TP1-RAID-LVM-STOCKAGE LOCAL - Sûreté de fonctionnement ECUE 3.2

samedi 19 juin 2021

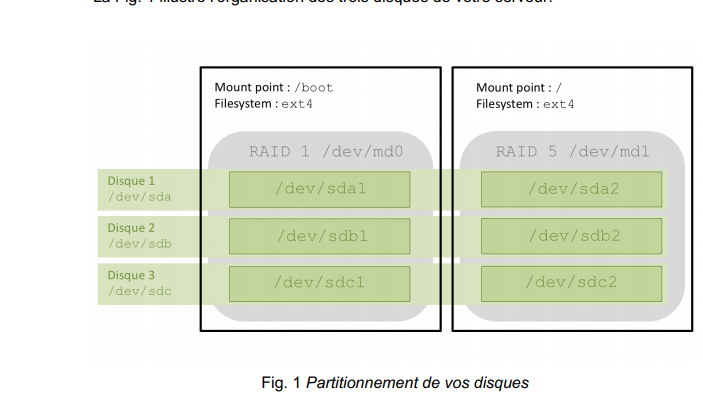
Amine ABDOUL-AZID

Kavirajan SARAVANANE

Martial SENE

**PRÉPARATION :**

* RAID : Stockage redondant sur plusieurs disques d'un même hôte
* LVM : Agrégation et gestion flexible des volumes de stockage



**Synthèse 1 :**

Les périphériques actifs sont :

Une image contenant texte

Description générée automatiquement

On voit les volumes du disque avec fdisk -l :

Une image contenant texte

Description générée automatiquement

La commande df -h

Une image contenant texte, tableau de points

Description générée automatiquement

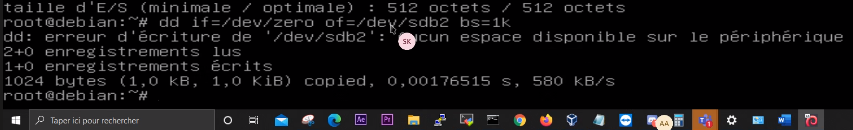
On constate que /dev/md0 et /dev/md1 corerspondent aux volumes raid

Sur la nouvelle VM : On voit le contenu :

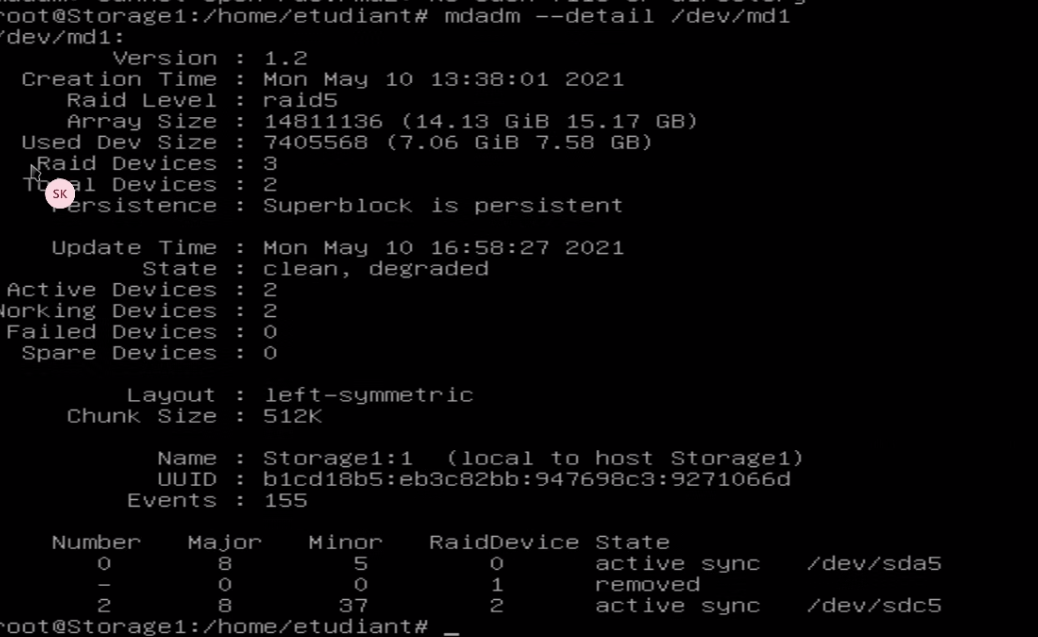
Une image contenant texte

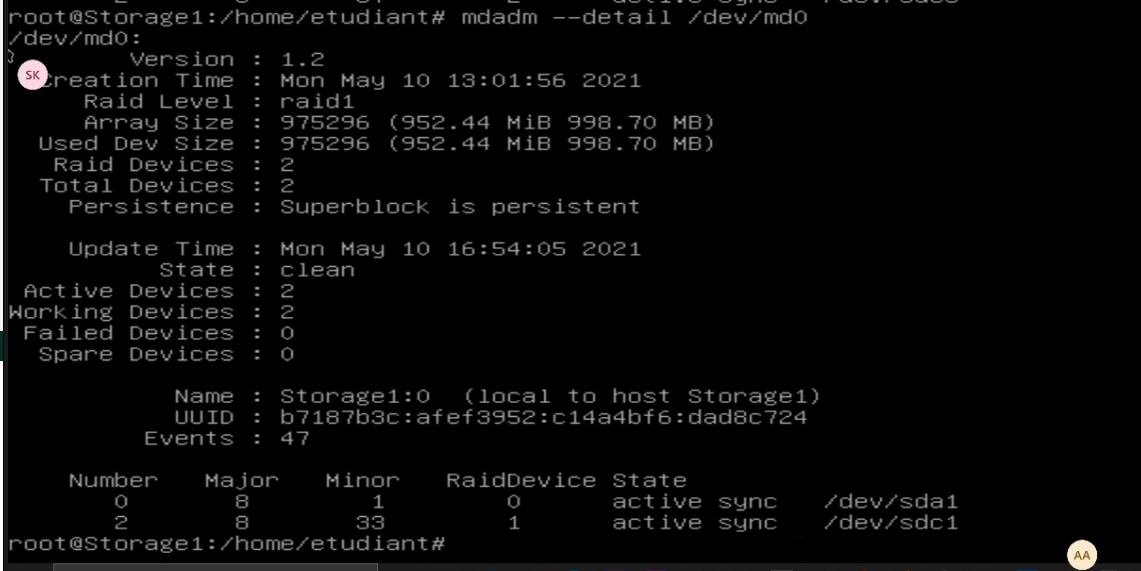
Description générée automatiquement

On efface en écrasant par des 0 le contenu de */dev/sdb2* avec une taille de bloc de 512 octets :

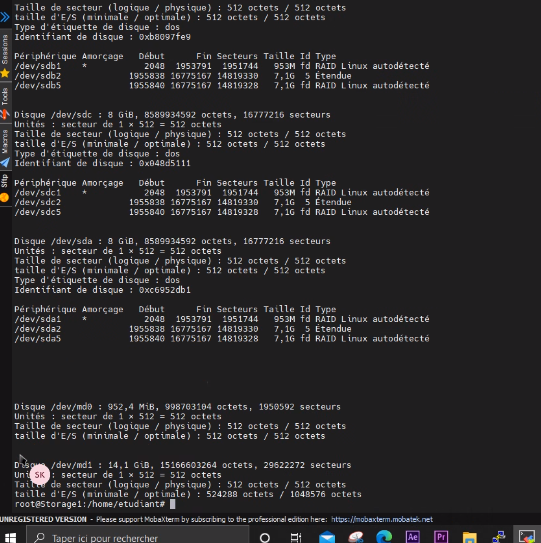


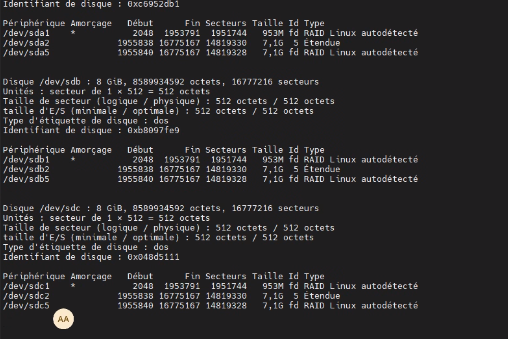
On utilise *mdadm* pour voir les détails du Volume RAID md1 :



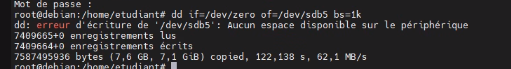
On utilise à nouveau la commande *mdadm* mais avec md2 :

On affiche les partitions avec *fdisk* :

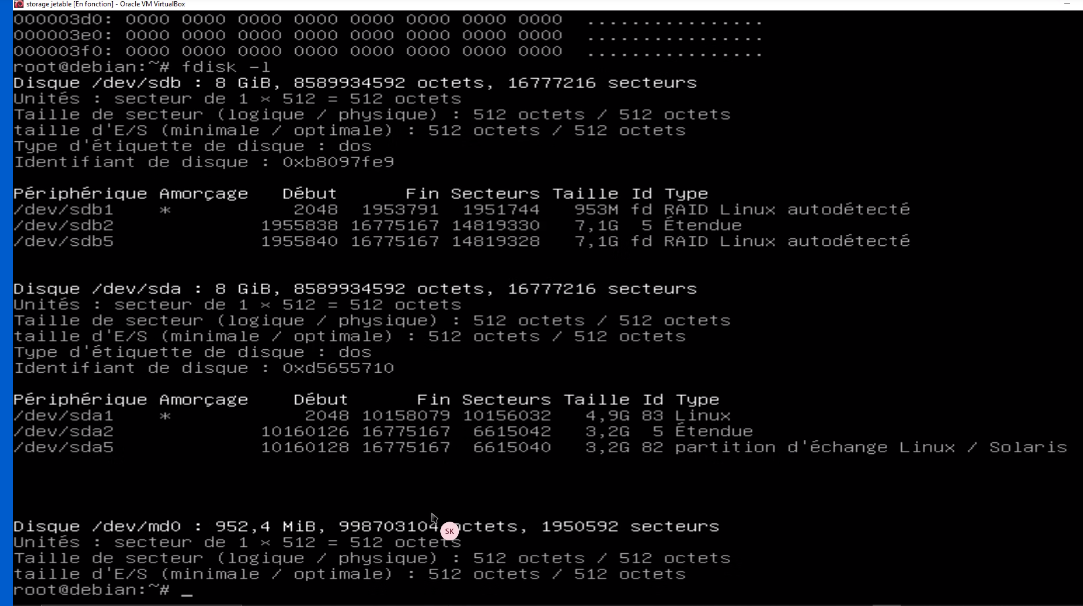




On a eu un bug avec dd qui refuse d’écrire sur la partition */dev/sdb5* :



Vérification avec *fdisk -l*



**Synthèse 1 :**

Tout d’abord, on a configuré RAID et les différents volumes avec une image ISO de Debian.

Pour cela, on a affiché leurs propriétés avec *fdisk* et *mdadm* ainsi que leur taille avec *df*.

On a arrêté Storage1 et on a débranché son second disque. Puis on a importé une VM Debian jetable.

On a compris pourquoi Storage1 démarre avec un disque illisible : Il est sans partition MBR, donnée.

Ensuite on a ajouté ce disque au volume RAID 5. En affichant l'état des volumes RAID : On a visualisé en temps réel : Sa reconstruction. Ainsi RAID a « réparé le disque défectueux après analyse ».

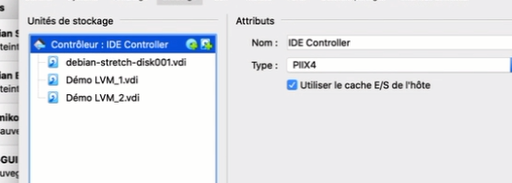
Désormais on a ajouté encore un espace de stockage à Storage1 et vérifier la table des partitions.

On a créé de ce volume RAID de niveau 5, puis on a visualisé en continu sa construction.

En outre, on a formaté ce volume et on a monté de façon persistante, démonté dans l'arborescence.

**Synthèse 2 :**

Les disques durs sont ajoutés de cette manière : *debian-stretch-disk001.vdi* en tant que disque dur primaire. Et le reste pour les volumes LVM.



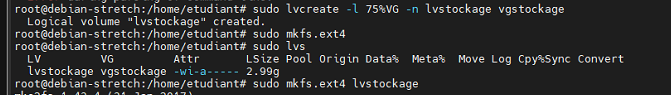
On liste le dossier */dev*:



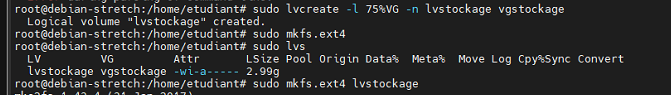
On crée les volumes :



On crée le volume logique en fonction du groupe de volume en lui attribuant dans notre cas 75% du Volume logique :



On liste les volumes logiques en fonction de leur groupe de volume :

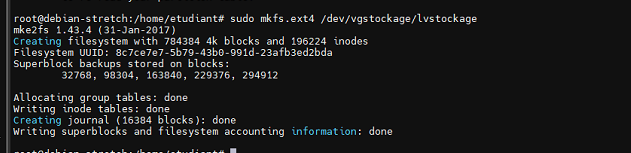


On repère le chemin et on teste les différents chemins pour formater :





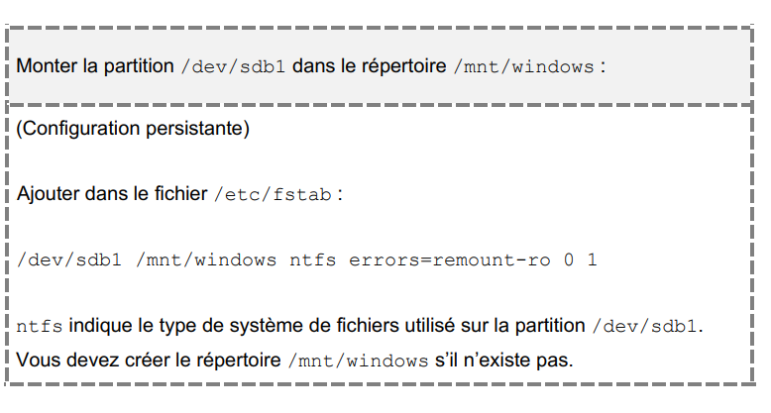
On formate en fonction du chemin */dev/vg\*/lv\** :



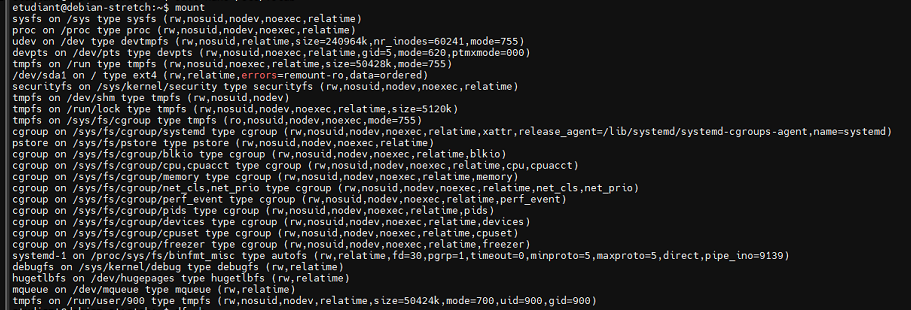
On a créé un point de montage et configuré *fstab* pour un montage permanent :



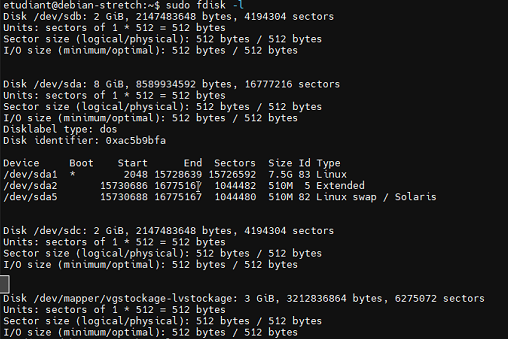
De cette manière :



On vérifie après redémarrage (ou du service *fstab* si cela fonctionne) avec la commande *mount* :

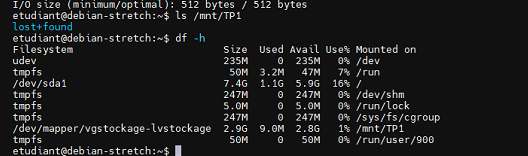


On liste les partitions avec *fdisk*



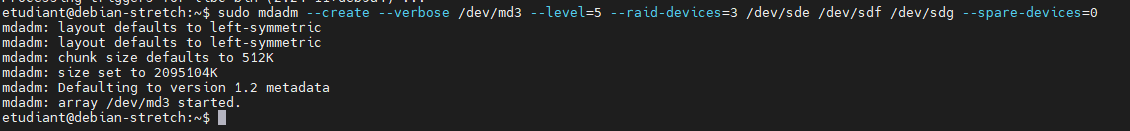
On liste le point de montage : Contient bien *lost+found*

On remarque la taille avec *df* :



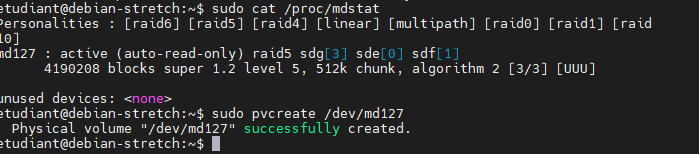
*Note : On a modifié notre chemin de montage /mnt/TP1 en tant que /mnt/windows suite à un changement de PC*

Après le montage des 3 disques SATA : On crée la partition */dev/md127* à partir de ces 3 disques

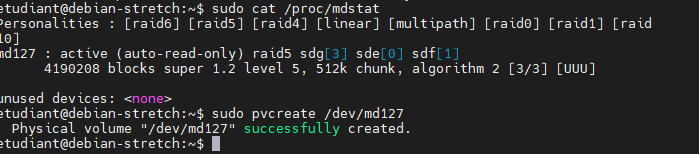


*Note : En fait on l’a nommé en tant que md127 :*

On liste les volumes RAID (md) actifs :



On initie ce volume RAID en tant que Physical Volume :



On liste pour vérifier la taille avant l’ajout dans vgstockage :

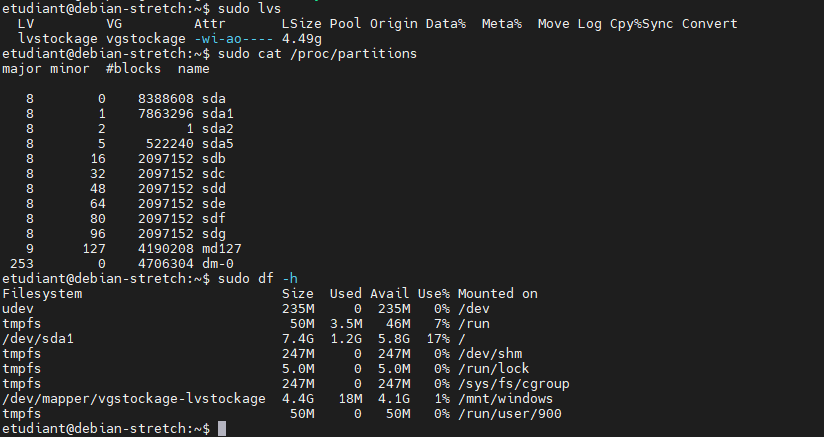


On se doit de mettre l’alias du vgstockage (=sans le chemin complet) et le chemin complet du volume RAID : On utilise *vgextend* : Pour scaler le volume :





On peut vérifier que ce dernier est dans le VG vgstockage : Mais on n’a pas augmenté sa capacité :



On étend la capacité de disque avec 4 Go de + plus pour le lv lvstockage :



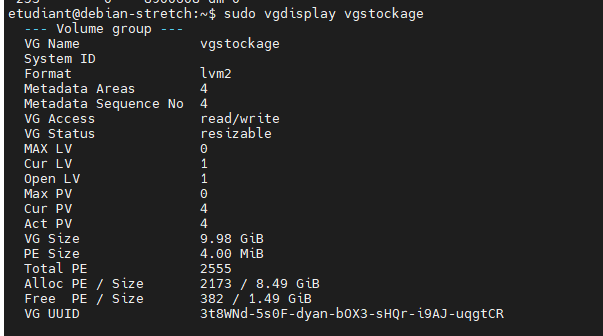
On remarque que le volume logique a été redimensionné : De 4.49Go à 8.49Go :

Mais que les partitions du système de fichiers ne détectent pas la modification :

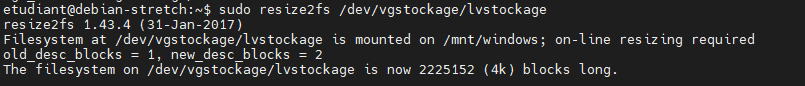


Par conséquent la taille du LV est supérieure (8.49GO) à la taille de la partition existant sur ce LV (4.49Go).

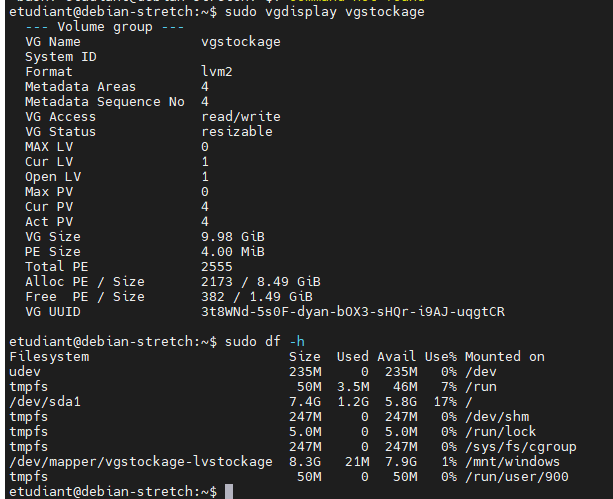
Et même le groupe de volume vgstockage affiche que la taille est de 8.49 et la taille maximale de 9.98



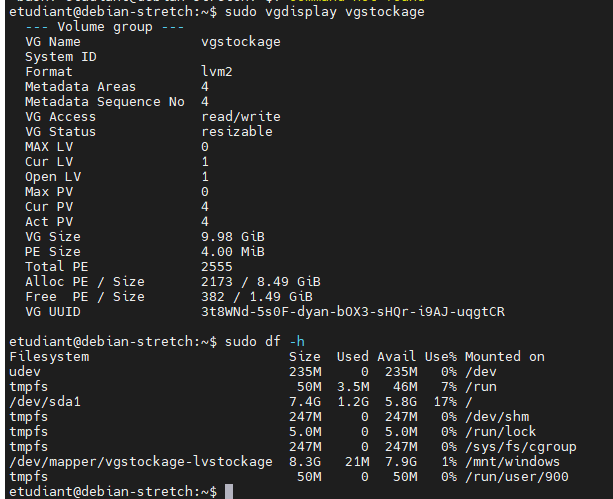
Pour remédier à cette situation, nous décidons d’étendre le système de fichiers avec *resize2fs* sur levolume logique:



On affiche à nouveau le vgstockage, on remarque que la taille est identique à la taille, avant d’étendre le système de fichiers :



Cependant comme on le prédisait désormais le système de fichiers détecte que la taille est de 8.49 Go (approximativement : 8.3Go)



**Synthèse 2 :**

Nous avons monté les disques IDE sur VirtualBox :

On a créé un VG nommé vgstockage. Notre VG est composé de PV : */dev/md2*.

Désormais on a créé un LV qui se nomme lvstockage avec 75% de vgstockage. Puis on a formaté ce LV en ext4 et on l’a monté de manière persistante.

Nous avons monté les 3 disques SATA :

On a ajouté un nouveau volume RAID pour voir la flexibilité de ce système.

On a initialisé ce volume en tant que PV, puis on a ajouté ce dernier dans vgstockage et enfin on élargit la capacité du LV avec 4 Go de plus.

La taille de la partition réelle est plus faible que celle du volume logique tant que le système de fichiers n’est pas étendu. Néanmoins une fois que le système est étendue, la capacité est augmentée.



Woo hoo ! Hasta la Vista ! On est génial n’est-ce pas !!

[**https://github.com/amineAUPEC/TP1-RAID-LVM\_LP.git**](https://github.com/amineAUPEC/TP1-RAID-LVM_LP.git)

**TP1 RAID LVM ECUE32 FIN :**