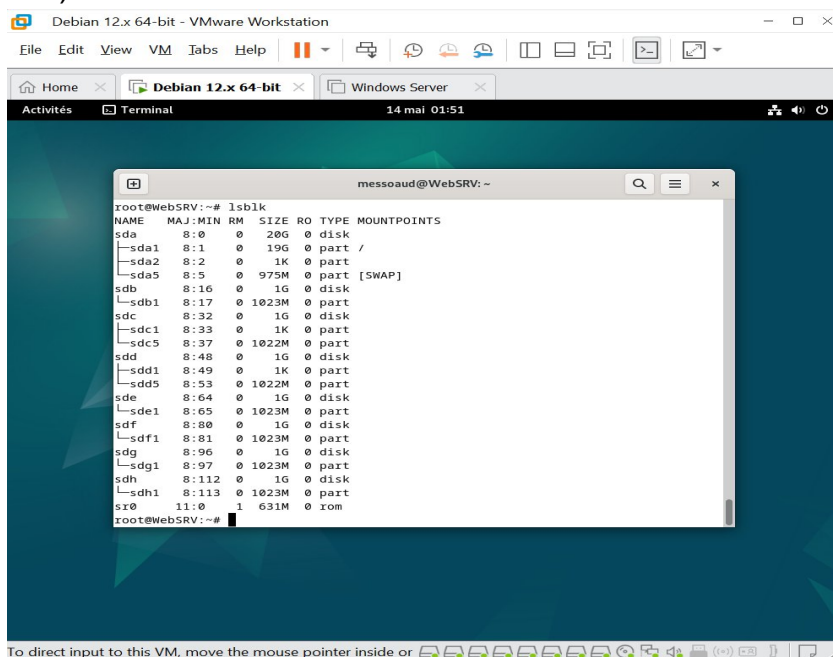
2.Préparation des disques

J'ai identifié les disques avec la commande **lsblk**



```
Paramétrage de bsd-mailx (8.1.2-0.20220412cvs-1) ...
update-alternatives: utilisation de « /usr/bin/bsd-mailx » pour fournir « /usr/b
in/mailx » (mailx) en mode automatique
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u10)
...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour initramfs-tools (0.142+debi
2u1) ...
update-initramfs: Generating /boot/initrd.img-6.1.0-34-amd64
root@WebSRV:~# 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 1G 0 disk
sdc 8:32 0 1G 0 disk
sdd 8:48 0 1G 0 disk
sde 8:64 0 1G 0 disk
sdf 8:80 0 1G 0 disk
sdg 8:96 0 1G 0 disk
sdh 8:112 0 1G 0 disk
sxi 11:0 1 631M 0 rom
root@WebSRV:~#
```

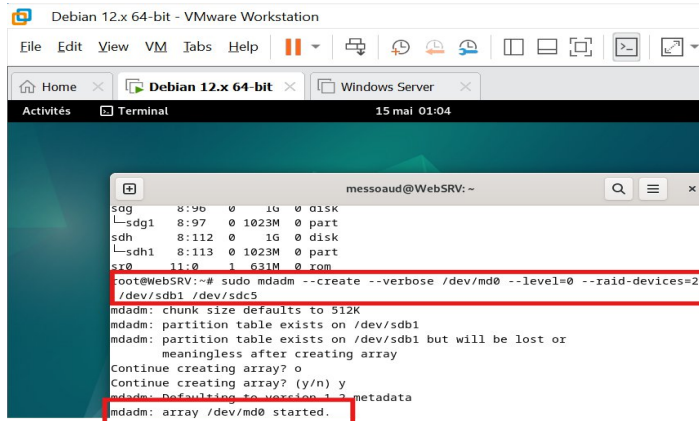
Puis j'ai utilisé **fdisk** pour créer une partition sur chaque disque. Par exemple, pour le disque **/dev/sdb**, j'ai fait : **sudo fdisk /dev/sdb**
J'ai répété cela pour tous les autres disques nécessaires (sdc, sdd, sde, sdf, sdg, sdh)



```
root@WebSRV:~# 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 1G 0 disk
├─sdb1 8:17 0 1023M 0 part
sdc 8:32 0 1G 0 disk
├─sdc1 8:33 0 1K 0 part
├─sdc5 8:37 0 1022M 0 part
sdd 8:48 0 1G 0 disk
├─sdd1 8:49 0 1K 0 part
├─sdd5 8:53 0 1022M 0 part
sde 8:64 0 1G 0 disk
├─sde1 8:65 0 1023M 0 part
sdf 8:80 0 1G 0 disk
├─sdf1 8:81 0 1023M 0 part
sdg 8:96 0 1G 0 disk
├─sdg1 8:97 0 1023M 0 part
sdh 8:112 0 1G 0 disk
├─sdh1 8:113 0 1023M 0 part
sxi 11:0 1 631M 0 rom
root@WebSRV:~#
```

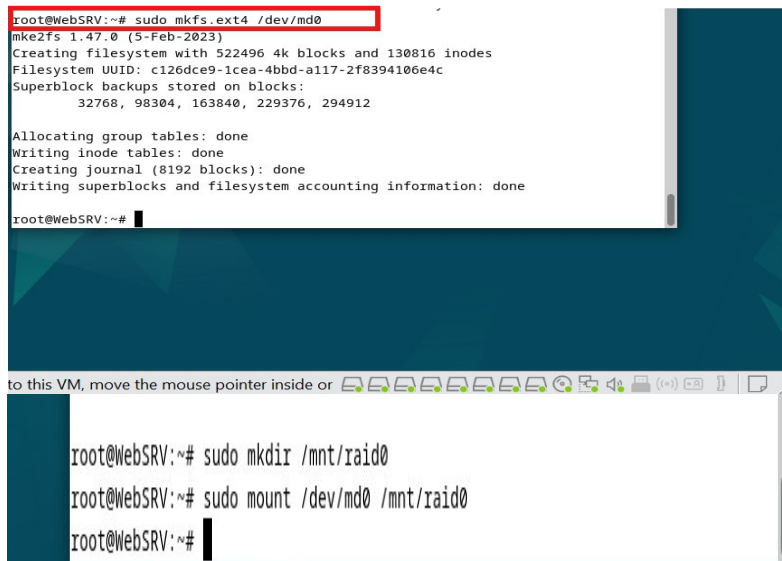
3. Création du RAID 0

Pour créer le RAID 0 avec deux partitions (/dev/sdb1 et /dev/sdc1), j'ai utilisé cette commande : **sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1**



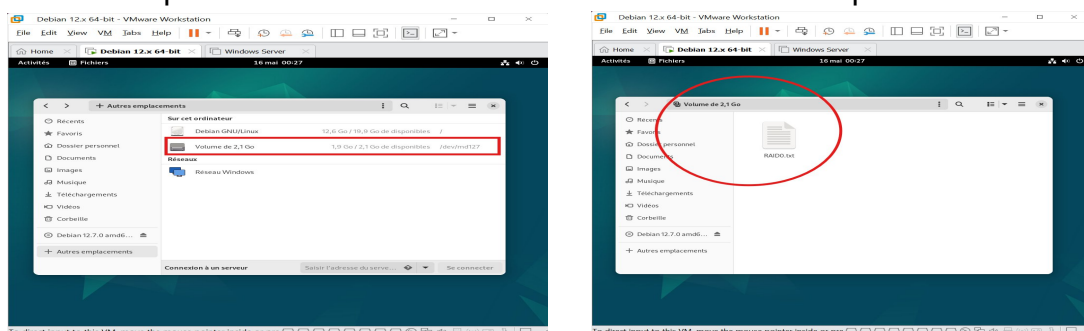
```
Debian 12.x 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home Debian 12.x 64-bit Windows Server
15 mai 01:04
messoaud@WebSRV: ~
sog 8:96 0 1G 0 disk
├─sog1 8:97 0 1023M 0 part
sdb 8:112 0 1G 0 disk
├─sdb1 8:113 0 1023M 0 part
sr0 11:0 1 631M 0 rom
root@WebSRV:~# sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1
mdadm: chunk size defaults to 512K
mdadm: partition table exists on /dev/sdb1
mdadm: partition table exists on /dev/sdb1 but will be lost or
meaningless after creating array
Continue creating array? o
Continue creating array? (y/n) y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

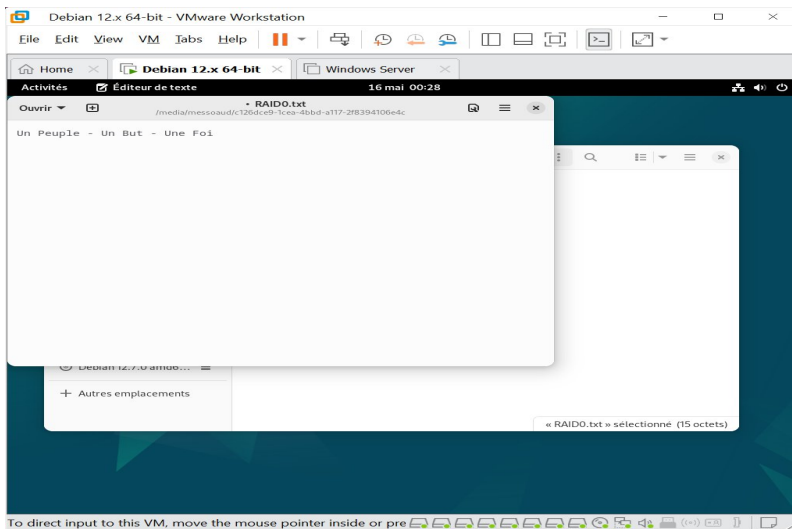
Une fois le RAID créé, je l'ai formaté avec **mkfs.ext4**, puis monté dans **/mnt/raid0**. J'ai également créé un fichier de test nommé RAID0.txt



```
root@WebSRV:~# sudo mkfs_ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 522496 4k blocks and 130816 inodes
Filesystem UUID: c126dce9-1cea-4bbd-a117-2f8394106e4c
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912
Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
root@WebSRV:~#
root@WebSRV:~# sudo mkdir /mnt/raid0
root@WebSRV:~# sudo mount /dev/md0 /mnt/raid0
root@WebSRV:~#
```

Ainsi nous pouvons voir le fichier.txt dans /mnt/raid0 via l'explorateur de fichier.

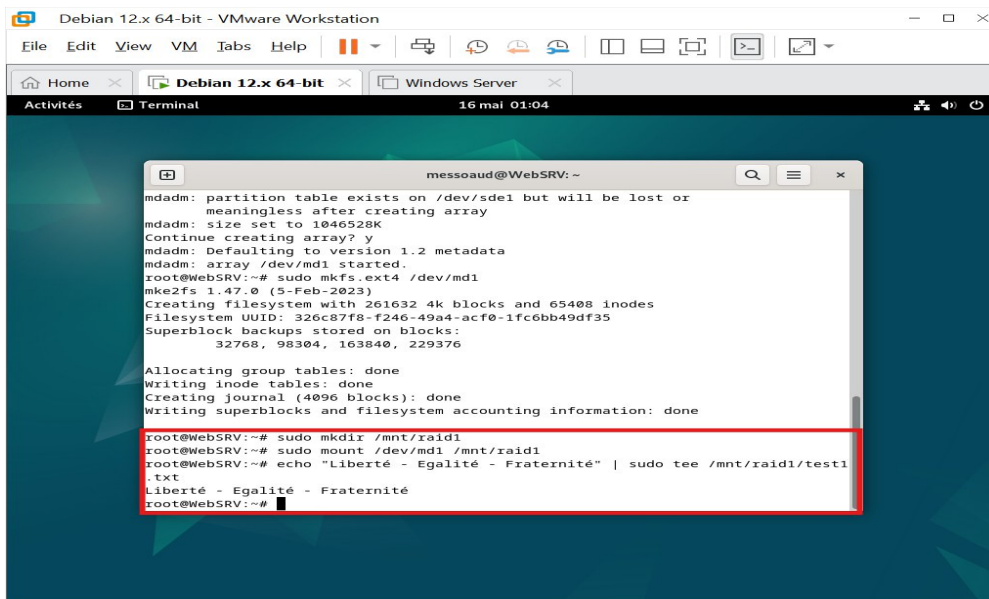




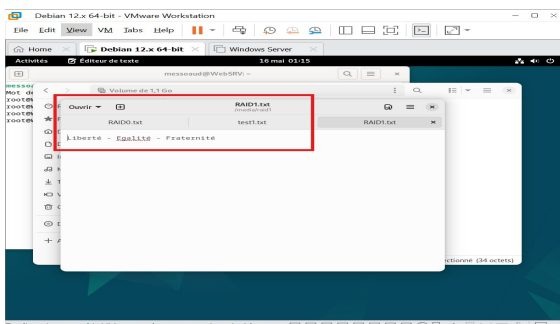
To direct input to this VM, move the mouse pointer inside or pre

4.Création du RAID 1

De la même manière, j'ai créé un RAID 1 avec `/dev/sdd1` et `/dev/sde1` :



To direct input to this VM, move the mouse pointer inside or pre



To direct input to this VM, move the mouse pointer inside or pre

5. Création du RAID 5

De la même manière, j'ai créé un RAID 5 avec ***/dev/sdf1***, ***/dev/sdg1*** et ***/dev/sdh1***.

```
messoaud@WebSRV: ~  
mdadm: partition table exists on /dev/sdh1  
mdadm: partition table exists on /dev/sdh1 but will be lost or  
meaningless after creating array  
mdadm: size set to 1043504K  
Continue creating array? y  
mdadm: Defaulting to version 1.2 metadata  
mdadm: array /dev/md2 started.  
root@WebSRV:~# sudo mkfs.ext4 /dev/md2  
mkfs2fs 1.47.0 (5-Feb-2023)  
Creating filesystem with 522752 4k blocks and 130816 inodes  
Filesystem UUID: 55fa7abb-2c0f-4a49-ba2b-1de6ae93be8  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (8192 blocks): done  
Writing superblocks and filesystem accounting information: done  
root@WebSRV:~# sudo mkdir /mnt/raid5  
root@WebSRV:~# sudo mount /dev/md2 /mnt/raid5  
root@WebSRV:~# echo "In god we trust" | sudo tee /mnt/raid5/RAID5.txt  
In god we trust  
root@WebSRV:~#
```

6. Sauvegarde de la configuration

Pour que les volumes RAID soient détectés au redémarrage, j'ai sauvegardé la configuration en utilisant les commandes suivantes :

```
root@WebSRV:~# sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf  
ARRAY /dev/md/WebSRV:0 metadata=1.2 name=WebSRV:0 UUID=54653b7e:4618c2be:47895e44:dc161d54  
ARRAY /dev/md1 metadata=1.2 name=WebSRV:1 UUID=fc3c5676:3be358c4:42f51ad6:3105d9bc  
ARRAY /dev/md2 metadata=1.2 name=WebSRV:2 UUID=d3d8596b:8a1c2537:8fd6725a:05f664c8  
root@WebSRV:~# sudo update-initramfs -u  
update-initramfs: Generating /boot/initrd.img-6.1.0-34-amd64  
root@WebSRV:~#
```

7. Test de panne

J'ai simulé une panne sur chaque RAID en "défaillant" un disque avec la commande ***sudo mdadm /dev/mdX --fail /dev/sdXY***

- ***/dev/mdX*** étant le périphérique RAID (md0, md1)
- ***/dev/sdXY*** étant le disque ou la partition que l'on veut simuler comme étant en fail (sdb1 pour le raid 0)

Cette manipulation nous a permis de comprendre le fonctionnement et les différences entre les types de RAID.

• RAID 1

J'ai marqué un des deux disques du RAID 1 comme défectueux.

→ Le système a continué à fonctionner normalement, sans perte de données.

Cela est dû au **principe de la redondance** dans le RAID 1 : les données sont copiées à l'identique sur les deux disques.

• RAID 5

J'ai simulé la défaillance de l'un des trois disques du RAID 5.

→ Le RAID est resté **fonctionnel** : le système a utilisé les **informations de parité** pour continuer à servir les données.

Cela confirme que le RAID 5 **supporte la perte d'un disque** sans perte de données.

• RAID 0

J'ai tenté de simuler une panne sur l'un des deux disques du RAID 0, mais :

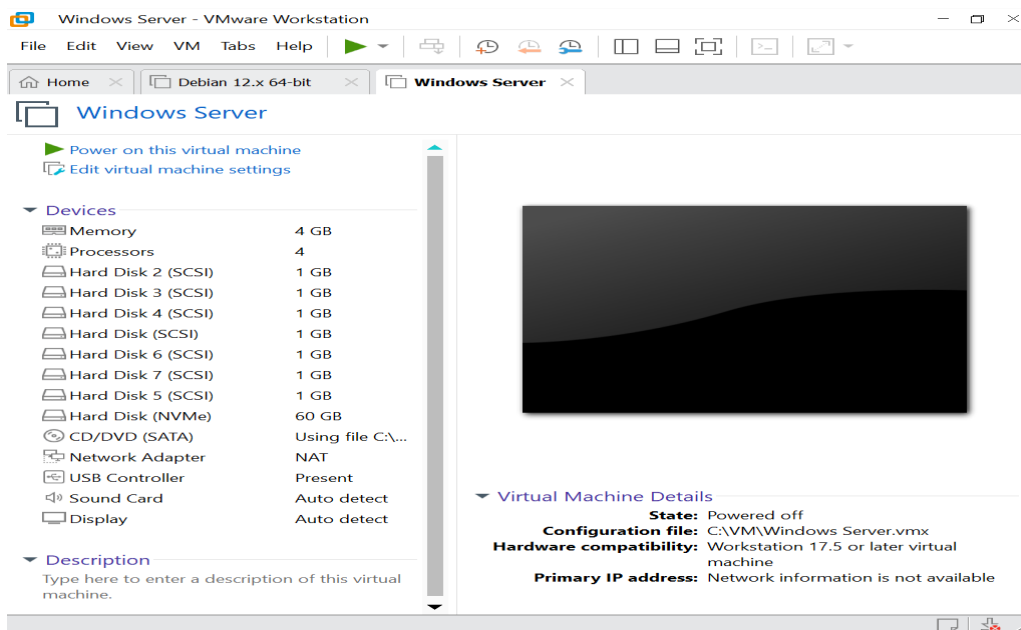
- Le système a immédiatement indiqué une **erreur fatale**.
- Le volume RAID 0 est devenu inutilisable.

Cela est logique car le RAID 0 **n'a aucune redondance** : les données sont réparties (strippées) sur les deux disques. Si un disque est perdu, la moitié des données est irrécupérable.

SUPPORT ET MISE À DISPOSITION

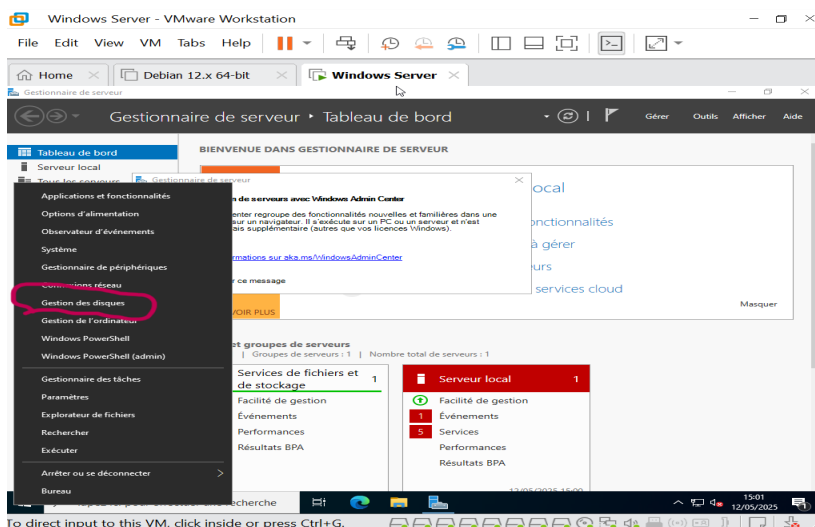
PROF : M. Vincent KRAHENBUHL

Compte Rendu – Création de Raids sur Windows Server



1. Créer un RAID 0 avec 2 disques (agrégé par bandes)

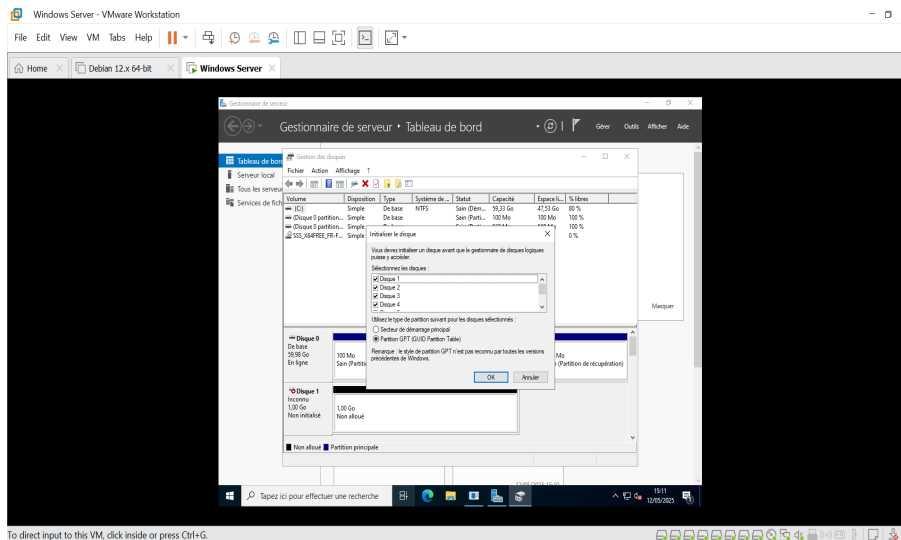
Pour commencer, j'ai configuré un RAID 0 à l'aide de deux disques durs de 1 Go. Je me suis rendu dans le **Gestionnaire de disques**.



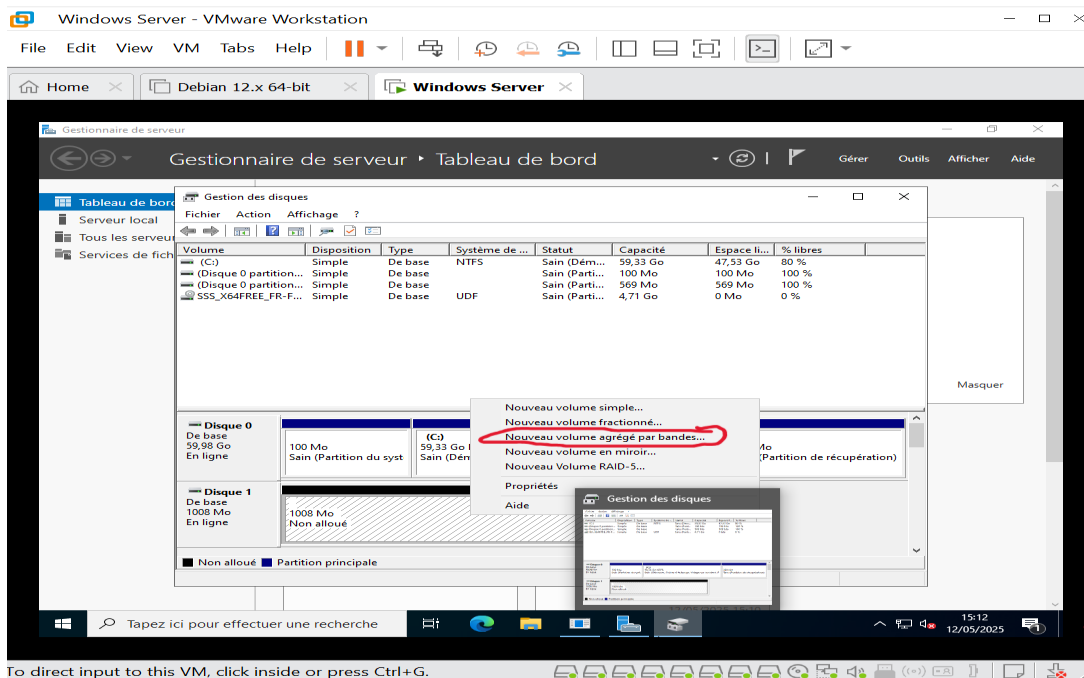
J'ai opté pour le **format GPT (GUID Partition Table)** pour tous les disques utilisés.

Le choix de GPT est important pour plusieurs raisons :

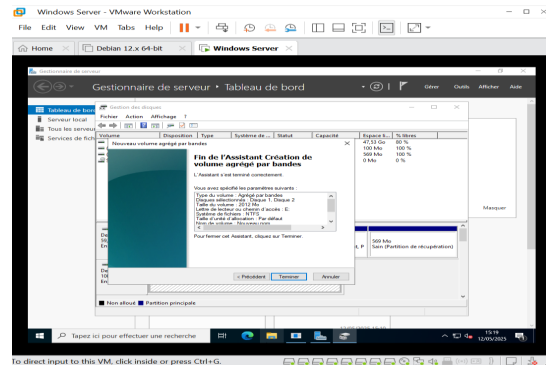
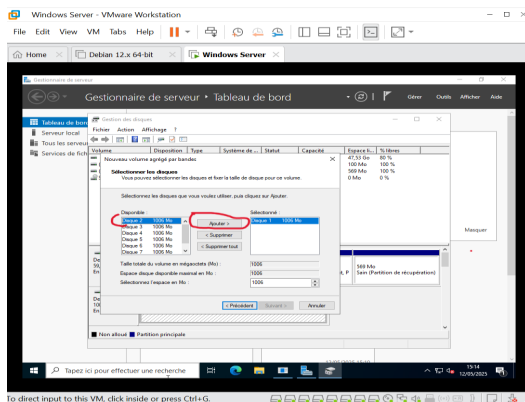
- GPT permet de **gérer des disques de plus de 2 To** (même si ici les disques font 1 Go, cela reste une bonne pratique).
- GPT est **recommandé pour les systèmes modernes** car il est plus robuste et compatible avec l'UEFI.
- Contrairement à MBR, GPT permet **plus de 4 partitions principales** par disque.



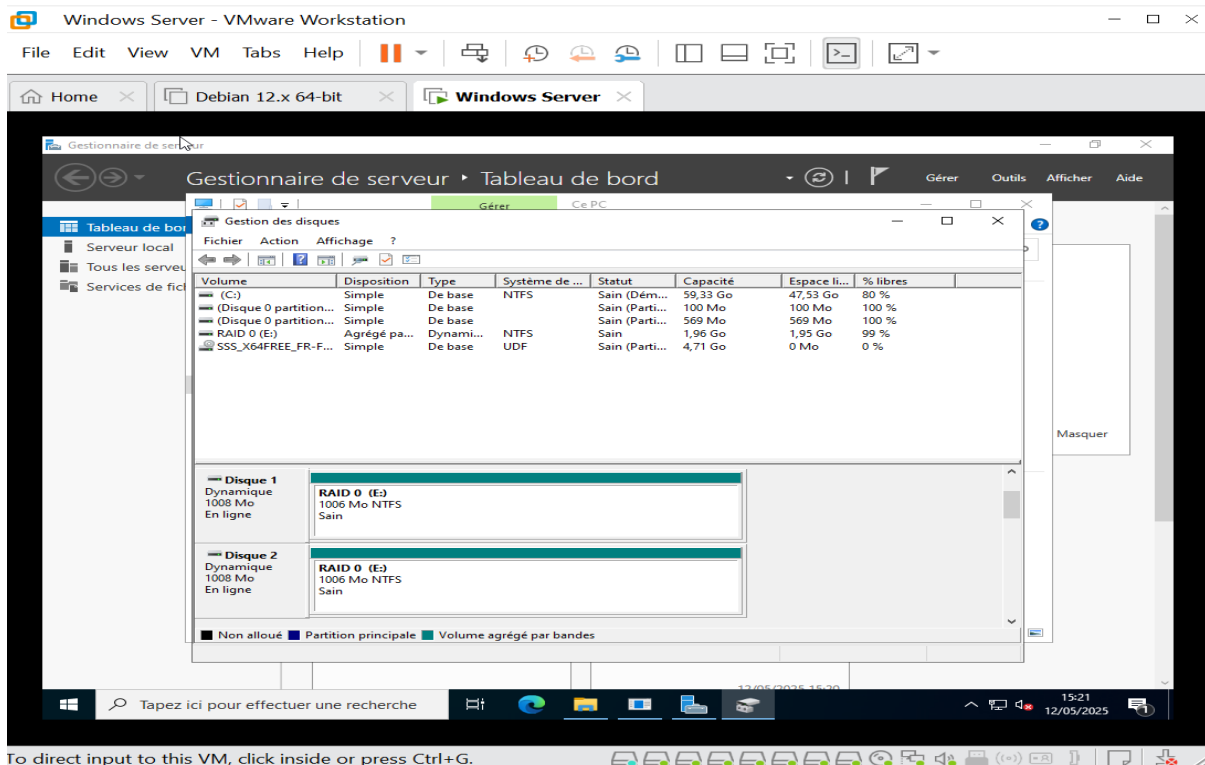
Puis j'ai effectué un clic droit sur l'un des disques non alloués. J'ai choisi l'option "Nouveau volume agrégé par bandes".



L'assistant de création de volume s'est lancé. J'ai sélectionné les deux disques disponibles pour constituer ce RAID 0. J'ai ensuite attribué une lettre "E" de lecteur au volume, puis je l'ai formaté en NTFS avec une étiquette de volume explicite (RAID 0).



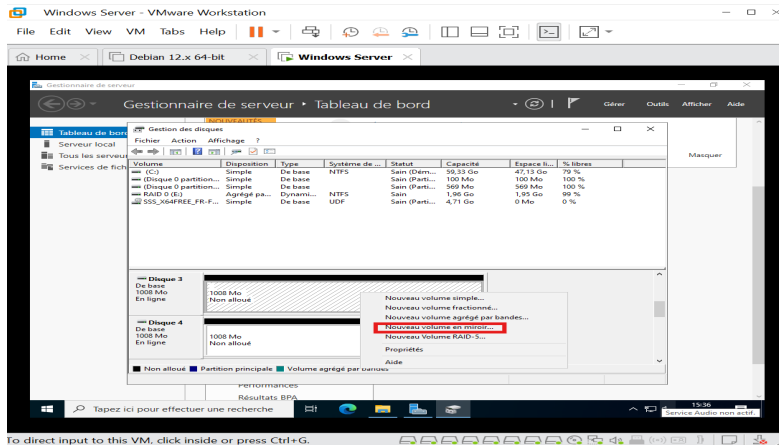
À la fin de la procédure, le volume RAID 0 était créé et monté dans l'explorateur Windows local comme nous pouvons le voir sur l'image ci-dessous.



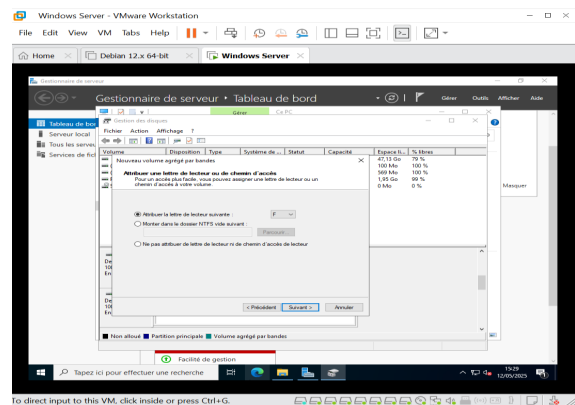
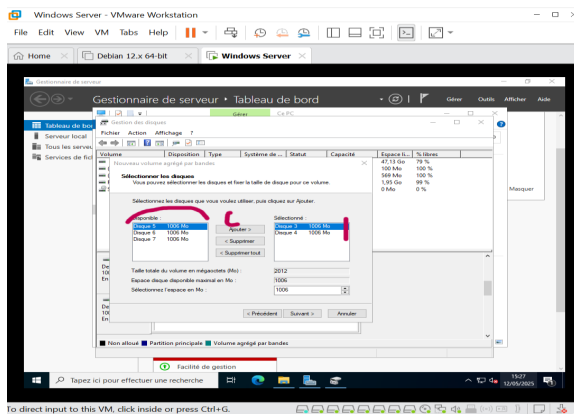
2. Créer un RAID 1 (miroir) avec 2 disques

Comme précédemment, je me suis rendu dans le **Gestionnaire de disques** en faisant un clic droit sur le menu démarrer ensuite je clique sur **Gestion des disques**.

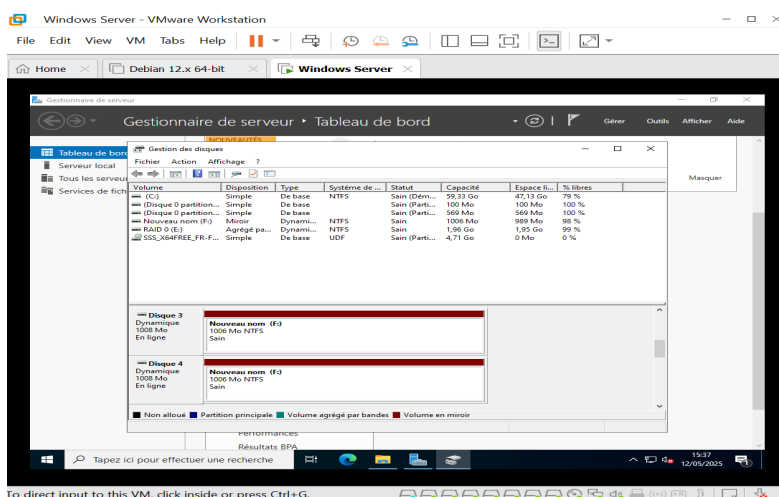
Et cette fois ci, j'ai sélectionné "**Nouveau volume en miroir**" en faisant un clic droit sur l'un des disques non alloués.



Dans l'assistant, j'ai ajouté le second disque pour former le miroir. J'ai ensuite attribué une lettre "F" de lecteur au volume, puis je l'ai formaté en NTFS avec une étiquette de volume explicite (RAID 1).

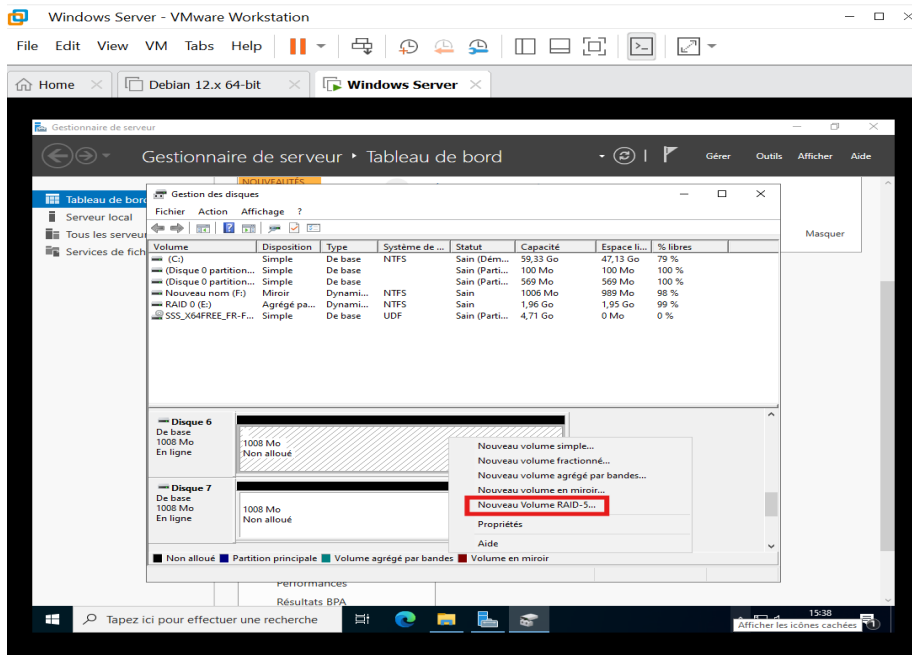


À la fin de la procédure, le volume RAID 1 était créé et monté dans l'explorateur Windows comme nous pouvons le voir sur l'image ci-dessous.

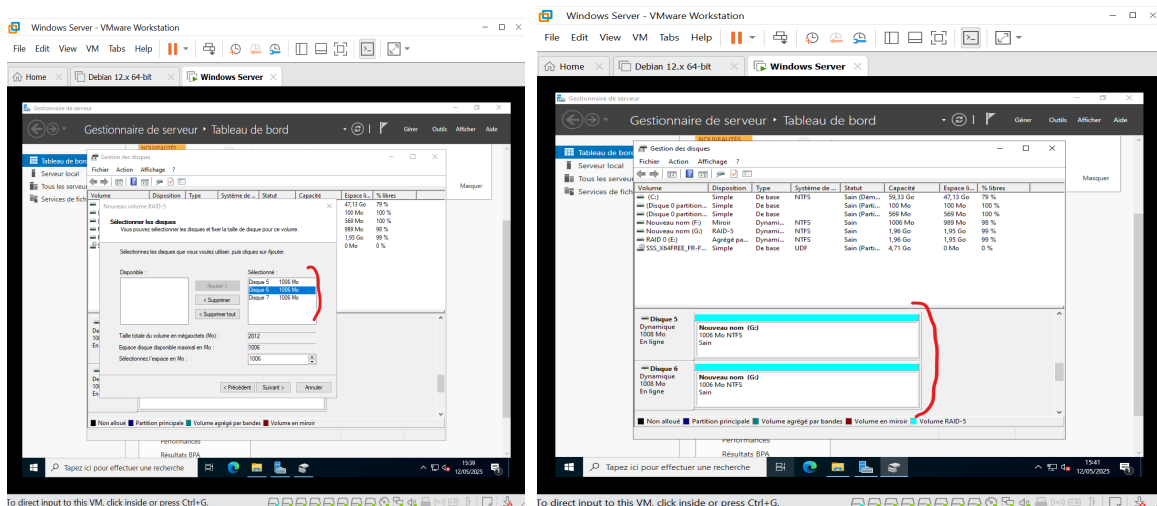


3. Créer un RAID 5 (parité) avec 3 disques

Pour cette fois-ci, j'ai sélectionné "Nouveau volume en RAID-5" en faisant un clic droit sur l'un des disques non alloués.

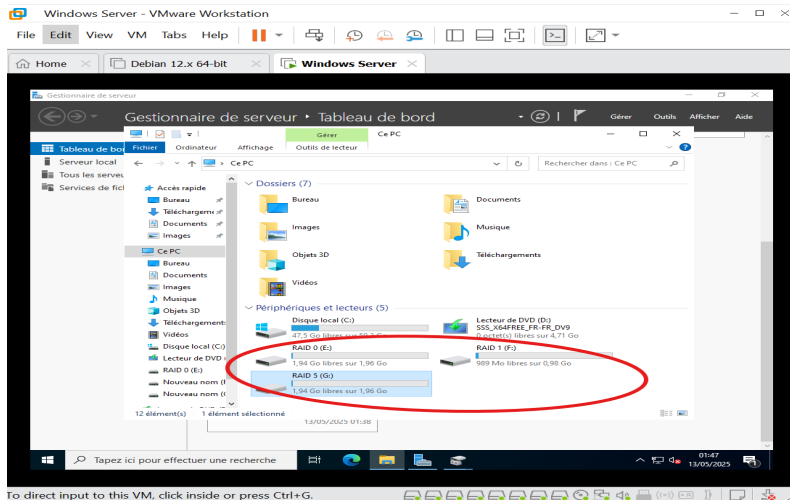


J'ai suivi les étapes de l'assistant en ajoutant les deux autres disques au volume. J'ai attribué une lettre de lecteur, formaté le volume en NTFS, et nommé le volume "RAID5".

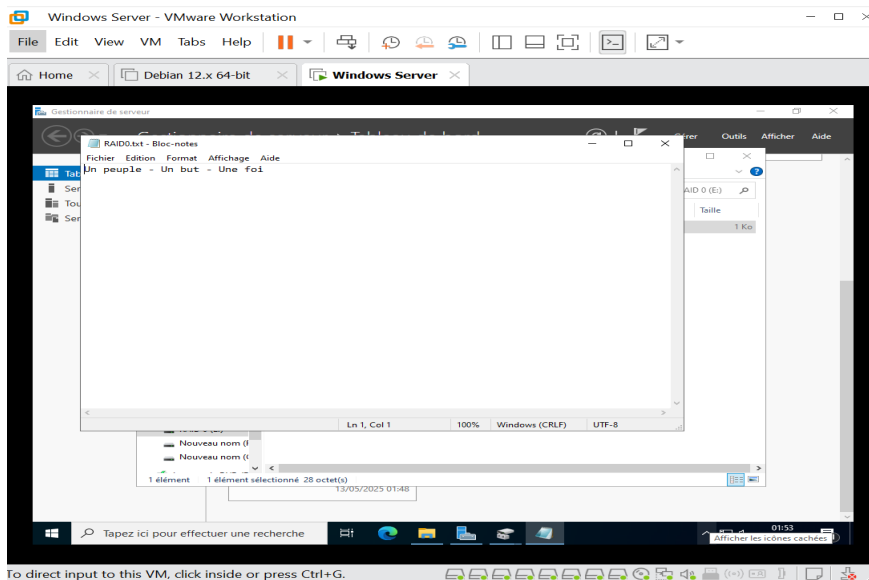


4. Création d'un fichier test sur chacun des RAID

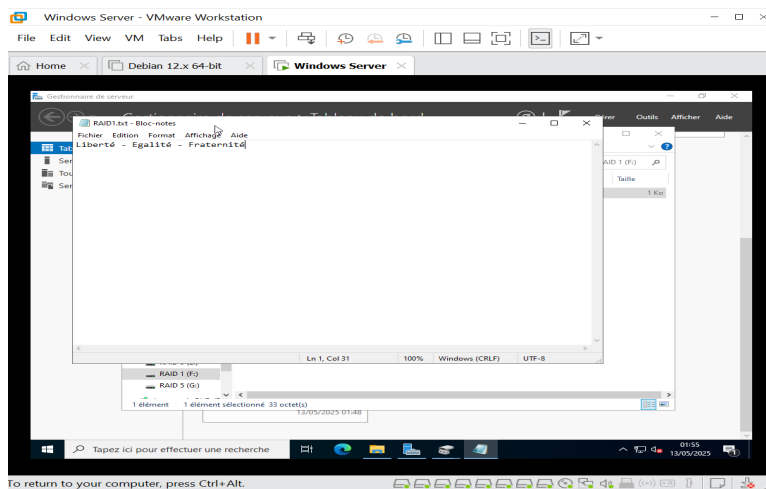
Pour valider le fonctionnement de chaque volume, je me suis rendu dans l'Explorateur de fichiers. Dans chaque volume RAID, j'ai créé un fichier texte de test.



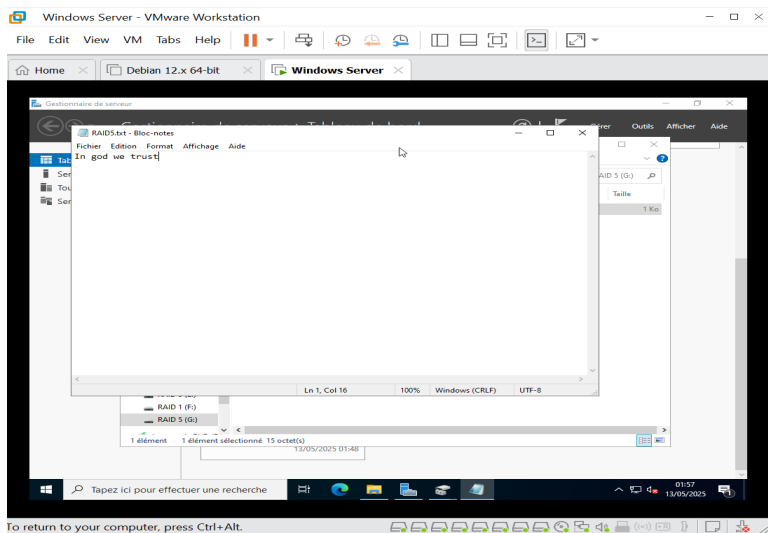
- Un fichier nommé RAID0.txt dans le volume RAID 0



- Un fichier nommé RAID1.txt dans le volume RAID 1

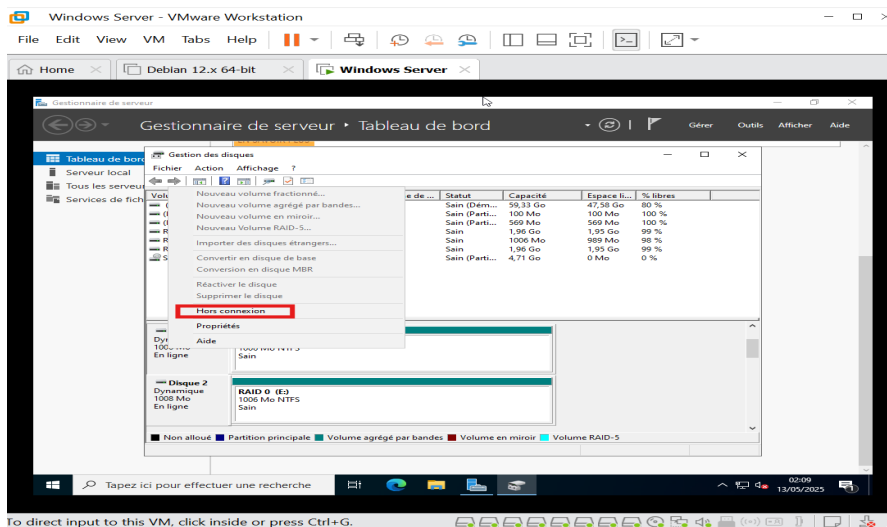


- Un fichier nommé RAID5.txt dans le volume RAID 5



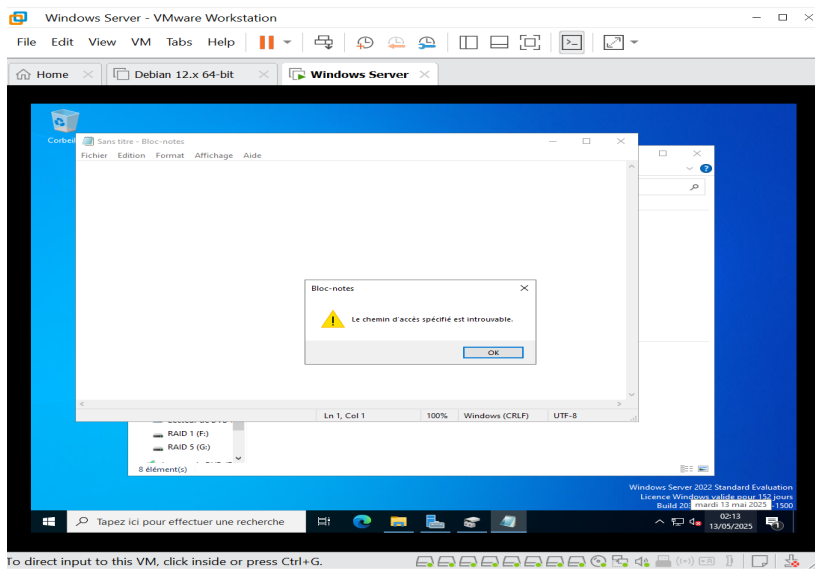
5. Tester la coupure d'un disque par RAID et observer les résultats

Pour tester la tolérance aux pannes des différents types de RAID, j'ai **simulé la perte d'un disque** dans chacun des volumes. Pour cela, je suis retourné dans le **Gestionnaire de disques**, j'ai identifié un disque faisant partie d'un RAID et je l'ai mis **hors connexion** (clic droit → Hors connexion).



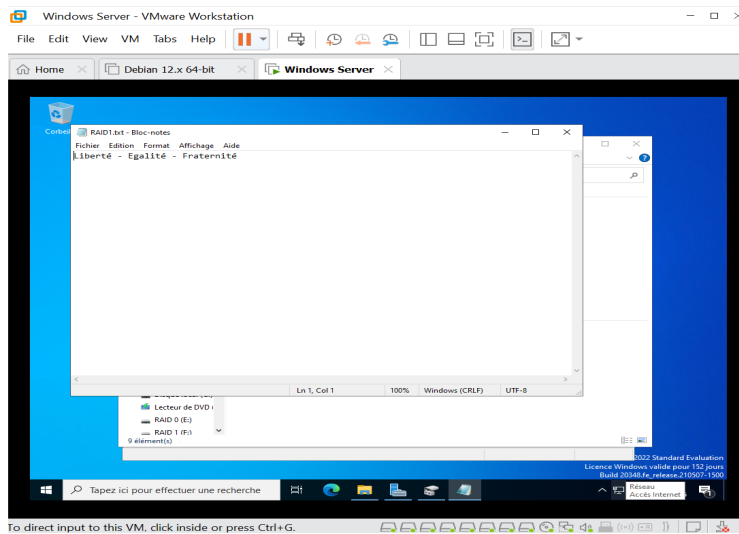
- RAID 0 :

Lorsque j'ai mis un des deux disques du RAID 0 hors ligne, le volume est devenu immédiatement **inaccessible**. Le fichier RAID0.txt n'était plus lisible. Cela confirme que **RAID 0 ne supporte pas la perte d'un disque** et que toutes les données sont perdues.



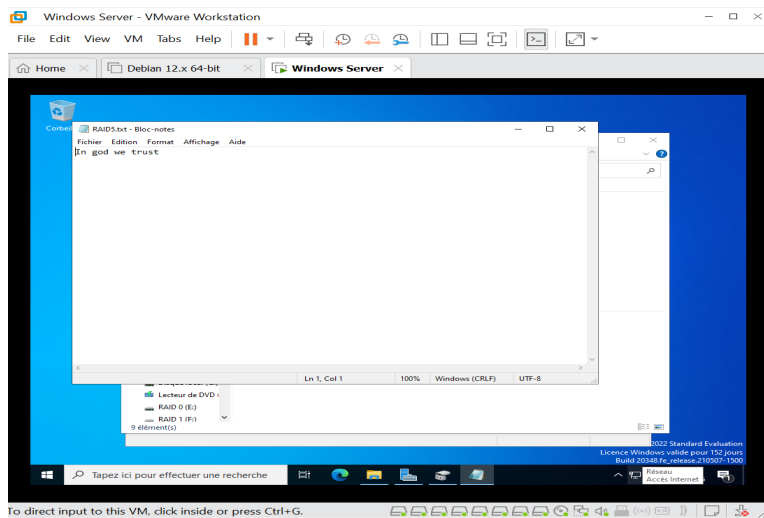
- RAID 1 :

En désactivant un des deux disques du RAID 1, le volume est resté **fonctionnel**. J'ai pu accéder au fichier RAID1.txt sans problème. RAID 1 assure bien une **redondance complète** des données.



- RAID 5 :

Enfin, pour le RAID 5, après avoir mis un disque hors ligne, le volume restait accessible et opérationnel. Le fichier RAID5.txt était toujours disponible. Cela confirme que **RAID 5 peut tolérer la perte d'un disque**, grâce au système de parité.



Conclusion

Cette manipulation m'a permis de constater le comportement des trois types de RAID les plus utilisés :

- **RAID 0** : offre de bonnes performances, mais **aucune tolérance de panne**. Toute perte d'un disque entraîne une perte totale des données.
- **RAID 1** : assure une **tolérance de panne totale** pour un disque, les données étant dupliquées.
- **RAID 5** : permet une **tolérance à une panne de disque**, avec un bon compromis entre sécurité et espace disque utilisable.