debian, système de fichiers, archive



**Raison de l'abandon**
J'ai plusiers fois rencontré des problèmes à l'occasion du remplacement d'un disque (pour finir par devoir recommencer à zéro et restaurer les données). Le système LVM me semble toutefois mature et efficace mais réservé à ceux qui passent leur vie sur Linux alors que ma vie professionnelle est orientée sur un autre système d'exploitation... Si l'on maîtrise LVM, même les groupes de volumes ou les volumes logiques peuvent être restaurés ce qui permet de limiter au maximum le risque d'y laisser des plumes .

# Késako ?

*Logical Volume Manager* (ou *LVM* pour les intimes) permet de regrouper des partitions ou *volumes physiques* (qui peuvent ou non être sur le même disque) dans un *volume logique* ou *groupe de volumes*.

# Mise en place

## Préparation du système



```
# aptitude install lvm2
```

Pffft... Purée c'est compliqué... 

## Préparation des disques

Le nouveau volume logique dont l'installation est décrite ici va utiliser 2 disques de 3To Western Digital Red « NAS Hard Drive » SATA III WD30EFRX utilisant des secteurs de 4Ko ce qui obligeait à une époque à aligner manuellement le premier secteur logique de chaque partition sur un secteur physique ce qui était loin d'être simple.

### Partitionnement

Sous Debian 7 « *Wheezy* », LVM n'a pas besoin que les partitions possèdent un système de fichier (c'était peut être possible avant mais c'était méconnu et je suis passé à côté). La seule contrainte dans ce cas étant que les disques ne peuvent être attribués à un volume logique qu'en entier et pas être partagés entre plusieurs volumes logiques. Comme c'est l'objectif ici pas de soucis.



L'avantage de ne pas attribuer de système de fichier aux partitions est de ne pas avoir à prendre le moindre risque quand au choix du type de partition (GPT indispensable pour les disques de plus de 2To alors que LVM requiert théoriquement des partitions au format éponyme dédié).

Donc il suffit d'écrire sur les disque une table de partition :

```
# fdisk /dev/sdb
```



Il faut ensuite procéder à la même opération sur le second disque qui va rejoindre le volume logique (*/dev/sdc* ici).

### Initialisation des volumes physiques

Pour que LVM puisse utiliser une partition, il faut l'initialiser en tant que volume physique LVM. Puisque qu'ici il n'y a qu'une partition par disque :

```
# pvcreate /dev/sdb1
```

Opération a répéter là aussi sur le second disque.

## Groupe de volumes

L'opération est très simple et consiste donc a créer un groupe de volumes en lui attribuant un nom (ici *vgdata*) et en y intégrant un volume physique précédemment initialisé :

```
# vgcreate vgdata /dev/sdb1
```



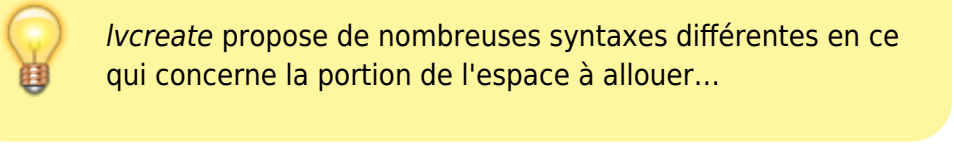
## Ajout d'espace

```
vgextend vgdata /dev/sdc1
```

## Volume logique

Il faut commencer par créer le volume logique (*lvdata*) et lui attribuer une partie plus ou moins importante de l'espace disponible dans le groupe de volumes (*vgdata*) :

```
lvcreate --name lvdata -l 100%FREE vgdata
```



*lvcreate* propose de nombreuses syntaxes différentes en ce qui concerne la portion de l'espace à allouer...

Puis il faut ensuite formater ce volume, par exemple :

```
mkfs.ext4 /dev/vgdata/lvdata
```

# En cas de corruption

## Symptômes

Dans mon cas, le premier symptôme a été un problème d'accès aux partages Samba après un reboot qui cachait tout simplement un problème d'accès au volume logique contenant les données.

J'ai évidement commencé par chercher du côté de Samba puis à force de chercher dans les logs, j'ai trouvé un paquets d'erreurs au niveau du système de fichiers de l'un des disques du volume logique (dans */var/log/kern.log*) :

L'étape suivante a été de tenter de forcer une vérification au démarrage avec les commandes suivantes :

```
# touch /forcefsck
# reboot
```

N'ayant pas d'écran sur cette machine j'ignore si un éventuel message a donné des indications sur le déroulement (ou plutôt le non-déroulement de l'opération) et la marche à suivre, mais, après le démarrage, le volume logique était à nouveau monté et les partages étaient à nouveau accessibles au bout de 3 ou 4 minutes (curieuse aussi cette latence mais je n'ai pas eu le temps sur le coup de

chercher l'explication et je ne l'ai pas trouvée par la suite). Avec le recul, le temps très court du redémarrage aurait dû me mettre la puce à l'oreille

## Forcer une vérification complète

Il faut d'abord retrouver le nom du (ou des) groupes de volumes (*VG*) et du (ou des) volumes logiques (*LV*) :

```
# lvm lvs
```

Ce qui va renvoyer quelque chose comme ceci :

Comme il va falloir démonter le volume en question, il est préférable d'arrêter les service correspondants (uniquement Samba dans mon cas) :

```
# /etc/init.d/samba stop
```

On peut ensuite démonter le volume et lancer la vérification complète (le paramètre de chacune des commandes dépends évidement de ce qu'a retourné la commande *lvm lvs* précédemment) :

```
umount /dev/vgdata/lvdata
e2fsck -y /dev/vgdata/lvdata
```

Et le retour très rapide a été le suivant :

Mouais... Mon œil!
Donc même si le commutateur *-y* est supposé tenter de détecter et réparer automatiquement toute corruption du système de fichier, il est manifestement insuffisant (et c'est probablement le même genre de vérification que la commande *touch /forcefsck* déclenche). Donc on insiste :

```
e2fsck -yfv /dev/vgdata/lvdata
```

Les commutateurs utilisés :

- y : est donc supposé tenter de détecter et réparer automatiquement toute corruption du système de fichier
- f : force une vérification approfondie même si le système semble propre
- v : force le mode verbeux afin d'obtenir un maximum d'infos en retour

On peut aussi ajouter le commutateur *c* pour forcer une vérification de chaque secteur mais c'est extrêmement long.

Et voici le résultat obtenu, cette fois après un long moment :



Le mode verbeux n'est pas si causant que çà, il y avait vraiment un paquet d'erreurs dans le log. Qu'a cela ne tienne...

```
reboot
```

Bingo ! Les partages Samba sont instantanément dispos et surtout, plus aucune erreur dans le log. 😎

Seule chose qui me chagrine : rien dans *lost+found* et à priori pas de données perdues si je compare aux sauvegardes (ou aux dossiers d'origine selon les cas). 😳 Voir ci-dessous pour la suite des aventures de ce disque (bon ça ne fait jamais plaisir mais au moins il y a une certaine logique 😕).

## Et paf ! le disque

Donc au premier reboot après la “réparation” tout semblait nickel, sauf que les erreurs sont revenues au reboot suivant 🤦.

Il faut donc vite passer au remplacement du disque tant qu'il marche encore sur 3 pattes...

Donc pour faire fonctionner le disque (même partiellement) donc j'ai réutilisé ces commandes:

```
# touch /forcefsck
# reboot
```

Quand on a plusieurs disques identiques et qu'il faut en identifier un, il faut récupérer son numéro de série :

```
# smartctl -i /dev/sdb
```





Pour avoir accès à la commande *smartctl*, il faudra que le paquet *smartmontools* soit installé.

Et il faut aussi connaître le nom du volume logique :

```
# lvm lvs
```



J'ai donc ensuite ajouté mon disque (à chaud dans mon cas) et vérifié qu'il était bien détecté par le système en jetant simplement un œil au contenu de */dev* (il faut évidement avoir une idée de son contenu ou avoir pensé à regarder avant ).

Il ne reste plus qu'à préparer le disque pour l'utilisation avec LVM, à l'ajouter au volume logique et y déplacer les données contenues sur le disque défectueux et voici le scénario :

```
# pvcreate /dev/sdd
# vgextend vgdata /dev/sdd
# pvmove /dev/sdb /dev/sdd
# vgreduce vgdata /dev/sdb
# pvremove /dev/sdb
```

Sauf que ça coince dès la commande *vgextend* :

Le but étant de récupérer ce qui est à récupérer sur le disque en question, il faut creuser la question en regardant l'état des volumes physiques :

```
# pvs
```

C'est l'attribut *m* (pour *missing* qui coince)...

Il est théoriquement possible de réinitialiser cet attribut avec cette commande :

```
# vgextend --restoremissing vgdata /dev/sdb1
```

Bon ben voilà... La messe est dite pour ce disque . Malgré mes recherches, je n'ai pas trouvé comment passer outre donc il est temps de retirer le disque du volume logique :

```
# vgreduce --removemissing vgdata
```

Pas le choix, il faut utiliser le commutatuer –*force* pour retirer ce fichu disque:

```
# vgreduce --removemissing vgdata --force
```

Comment ça « *Logical volume “lvdata” successfully removed* » ?!

Bon ben voilà, ça c'est fait, y'a plus qu'à recréer un volume logique tout neuf et restaurer les données. <blockquote>Braves gens ! Notre enchanteur m’informe, que d’habitude il y arrive

très bien mais que là, visiblement, ça marche pas ! <cite>Kaamelott (Livre II - L'Invincible)</cite>
</blockquote>



Ce fameux volume logique dont la fin funeste est décrite ci-dessus utilisait 2 disques de 3To chacun, or, pour utiliser une partition de plus de 2To, il faut une partition de type GPT qui n'est pas géré par l'outil intégré à Debian *fdisk*. Par ailleurs l'alignement des partitions sur ces disques utilisant des secteurs de 4Ko était très délicat et j'ai fait le choix de passer outre. Il est possible que la combinaison GPT/LVM ou le mauvais alignement des partition soit responsable de la suppression non désirée du volume logique... Espérons en tout cas...

From:
https://wiki.geekitude.fr/ - **Geekitude**

Permanent link:
**https://wiki.geekitude.fr/archives/info/os/debian/lvm?rev=1525338540**

Last update: **2018/05/03 11:09**