**LVM - Les bases**

**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 ne sont pas allergiques à la ligne de commande :)

Il existe le paquet system-config-lvm dans les dépôts Universe, 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 fonctionnement de LVM, les avantages qu'il offre et les outils qui lui sont associés.

Note : Il n'est pas possible de changer de mode de fonctionnement (avec ou sans lvm) simplement. Cela nécessite une recopie des fichiers. Autrement dit, concernant la partie système (/, swap...) il n'est pas évident de "convertir" une installation (Et cela n'apporterait pas grand chose). LVM est donc surtout utilisé pour la partie "données".

**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**

Installez 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 très clair lors de la lecture, ça viendra au fur et à mesure, en passant à la pratique.

**Notions et vocabulaire**

**Volume physique**

Un **volume physique** ou « PV » (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 volumes**

Un **groupe de volumes** ou « VG » (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. En entreprise ce type de fonctionnalité peut être utilisée pour étendre temporairement un espace de stockage. 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 dangereuse !** (deux disques donc deux fois plus de risques de pannes). De plus, elle n’apporte aucun gain de performance contrairement à du RAID-0 (qui lui aussi permet de cumuler la volumétrie de plusieurs espaces de stockage).

D'une façon générale, je recommande donc de ne mettre qu'un seul PV dans un groupe de volumes, sauf si il s'agit de fichiers sans importance, et que seuls plusieurs "petits" disques de capacités différentes sont disponibles pour constituer l'espace de stockage.

**Volume logique**

Un **volume logique** ou « LV » (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 (swap), un pour *home*, etc. Peut-on mettre */* sur un volume logique ? Oui. Mais a mon avis c'est vraiment se compliquer la vie pour pas grand chose. (sauf besoin précis, genre snapshot du / mais nous n'en sommes pas la ! :)

**Synthèse graphique**

A venir... je cherche un schéma qui me plaise :)

**Création**

Passons à la pratique. Puisque nous allons passer toutes les commandes en tant que root, je vous conseille de prendre les droits administrateur une bonne fois pour toutes :

 sudo -i

Cela m’évitera (et vous évitera) d’avoir à mettre sudo devant chaque commande...

Bonne nouvelle, les commandes LVM sont très 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 connaitre 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. Notons aussi l'existence des commandes : pvs, vgs, et lvs. Ces commandes permettent de lister les objets correspondants.

**Disque physique**

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

 pvcreate /dev/sdc1

Cette commande "formate" /dev/sdc1 en LVM (avec de très gros guillemets).

"Je veux utiliser la totalité d’un disque, /dev/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. Cela permettra également de modifier le type de la partition pour indiquer qu'elle sera utilisé pour LVM (type 8e dans fdisk).

**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 Physical...](http://hoper.dnsalias.net/tdc/optionnellement%20d%E2%80%99autres%20PhysicalVolume)

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". Essayez aussi la commande "pvs", qui liste les volumes physiques, ainsi que le groupe de volume auquel ils appartiennent.

**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. 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, je fais ici deux volumes, 10 Gio et 50 Gio :

 lvcreate -n vol1 -L 10g mvg

 lvcreate -n Vol2 -L 50g mvg

Lancez les deux commandes : "lvdisplay", et "lvs", pour voir le résultat.

**Système 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 des périphériques de ces volume logique sont "/dev/mvg/vol1" et "/dev/mvg/vol2".

À partir de maintenant, la syntaxe */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 /Essai1

 mount /dev/mvg/vol1 /Essai1

 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 device 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

 lvremove /dev/mvg/vol1

**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 sdc2 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 sdc2 au groupe de volume mvg :

 vgextend mvg /dev/sdc2

**ATTENTION** : Je répète encore qu'avoir des systèmes de fichiers répartis sur plusieurs disques est risqué ! Ne le faite que si vous n'avez pas d'autre possibilités. (Et que toutes vos données vraiment critiques sont sauvegardés).

**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 boite, le système de fichier est le contenu de la boite. Augmenter la taille de la boîte sans augmenter la taille du contenu ne pose pas de problème, mais l’inverse…

**Agrandissement**

La plupart des formats (ext3, reisersfs, XFS, ext4...) supportent cette modification "à chaud" (avec des données qui restent donc accessibles en lecture/écriture durant toute l'opération. Mais si vous voulez vraiment ne courir aucun risques, il faut dans l'ordre : Démonter le système de fichier, augmenter la taille du volume logique (la "boite"), puis augmenter la taille du système de fichier. Il ne reste plus qu'à le remonter.

Voila 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

**TRES IMPORTANT** : le paramètre '55g' passé à la commande lvresize correspond à la taille totale finale du lv et doit donc être supérieur à sa taille de départ !

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 boite dont vous venez de diminuer la taille). Donc, indiquez bien une taille supérieure à celle qu'elle avait. Il est moins risqué d'utiliser une autre syntaxe, la notion « *+ng* » par exemple indiquer : -L +10g ou +100m ou...

J'utilise resize2fs car le système de fichier est en extX (ou 2, 3 ou 4), 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, l'agrandissement à chaud d'un système de fichier ne pose plus vraiment de soucis, et les opérations précédentes peuvent donc êtres 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 plus délicat. En effet, si jamais on commet l'erreur de diminuer la taille de la "boite" (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) :

1. Réduire la taille du système de fichier plus que nécessaire
2. Réduire la taille du volume logique pour lui donner exactement la nouvelle taille souhaitée.
3. 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 reiserfs, ext3 et ext4, cela se fait très bien. Voila 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... Évidement, retirer plus de place qu'il n'en reste serait suicidaire...

 e2fsck -f /dev/svg/ca

 resize\_reiserfs -s -258M /dev/svg/ca

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). les bonnes commandes auraient été :

 e2fsck -f /dev/svg/ca

 resize2fs -p /dev/svg/ca 254M

Maintenant que le système de fichier à diminué, il faut donner au volume logique sa nouvelle taille, 256 Mo au lieu de 512

 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](http://hoper.dnsalias.net/tdc/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**

J'explique le fonctionnement de LVM (et ce qu'il est possible de paramétrer) [ici](http://hoper.dnsalias.net/tdc/index.php?pages/LVM-Fonctionement-interne). J'explique aussi comment utiliser les "snapshost" sur [cette page](http://hoper.dnsalias.net/tdc/index.php?pages/LVM-Les-snapshots).

Publié le jeudi 4 novembre 2010 par hoper