

LVM, une autre manière de partitionner

Introduction:

LVM (Logical Volume Manager, ou *gestionnaire de volumes logiques* en bon français) permet la création et la gestion de volume logique sous Linux. L'utilisation de volumes logiques remplace en quelque sorte le partitionnement des disques. C'est un système beaucoup plus souple, qui permet par exemple de diminuer la taille d'un système de fichier pour pouvoir en agrandir un autre, sans se préoccuper de leur emplacement sur le disque.

Avantages de LVM

- Il n'y a pas de limitations « étranges » comme avec les partitions (primaire, étendue, etc.).
- On ne se préoccupe plus de l'emplacement exact des données.
- On peut conserver quelques giga-octets de libres pour pouvoir les ajouter n'importe où et n'importe quand.
- Les opérations de redimensionnement deviennent quasiment sans risques, contrairement au redimensionnement des partitions.

Inconvénients de LVM

- Aucun, pour ceux qui aiment utiliser la ligne de commande

Depuis Intrepid, il existe le paquet `system-config-lvm` dans les dépôts Universe Ubuntu, qui propose une interface graphique. Je déconseille vivement son utilisation. Le seul usage qui peut en être fait sans risques, est la simple consultation. Concrètement cela vous permettra de visualiser la configuration (contenue dans les fichiers textes `/etc/lvm/backup/*`) d'une façon plus graphique. Mais pour ce qui est du paramétrage et de la mise en place de la configuration, surtout tenez vous en aux lignes de commandes.

But de ce document:

Découvrir le principe de LVM, créer un ou deux volumes logiques **sans modifier son système existant**.

La « conversion » de son système pour le migrer entièrement vers LVM ne fait donc pas partie de ce document pour l'instant, mais au moins, vous aurez les bases pour le faire.

Pré-requis:

- Un disque ou une partition libre.
- Ne pas être allergique à la ligne de commande.
- Avoir un peu de temps et l'esprit grand ouvert.

Installation:

Installer le paquet `lvm2`. Ok j'avoue, c'était la partie la plus facile. Maintenant, avant d'utiliser réellement LVM, il faut comprendre un peu de vocabulaire. Ne vous inquiétez pas si tout n'est pas encore très clair, ça viendra au fur et à mesure, en passant à la pratique.

Notions et vocabulaire:

Volume physique

Un volume physique ou « PV » pour « *physical volume* » est tout simplement un disque ou une partition. Bref, c'est un espace de stockage bien réel (autrement dit un périphérique de la forme /dev/sda2 par exemple), que l'on va confier à LVM. Bien évidemment, tout ce qui était présent sur la partition sera effacé.

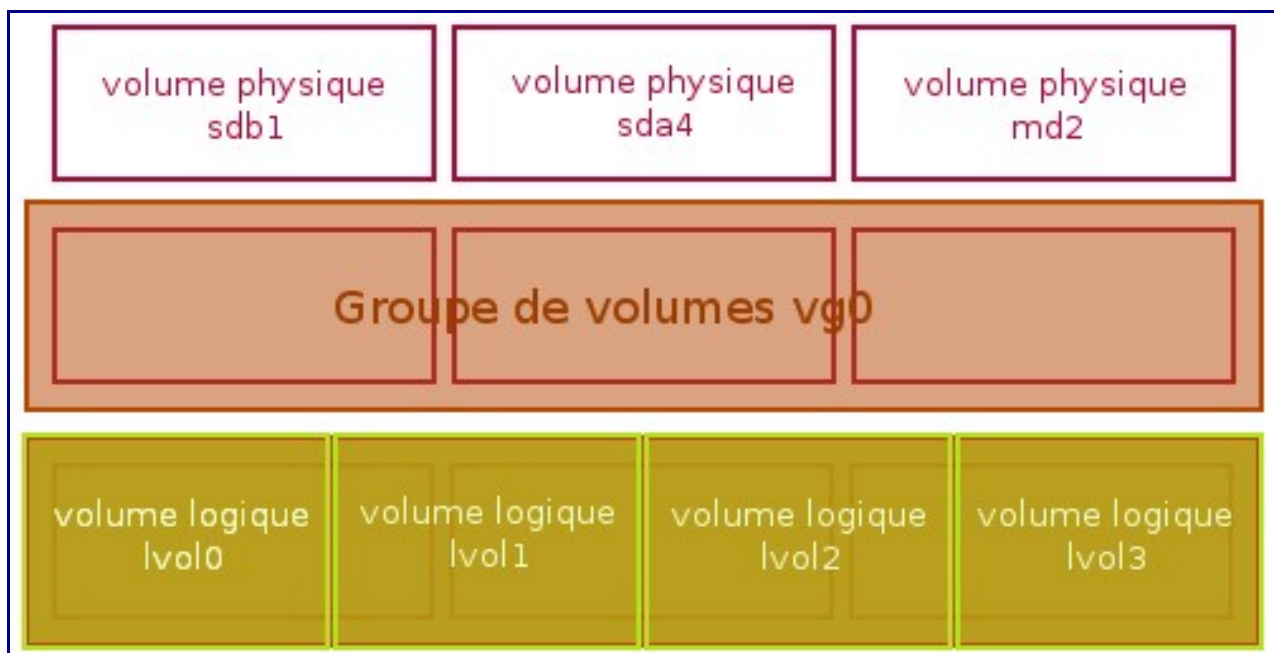
Groupe de volume

Un groupe de volumes ou « VG » pour « *volume group* » est, comme son nom l'indique, un ensemble de volumes physiques. On a donc un ou plusieurs volumes physiques dans un groupe de volumes, et pour utiliser LVM, il faut obligatoirement au moins un groupe de volumes. Habituellement, sur les gros serveurs, on essaye de regrouper les disques en fonction de leur caractéristiques (capacités, performances, etc.). Pour un particulier, le fait de mettre plusieurs disques dans un même groupe de volume peut permettre « d'étaler » un système de fichiers sur plusieurs disques, et d'avoir donc /home par exemple qui utiliserait 2 disques. Une telle configuration est tout de même assez dangereuse en cas de perte d'un disque... De plus, cela n'apporterait aucun gain de performance contrairement à du RAID-0 par exemple. :)

Volume logique

Un volume logique ou « LV » pour « *logical volume* » est ce que nous allons utiliser au final. Un volume logique est un espace « quelque part dans un groupe de volume » où l'on peut mettre un système de fichiers. C'est donc ce qui remplace les partitions. On peut donc utiliser un volume logique pour mettre la mémoire virtuelle, un pour /home, etc. Peut-on mettre / sur un volume logique ? Oui et non. Techniquement, oui, ça fonctionne maintenant à peu près correctement. Mais Grub peut se montrer capricieux dans certaines configurations. Donc, à moins d'utiliser Lilo, je vous conseille vivement d'éviter les ennuis inutiles pour commencer et de ne pas tenter tout de suite d'installer votre système avec un / sur un volume logique. Plus tard peut-être ?

Synthèse graphique > On peut voir les choses ainsi :



Création:

Bien, passons à la pratique !

Puisque nous allons passer toutes les commandes en tant que *root*, je vous conseille de passer *root* « une bonne fois pour toutes » :

```
sudo -i
```

Voilà, maintenant que votre *prompt* est un « # ». Cela m'évitera (et vous évitera) d'avoir à mettre *sudo* devant chaque commande.

Bonne nouvelle, les commandes LVM sont extrêmement simples à utiliser et elles intègrent toutes une aide en ligne très bien conçue, claire, courte, mais suffisante. De plus, leur nom se « devine » assez facilement :

- toutes les commandes agissant sur les volumes physiques commencent par *pv* ;
- toutes les commandes agissant sur les groupes de volumes commencent par *vg* ;
- toutes les commandes agissant sur les volumes logiques commencent par *lv*.

La première chose à faire est de créer **un disque physique**, en donnant une partition à LVM.

Essayez la commande suivante pour connaître la liste des commandes disponibles pour les volumes physiques :

```
man -k pv
```

Parmi toutes les commandes renvoyées, on remarque « *pvcreate* », ça doit être celle-là... Cette astuce fonctionne avec toutes les commandes LVM et permet de les retrouver facilement.

Disque physique

J'utilise /dev/sdc1 pour l'exemple, adaptez à votre cas ! Toutes les données de cette partition seront effacées !

```
pvcreate /dev/sdc1
```

Si tout s'est bien passé, aucun message d'erreur n'apparaît.

Si vous obtenez le message « No program "pvcreate" found for your current version of LVM. », c'est que le service LVM n'est pas actif. Lancez-le avec la commande suivante :

```
/etc/init.d/lvm start
```

Si votre distribution est récente, il est probable que cela ne soit pas nécessaire. La prise en compte de LVM et l'activation des volumes groupes déjà présent est automatique lors de l'installation du paquet.

Je veux utiliser la totalité d'un disque, *sdc* par exemple, puis-je faire *pvcreate /dev/sdc* (sans numéro de partition) ?

Oui et non... Techniquement, oui, pas de problème. Seulement, le fait que votre disque ne soit même plus partitionné pourrait vous poser problème un jour ou l'autre. Je vous conseille plutôt de créer une seule partition *sdc1* (primaire) utilisant tout le disque dur, sans la formater. Ensuite, lancez la commande en utilisant cette partition (*pvcreate /dev/sdc1*).

On peut vérifier le résultat avec la commande *pvdisplay*.

Groupe de volume

Il existe de nombreuses options lors de la création d'un groupe de volume... Mais continuons de faire au plus simple. Le seul paramètre indispensable sera de lui donner un nom, nous utiliserons les valeurs par défaut pour tout le reste. Pour une raison que j'expliquerai par la suite, donnons-lui un nom très court (2 ou 3 caractères). Par exemple : « mvg » pour « mon vg ».

Pour connaître la syntaxe de la commande `vgcreate` (comme pour toutes les autres commandes LVM), tapez simplement son nom :

```
vgcreate
```

La syntaxe est donc :

```
vgcreate VolumeGroupName PhysicalVolume [optionnellement d'autres  
PhysicalVolume]
```

Allons-y :

```
vgcreate mvg /dev/sdc1
```

Si tout se passe bien, vous avez maintenant un groupe de volume, contenant un disque physique. Vous pouvez obtenir d'autres informations sur ce groupe de volume en tapant `vgdisplay`:

```
vgdisplay
--- Volume group ---
VG Name                mvg
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   3
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                2
Open LV                0
Max PV                 0
Cur PV                1
Act PV                 1
VG Size                186,31 GiB
PE Size                4,00 MiB
Total PE               47695
Alloc PE / Size        15360 / 60,00 GiB
Free PE / Size         32335 / 126,31 GiB
VG UUID                BaTuai-1I8o-3rkY-Ut1r-ybta-mJnl-9X0oNZ
```

Volume logique

Nous y voilà... Cette fois-ci, nous allons vraiment créer deux espaces que l'on pourra ensuite « formater » en `ext4` par exemple.

Comme précédemment, le plus simple est de commencer par jeter un œil sur la syntaxe de la commande :

```
lvcreate
```

Les deux options vraiment importantes sont « -n » pour son nom, et « -L » pour sa taille. Le paramètre principal est « OriginalLogicalVolume ». Il s'agit peut-être d'une erreur dans le manuel (*man*). En fait, ce qu'il faut indiquer, c'est bien le groupe de volume dans lequel nous allons créer le volume logique. Pour l'exemple présent, je fais ici deux volumes, 10 Gio et 50 Gio :

```
lvcreate -n Vol1 -L 10g mvg
lvcreate -n Vol2 -L 50g mvg
```

Idem, on peut vérifier avec la commande *lvdisplay*:

```
~# lvdisplay
--- Logical volume ---
LV Name                /dev/mvg/Vol1
VG Name                mvg
LV UUID                q0D6cQ-mcMP-q8sf-XTlI-DdxX-QHd1-qkaB5J
LV Write Access        read/write
LV Status               available
# open                  0
LV Size                 10,00 GiB
Current LE              2560
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to    256
Block device            252:0

--- Logical volume ---
LV Name                /dev/mvg/Vol2
VG Name                mvg
LV UUID                JZjMxI-cTAw-cbs6-02BM-4Mev-P2E7-b8JX0x
LV Write Access        read/write
LV Status               available
# open                  0
LV Size                 50,00 GiB
Current LE              12800
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to    256
Block device            252:1
```

Systeme de fichiers

Avec les partitions, on avait des noms ressemblant à */dev/sda3*, etc. Avec LVM, on utilise aussi des périphériques dans */dev*, mais le chemin est de la forme */dev/nom_du_vg/nom_du_lv*. Autrement dit, puisqu'on a décidé d'appeler nos volumes logiques "Vol1" et "Vol2", les noms de ces périphériques de ce volume logique sont */dev/mvg/Vol1* et */dev/mvg/Vol2*. À partir de maintenant, */dev/mvg/Volx* peut être utilisé dans toutes les situations et avec toutes les commandes qui attendent quelque chose de la forme */dev/...* Par exemple :

```
mkfs -t ext4 /dev/mvg/Vol1
mkfs -t ext4 /dev/mvg/Vol2
mkdir /Essai
mount /dev/mvg/Vol1 /Essai
df -h
```

Et normalement, */dev/mvg/Vol1* devrait être monté sur */Essai*. Regardez bien la ligne correspondante. Si on avait choisi un nom de VG ou de LV plus long, la sortie de *df* aurait été modifiée, car le nom aurait « touché » les valeurs... On aurait été obligé de passer des lignes et l'affichage aurait été plus difficile à lire. Techniquement, choisir des noms « longs » pour les VG et les LV ne pose aucun problème, mais c'est l'affichage qui sera parfois délicat. Pour cette raison uniquement, je préconise de se limiter à 7 caractères au total (donc par exemple 3 pour le vg et 4 pour le lv, ou 2 et 5, comme vous voulez).

Pourquoi est-il écrit `/dev/mapper/mvg-Vol1` et non `/dev/mvg/Vol1` ?

Avec LVM en version 1, c'est bien `/dev/mvg/Vol1` qui aurait été affiché. Depuis la version 2, LVM utilise le périphérique *mapper*, ce qui permet pas mal de choses (comme chiffrer les volumes logiques, etc.). Pour simplifier, disons que ces deux notations « `/dev/mvg/Vol1` » et « `/dev/mapper/mvg-Vol1` » sont synonymes. Dans la pratique, il est conseillé quand même d'utiliser plutôt la forme « `/dev/mvg/Vol1` », certaines commandes ne passeront pas autrement.

Suppression:

Rien de plus simple :

```
umount /Essai1 # si le volume Vol1 est monté en /Essai1
lvremove /dev/mvg/Vol1
```

⚠ Attention, une fois un volume logique effacé, il est **totalemment impossible** de récupérer les données qu'il contenait.

Redimensionnement:

Volume physique

Imaginons maintenant que notre groupe de volume (mvg) n'ait plus suffisamment d'espace libre. On souhaite donc lui rajouter un volume physique afin de rajouter de l'espace libre. Ça tombe bien, on dispose d'un volume physique `sd2` que l'on va pouvoir ajouter à `mvg` :

On initialise le volume en vue de son utilisation dans LVM :

```
pvcreate /dev/sdc2
```

On rajoute le volume `sd2` au groupe de volume `mvg` :

```
vgextend mvg /dev/sdc2
```

Volume logique

Il est très facile d'augmenter ou de diminuer la taille d'un volume logique. Mais attention, la taille d'un LV n'a pas de lien direct avec la taille de ce qu'il contient (*swap* ou système de fichier). Le LV est une boîte, le système de fichier est le contenu de la boîte. Augmenter la taille de la boîte sans augmenter la taille du contenu ne pose pas de problème, mais l'inverse...

Agrandissement

Bien qu'il soit évidemment moins risqué d'agrandir ou de diminuer la taille d'un système de fichier après l'avoir démonté, la plupart des formats (`ext3`, `reiserfs`, `ext4`...) supportent désormais cette modification "à chaud" (avec un des données qui restent donc accessibles en lecture/écriture durant toute l'opération).

Pour ne courir aucun risques, il faut dans l'ordre : Démontez le système de fichier, augmentez la taille du volume logique (la "boîte"), vérifiez que tout est OK, puis augmentez la taille du système de fichier. Il ne reste plus qu'à le remonter.

Voilà donc un exemple :

```
umount /Essai2          # démontage du volume
e2fsck -f /dev/mvg/Vol2  # vérification du volume
lvresize -L 55g /dev/mvg/Vol2 # redimensionnement du volume
resize2fs /dev/mvg/Vol2  # redimensionnement du système de fichier
mount /dev/mvg/Vol2 /Essai2
```

Attention : le paramètre '55g' passé à la commande *lvresize* correspond à la taille totale finale du lv et doit donc être supérieur à la taille originale du lv !

Notez que :

- Si jamais vous indiquez une taille (ici 55 Gio) inférieure à celle de départ, vous **détruirez** le système de fichier (toute la partie qui ne « rentrera plus dans la boîte dont vous venez de diminuer la taille ». Donc, indiquez bien une taille supérieure à celle qu'elle avait. (Vous pouvez aussi utiliser la notion « +ng » par exemple indiquer +10g ou +100m ou...)
- J'utilise *resize2fs* car le système de fichier est en ext4 (ou 3), il faudrait utiliser un autre programme pour reiserfs ou autre.
- Si vous obtenez un message d'erreur vous demandant de refaire un *fsck* (ou un *e2fsck*) alors que vous l'avez fait et qu'il n'y a pas d'erreur, il est possible de forcer le redimensionnement (option « -f »). N'utilisez pas cette option dans d'autres circonstances...
- Tous les systèmes de fichiers ne sont pas extensibles de cette façon !

Globalement, la modification à chaud (surtout l'agrandissement !) ne pose vraiment plus de soucis, et les opérations précédentes peuvent donc être résumés en :

```
lvresize -L 55g /dev/mvg/Vol2 # redimensionnement du volume
resize2fs /dev/mvg/Vol2      # redimensionnement du système de fichier
```

Rétrécissement

Diminuer la taille d'un système de fichier est un peu plus délicat. En effet, si jamais on commet l'erreur de diminuer la taille de la "boîte" (le volume logique) avant de réduire la taille du contenu (le système de fichier lui même) alors on détruit son système de fichier... pareil si on réduit trop la taille du volume logique.

Pour éviter tout risque, je conseille d'utiliser la méthode suivante (un peu plus longue que la normale, mais beaucoup plus fiable) :

- Réduire la taille du système de fichier plus que nécessaire
- Réduire la taille du volume logique pour lui donner exactement la nouvelle taille souhaitée.
- Agrandir le système de fichier pour qu'il occupe tout l'espace disponible.

De cette façon, le risque d'erreur est beaucoup plus faible.

Attention, tous les systèmes de fichiers ne peuvent pas être "réduits". Pour ext3 et reiserfs, cela se fait très bien. Voilà un exemple avec reiserfs...

```
df -h | grep ca
/dev/mapper/svg-ca  512M  230M  283M  45% /home/ca
```

Dans cet exemple, le volume "ca", se trouve dans le groupe de volume svg. Sur ce volume logique existe un système de fichier en reiserfs d'une taille de 512 Mo. Or, je n'utilise que 230 Mo. De plus, je sais que je n'ajouterai jamais rien dans ce volume. Je désire donc diminuer sa taille à 256 Mo (pour se laisser une marge de sécurité, et parce que ça fait un compte rond ;) Je commence par démonter le système de fichier :

```
umount /home/ca
```

Ensuite je vais donc diminuer la taille du système de fichier, plus que nécessaire. Plutôt que de retirer 256 Mo, je vais donc en retirer 258. Je peux le faire, car il reste 283 Mo de libre... Evidement, retirer plus de place qu'il n'en reste serait suicidaire...

```
e2fsck -f /dev/mvg/toto  
resize_reiserfs -s -258M /dev/svg/ca
```

ATTENTION : Si vous êtes en ext3, on ne peut pas indiquer la quantité d'espace à enlever, il faut donner la taille finale voulue (512-258). la bonne commande aurait été :

```
e2fsck -f /dev/mvg/toto  
resize2fs -p /dev/svg/ca 254M
```

Remarque : On notera l'utilisation de e2fsck pour vérifier l'intégrité de sa partition et la continuité des données.

Maintenant que le système de fichier à diminué, il faut donner au volume logique sa nouvelle taille, 256 Mo au lieu de 512 (**ATTENTION** : le paramètre "-256M" de la commande lvresize signifie que l'on enlève 256 Mo au volume logique et non pas que l'on fixe sa taille à 256 Mo) :

```
lvresize -L -256M /dev/svg/ca  
WARNING: Reducing active logical volume to 256.00 MB  
THIS MAY DESTROY YOUR DATA (filesystem etc.)  
Do you really want to reduce ca? [y/n]: y  
Reducing logical volume ca to 256.00 MB
```

Plus qu'une dernière étape, on indique au système de fichier qu'il peut s'étendre automatiquement pour prendre tout l'espace disponible. Il devrait donc pouvoir grandir de 2 Mo. Il trouvera tout seul la taille exacte en nombre de blocs etc... On n'a pas pris le risque de faire une erreur en le réduisant "pile poil" de la même taille que le volume logique, car la moindre erreur aurait pu corrompre le système de fichier à quelques blocs près.

```
resize_reiserfs /dev/svg/ca
```

ou, si vous utilisez ext3 :

```
resize2fs /dev/svg/ca
```

Il ne reste plus qu'à remonter le système de fichier :

```
mount /dev/svg/ca /home/ca  
df -h | grep ca  
/dev/mapper/svg-ca 256M 230M 27M 90% /home/ca
```

C'est gagné... Le système de fichier fait maintenant 256 Mo, et nous avons toujours nos 230 Mo de données à l'intérieur. Conclusion : Jouer avec la taille des volumes logiques fonctionne très bien, il faut juste prendre son temps et ne pas faire n'importe quoi :)

Mieux comprendre ou aller plus loin:

Notion d'« extend »

Un *extend*, ou « *physical extend* » aussi appelé « PE », est un tout petit morceau d'un groupe de volumes. En fait, au moment de la création d'un groupe de volumes, le ou les disques sont découpés en morceaux de quelques Mio (4 Mio par défaut). Lorsqu'on crée un volume logique, LVM va utiliser autant de PE que nécessaires. La taille d'un volume logique sera donc toujours un multiple de la taille d'un PE.

Glossaire

abréviation	anglais	français	description
VG	V olume G roupe	G roupe de V olumes	une "partition" dans un groupe de volume
LV	L ogical V olume	V olume L ogique	
PV	P hysical V olume	V olume P hysique	
PE	P hysical E xtend		un tout petit morceau d'un groupe de volumes

LVM et RAID

Il est tout à fait possible d'utiliser LVM sur un volume en RAID logiciel. Une fois que le RAID a été créé (`/dev/md0` par exemple), il suffit de le donner à LVM, avec la commande habituelle :

```
pvcreate /dev/md0
```

Bien qu'il soit possible de partitionner le raid `/dev/md0` comme n'importe quel disque ordinaire (ce qui permet d'obtenir des devices de la forme `/dev/md0p1`, `/dev/md0p2` etc), je vous le déconseille vivement. En effet le but est bien d'utiliser lvm pour découper l'espace, et plus l'ancienne notion de partition physique. De plus, l'utilisation de ces partitions nécessiterait des changements dans la configuration de lvm (filter dans `/etc/lvm/lvm.conf`). Bref, C'est se donner du mal pour pas grand chose.

Donc comme le montre la commande plus haut, le mieux est de donner l'intégralité du raid.