

## Sommaire

---

- Le LVM (Logical Volume Manager)
- De l'utilité du LVM
- Les composants du LVM
- Utiliser le LVM : ce dont il faut disposer
  - Kernel et utilitaires
  - Sur quels périphériques et systèmes de fichiers puis-je faire du LVM ?
- Configuration du LVM
  - L'arborescence du LVM
  - Création d'un volume physique
  - Création d'un groupe de volume
  - Création d'un volume logique
- Recueillir des informations sur le LVM
- Commandes complémentaires
  - Redimensionner un groupe de volumes
  - Redimensionner un volume logique
  - Une fonction particulière du LVM : la réalisation de snapshots
  - Autres commandes
- Utilisation pratique du LVM dans la gestion de l'espace
- Evolution du LVM : LVM 2
  - Le device-mapper
  - Utilisation d'un arbre binaire
  - Une plus grande configurabilité
  - Compatibilité LVM1 / LVM2
  - Autre
- En conclusion...
- Liens utiles
- Copyright

## Le LVM (Logical Volume Manager) >>

---

Par **Anne**

Un des aspects cruciaux dans l'administration d'un serveur ou d'une machine de bureau est la gestion de l'espace disque. Quoi de plus énervant que de voir l'installation d'une application échouer par manque d'espace, ou un serveur rendu indisponible parce que le système de fichiers /var était plein du fait des fichiers de log ?

Un outil apporte une solution satisfaisante et efficace : le LVM (Logical Volume Manager).

## De l'utilité du LVM >>

---

Le LVM ou Logical Volume Manager, est une technique créée à la base par IBM consistant à fournir la possibilité de modifier la taille des partitions sur les disques durs sans avoir besoin de tout reformater, voire de créer des partitions s'étalant sur plusieurs disques. L'objectif est ainsi d'éviter arrêt et redémarrage d'une machine en production. Cette technique est disponible sur linux depuis la version 2.4 du noyau (très exactement 2.3.47).

Dans un partitionnement de type classique, à l'aide des commandes fdisk, vous ne pouvez avoir que 4 partitions primaires pour chaque disque (en IDE) ou éventuellement 3 partitions primaires et une partition étendue qui contiendra des partitions logiques. L'inconvénient de ce type de partitionnement est que lorsque vous souhaitez réduire la taille d'une partition ou l'augmenter, vous devez notamment disposer d'outils spécifiques comme GNU-parted. De plus, le partitionnement ne se fera que disque par disque. Imaginez alors que vous souhaitiez ajouter un deuxième disque sur votre machine ou agrandir la taille de votre système de fichiers /home...

Vous ne pourrez pas profiter de l'espace disponible sur ce deuxième disque pour agrandir /home, à moins d'y accrocher un nouveau système de fichiers. Agacant non ?

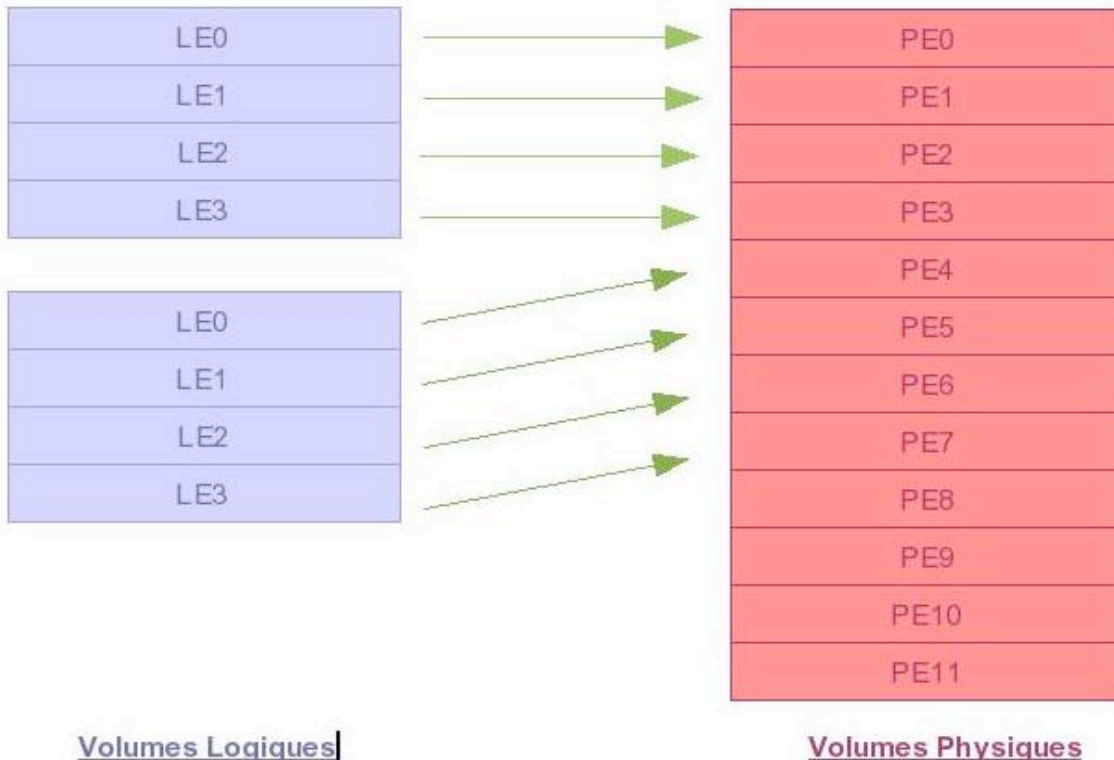
Eh bien, LVM est là pour vous simplifier la vie.

## Les composants du LVM >>

Le principe de fonctionnement de LVM est relativement simple. Il s'agit en fait pour effectuer ce type de partitionnement, de s'affranchir complètement des limites physiques du ou des disques disponibles. Les étapes ci-dessous en décrivent le fonctionnement. Le LVM s'accompagne aussi d'un certain nombre de termes techniques énoncés également ci-dessous.

- Chaque disque dur ou partition va être transformé en **volume physique**. Cette opération consiste à découper le disque en tranches, appelées Physical Extents (PE). Par défaut (et convention), 1 PE = 4Mo.
- Chaque volume physique va être inséré dans un **groupe de volumes**. Celui-ci peut contenir un ou plusieurs volumes physiques (donc disques ou partitions). Un groupe de volume est un espace logique découpé en Logical Extents (LE) de même tailles que les physical extents, soit 4 Mo par défaut.

Le système va ensuite établir des **pointeurs** entre un physical extent et un logical extent comme indiqué sur le schéma ci-dessous. (schéma 1) **Schéma 1 : système des pointeurs LVM**

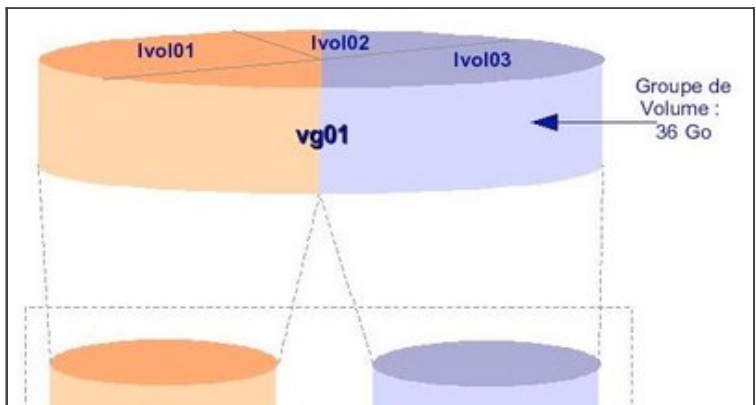


- La dernière étape va consister à découper le groupe de volumes en partitions appelées **volumes logiques** dans lesquelles nous pourrions au choix, créer un système de fichier, une partition de swap. Ces partitions pourront être redimensionnées et/ou déplacées.

Voici de manière schématique à quoi pourrait ressembler votre(vos) disque(s) après ce traitement (Schéma 2).

### Schéma 2 : impact de LVM sur les disques durs

Dans ce cas de figure, les deux disques hda et hdb ont été transformés en volumes physiques. Puis ils ont été insérés tous les deux dans un groupe de volumes appelé vg01. A partir de ce moment-là, il n'est plus nécessaire de tenir compte des limites physiques des disques, en effet, nous n'avons plus qu'un espace, et un seul, de 36 Go (regroupant donc nos deux unités de disques hda et hdb). Le groupe de volumes a ensuite été découpé en 3 volumes logiques, nommées lvol01, lvol02, lvol03. On remarquera que lvol02 est composé de



logical extents pointant sur des physical extents appartenant à hda et hdb. La dernière étape consistera à créer un système de fichiers dans chacun de ces volumes logiques.

[Schéma d'un groupe de volumes sur deux disques](#)



#### Remarques :

- la taille que l'on attribuera à un volume logique s'exprime en nombre de logical extents. Ceux-ci sont indivisibles. Si sa taille est de 4 Mo et que je souhaite un volume logique de 14 Mo, le système attribuera 4 logical extents au volume logique, soit 16 Mo (arrondi au nombre de LE supérieur)
- la taille d'un PE, et donc d'un LE, est personnalisable lors de la création d'un volume physique.
- les noms des groupes de volumes et volumes logiques sont personnalisables à leur création
- Attention : la création d'un volume physique écrase toutes les données existantes sur la partition et/ou le disque !!**
- L'utilisation du LVM pour partitionner un disque entraîne une perte d'espace liée à l'écriture des données nécessaires au système pour gérer le LVM (métadatas) :
  - la PVRA : Physical Volume Reserved Area. Comme son nom l'indique, elle contient les informations LVM spécifiques au volume physique.
  - la VGRA : Volume Group Reserve Area. Elle contient les informations liées au groupe de volumes mais aussi aux volumes logiques contenus dans le groupe de volumes
  - la BBRA : Bad Block Relocation Area : cette zone contient des informations liées au mécanisme de ré-allocation des blocs défectueux.

## Utiliser le LVM : ce dont il faut disposer >>

Pour être utilisé, le LVM nécessite de disposer du driver qui permet de générer la couche assurant le mapping (la carte) entre périphérique physique et vue logique, des utilitaires pour manipuler ce mapping et des périphériques physiques.

### Kernel et utilitaires

Le LVM est fourni lors de l'installation, sauf sur de très vieux systèmes (noyau 2.2). Il faut quand même savoir que l'implémentation du LVM se fait à deux niveaux :

- le kernel,
- les commandes nécessaires pour gérer les structures LVM.

Au niveau du kernel lors de l'installation ou de la recompilation du noyau, vous devez avoir intégré ou mis en module le driver LVM. Celui-ci se situe dans le menu de compilation "Multi-device support". On peut le vérifier de la manière suivante :

```
root@pingu# grep -i lvm /boot/config
# Multi-device support (RAID and LVM)
CONFIG_BLK_DEV_LVM=m
```

Vous devez également disposer des commandes nécessaires à l'administration du LVM. Elles sont incluses dans le package lvm. Pour vérifier qu'il est installé (dans le cas d'une Mandriva) :

```
root@pingu# rpm -qa|grep lvm
lvm-1.0.3-9
```

Toutes les commandes passées en revue dans la suite de cet article font partie de ce paquetage.

## Sur quels périphériques et systèmes de fichiers puis-je faire du LVM ?

Pour pouvoir utiliser le LVM, vous devez disposer soit d'un disque vierge (rappel : la création d'un volume physique entrainera la perte de données existantes) et/ou d'une partition primaire vierge (moins utile mais faisable).

Toutefois attention : vous ne pourrez pas mélanger partitionnement classique et LVM au sein d'un groupe de volumes.

Il est possible d'utiliser le LVM pour tous vos systèmes de fichiers, exception faite de « /boot », qui pose quelques problèmes. Par contre il est tout à fait possible de l'inclure sur un ou plusieurs volumes logiques. D'ailleurs la plupart des distributions proposent aujourd'hui cette option dès l'installation. La seule contrainte est de prévoir le chargement du module LVM dès le démarrage. Pour ce faire, on créera une image initrd contenant le module LVM à l'aide de la commande `lvcreate_initrd`. On modifiera également en conséquence le fichier `/etc/lilo.conf`.

## Configuration du LVM >>

Passons maintenant à la pratique.

Pour la mise en application, nous allons reprendre les 3 étapes énoncées ci-dessus. A chaque fois, je présenterai la commande correspondante à la création de l'élément et une commande de recueil d'informations pour vérifier que l'opération a été correctement réalisée.

## L'arborescence du LVM

À chaque élément du LVM correspond un fichier spécial dans `/dev`. Le volume physique est représenté par le fichier spécial du disque dur ou de la partition correspondant. Le groupe de volume dispose d'un répertoire portant son nom dans lequel on trouvera le fichier spécial group.

```
# ls -l /dev/datas/group
crw-r----- 1 root root 109, 1 jan 1 1970 group
```

Dans ce répertoire, on trouvera également un fichier spécial par volume logique créé à l'intérieur de ce groupe de volumes.

```
# tree /dev/users /dev/public
/dev/users
|--datas
|--group
`--private
/dev/public
|--ftp
|--group
`--web
```

Outre les fichiers spéciaux, on trouve également le fichier `/etc/lvmtab` et `/etc/lvmtab.d`. Ils contiennent la base de données manipulée par les commandes `lv`.

## Création d'un volume physique

La première étape consiste à transformer notre disque en volume physique. L'opération s'effectue en trois temps :

**Préparation de l'espace** à utiliser avec le LVM : cette étape s'effectue grâce à la commande `fdisk`. Vous allez devoir attribuer le type `lv` à votre disque ou votre partition :

```
root@pingu# fdisk /dev/hdb
Commande (m pour aide) : p
Disk /dev/hdb: 13.5 GB, 13578485760 bytes
255 heads, 63 sectors/track, 1650 cylinders
Units = cylindres of 16065 * 512 = 8225280 bytes
Périphérique Amorce Début Fin Blocs Id Système
/dev/hdb1      1      730 5863693+ 8e Linux LVM
/dev/hdb2     731     1339 4891792+ 8e Linux LVM
/dev/hdb3     1340     1650 2498107+ 8e Linux LVM
```

Ensuite, il est nécessaire de lancer la commande `vgscan` si vous utilisez LVM pour la première fois sur le

disque dur ou la partition. La commande va créer notamment le fichier `/etc/lvmtab` et le répertoire `/etc/lvmtab.d`.

```
# vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of your volume group
```

Création du volume physique : la commande est `pvccreate` (physical volume creation). `pvccreate [-f] </dev/hdxx>` où  
 -f : force la création du volume. A utiliser si le disque avait déjà été transformé en volume physique.  
 /dev/hdxx : fichier spécial du disque ou de la partition à transformer en volume physique Exemple :  
 création d'un volume physique à partir de hdb1

```
# pvccreate /dev/hdb1
pvccreate -- physical volume "/dev/hdb1" successfully created
```

## Création d'un groupe de volume

Une fois le volume physique créé, il faut alors insérer le ou les volumes physiques ainsi créés dans un groupe de volumes. On utilise la commande `vgcreate` :

`vgcreate <nom_du_volume></dev/hdxx>` où  
 <nom\_du\_volume> : nom du groupe de volume - l'opération crée alors le répertoire `/dev/nom_du_volume` contenant le fichier spécial group qui représente ce groupe de volumes.  
 </dev/hdxx> : fichier spécial du volume physique

Exemple : création d'un groupe de volumes nommé `volume1` avec `hdb1`

```
# vgcreate volume1 /dev/hdb1
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "volume1"
vgcreate -- volume group "volume1" successfully created and activated
```

## Création d'un volume logique

Une fois le groupe de volume créé, on peut alors le découper en un ou plusieurs volumes logiques grâce à la commande `lvcreate` :

`lvcreate -L tailleK|M|G [-n nom] <nom_volume>` où  
 -L tailleK|M|G : taille du volume logique exprimable en Ko, Mo ou Go  
 -n nom : nom du volume logique - l'opération crée un fichier spécial portant ce nom pour le volume logique et sera placé dans le répertoire `/dev/nom_volume`

<nom\_volume> : nom du groupe de volumes dans lequel sera créé le volume logique.

Exemple : création d'un volume logique de 600 Mo nommé `part1` dans le groupe de volume `volume1`

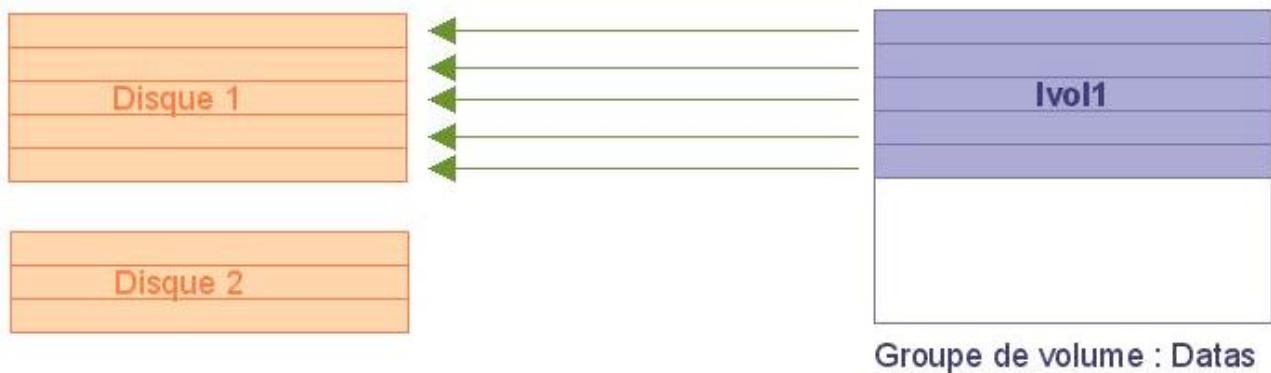
```
# lvcreate -L 600 -n part1 volume1
lvcreate -- doing automatic backup of "volume1"
lvcreate -- logical volume "/dev/volume1/part1" successfully created
```

Une particularité de la création d'un volume logique est le choix du mapping entre les LE et les PE. Par défaut, ce mapping est effectué de manière linéaire. Il est possible également de réaliser du stripping (répartition des données sur un ou plusieurs disques), ce qui permet d'améliorer le temps d'accès aux données. Ci-dessous, ce schéma montre la différence de répartition des LE en fonction de ces deux modes.

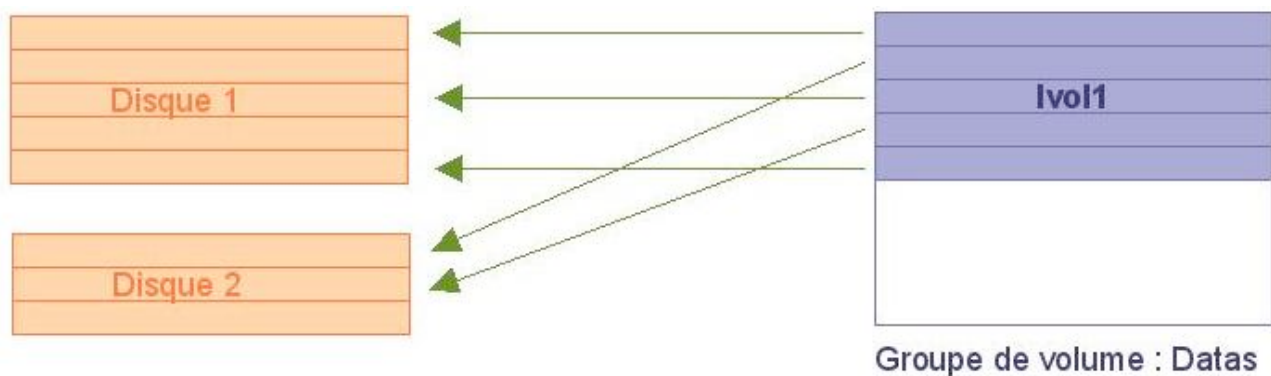
Des options de la commande `lvcreate` permettent de donner le nombre de stripes (et donc de volumes

physiques utilisés) et leur taille. (Schémas 4 et 5)

**Schéma 4 : Volume logique en mode linéaire**



**Schéma 5 : Volume logique en mode stripping**



## Recueillir des informations sur le LVM >>

A tout moment il est possible de recueillir des informations sur les structures LVM : volume physique, groupe de volumes et volumes logiques. On utilisera pour cela respectivement les commandes `pvdisplay`, `vgdisplay` et `lvdisplay`. Ces commandes ne font en fait qu'afficher dans un format lisible le contenu respectif de la PVRA, VGRA. Nous allons détailler les principales informations au moyen d'exemples ci-dessous.

La description d'un volume physique procure notamment le nom du volume physique, le nom du groupe de volume dans lequel est inséré le dit volume physique, sa taille, le nombre de volumes logiques contenus, la taille des physical extents, le nombre de PE contenus dans le volume physique, le nombre de PE libres.

```
# pvdisplay /dev/hdb1
--- Physical volume ---
PV Name      /dev/hdb1
VG Name      volume1
PV Size      5.59 GB [11727387 secs] /
   NOT usable 4.19 MB [LVM: 133 KB]
PV#          1
PV Status     available
Allocatable   yes
Cur LV       1
PE Size (KByte) 4096
Total PE      1430
Free PE       1230
Allocated PE   200
PV UUID       DTWrrWh-5oUP-Krdb-US55-c9wP-eKii-6z3uU7
```

La description d'un groupe de volumes permet de vérifier son nom, le type d'accès aux données

(écriture, lecture), le nombre maximum de volumes logiques créables dans ce groupe de volume, sa taille en PE et LE...

```
# vgdisplay volume1
--- Volume group ---
VG Name          volume1
VG Access         read/write
VG Status         available/resizable
VG #             0
MAX LV           256
Cur LV          1
Open LV          0
MAX LV Size      255.99 GB
Max PV           256
Cur PV          1
Act PV           1
VG Size          5.59 GB
PE Size          4 MB
Total PE         1430
Alloc PE / Size  150 / 600 MB
Free PE / Size   1280 / 5 GB
VG UUID          5XxOO1-ZNI8-zocw-6dR5-44LX-oyYc-MYHpN2
```

Enfin la description d'un volume logique contient son nom, le type d'accès aux données, sa taille...

```
# lvdisplay /dev/volume1/part1
--- Logical volume ---
LV Name          /dev/volume1/part1
VG Name          volume1
LV Write Access   read/write
LV Status         available
LV #             1
# open           0
LV Size          600 MB
Current LE        150
Allocated LE      150
Allocation        next free
Read ahead sectors 1024
Block device      58:0
```

## Commandes complémentaires >>

Jusqu'à maintenant, nous avons vu comment créer des structures LVM et obtenir de l'information. Nous allons voir maintenant d'autres opérations réalisables pour la gestion du partitionnement et qui mettent en évidence toute la souplesse apportée par le LVM en la matière : agrandir ou réduire un groupe de volume, redimensionner un volume logique.

## Redimensionner un groupe de volumes

Un groupe de volumes est constitué d'un ou plusieurs volumes physiques. Il est possible à tout moment d'ajouter ou retirer un ou plusieurs volumes physiques afin d'augmenter ou diminuer l'espace disponible d'un groupe de volumes.

Les commandes sont respectivement `vgextend` et `vgreduce` :

```
vgextend nom_volume /dev/hdxx
vgreduce nom_volume /dev/hdxx
```

où

`nom_volume` : nom du groupe de volume à redimensionner

`/dev/hdxx` : volume physique à ajouter ou retirer du groupe de volumes.



Vous devez augmenter l'espace du groupe de volume volume1 de 6 Go. Pour cela vous disposez d'un disque que vous allez ajouter. Ci-dessous les étapes à réaliser :

1. création du volume physique à partir de hdb2

```
# pvcreate /dev/hdb1
pvcreate -- physical volume "/dev/hdb1" successfully created
```

2. ajout de hdb2 au groupe de volumes volume1

```
# vgextend volume1 /dev/hdb2
vgextend -- INFO:maximum logical volume size is 255.99 Gigabyte vgextend
--doing automatic backup of volume group "volume1" vgextend
--volume group "volume1" successfully extended
```

On pourra vérifier la bonne réalisation de l'opération grâce à la commande vgdisplay.

## Redimensionner un volume logique

De la même façon, il est possible de diminuer ou augmenter la taille d'un volume logique au moyen des commandes lvreduce et lvextend.

```
lvextend -L taille /dev/nom_volume/vol_logique
lvreduce -L taille /dev/nom_volume/vol_logique
```

où

nom\_volume : nom du groupe de volume à redimensionner

dev/hdxx : volume physique à ajouter ou retirer du groupe de volumes

-L taille : taille finale du volume logique (après redimensionnement).

Pour ajouter ou retirer de l'espace sur un groupe de volume sans avoir à faire le calcul "espace actuel (+ ou -) espace à ajouter = taille", on utilisera les signes + ou - :

```
lvextend -L +taille /dev/nom_volume/vol_logique
lvreduce -L -taille /dev/nom_volume/vol_logique
```

où

-L +taille : taille initiale augmentée de la valeur utilisée pour taille.

-L -taille : taille initiale retranchée de la valeur utilisée pour taille.

Les étapes à respecter pour redimensionner un volume logique :

1. démontage du système de fichier (commande umount) ;
2. réduction / augmentation de la taille du système de fichiers : on utilisera pour cela un utilitaire fourni dans le package LVM ;
3. réduction / augmentation de la taille du volume logique ;
4. remontage du système de fichiers.

Si l'on veut augmenter l'espace du volume logique part1 en passant de 600 à 800 Mo, il ne nous restera plus qu'à procéder de la manière suivante :

```
# lvextend -L 800 /dev/volume1/part1
lvextend -- extending logical volume "/dev/volume1/part1" to 800 MB
lvextend -- doing automatic backup of volume group "volume1"
lvextend -- logical volume "/dev/volume1/part1" successfully extended
```

Là encore on pourra vérifier le bon déroulement de l'opération grâce aux commandes lvdisplay pour le volume logique et df pour le système de fichiers. Les remarques ci-dessus sont une description générale de l'opération. Ces étapes peuvent toutefois varier en fonction du système de fichiers utilisé. Certains procurent en effet la possibilité d'effectuer les opérations de redimensionnement « à chaud » c'est-à-dire sans démontage. Ci-contre un tableau comparatif des différents systèmes de fichiers les plus



couramment rencontrés (Tableau 1).

**Tableau 1 : manipulation des systèmes de fichiers**

Système de fichiers	diminution	Augmentation	Utilitaires
ext3	démontage préalable	démontage préalable	e2fsprogs (resize2fs)
ReiserFS	démontage préalable	opération on-line	reiserfsprogs (resize_reiserfs)
XFS	impossible	opération on-line	xfsprogs (xfs_growfs)

## Une fonction particulière du LVM : la réalisation de snapshots

La difficulté fréquemment rencontrée pour la réalisation de sauvegardes est de disposer de données cohérentes. Cela implique parfois d'arrêter un ou plusieurs services comme dans le cas des bases de données. Le LVM apporte un élément de réponse avec la possibilité de créer des snapshots. Il s'agit d'image à un moment  $t$  des données situées sur un volume logique. Le volume de snapshot ne nécessite pas autant d'espace que le volume initial dans la mesure où il ne contiendra réellement que les métadatas concernant les données à sauvegarder.

## Autres commandes

Cet article n'abordera pas le détail de toutes les commandes, qui sont plutôt simples à utiliser une fois que les concepts de base du LVM sont compris. Dans le tableau ci-après une liste de commandes utilisables et la description rapide de leur rôle. Voir le man de la commande pour plus d'informations.

### Commandes générales

- `lvcreate_initrd` : création d'une image initrd lorsque le système utilise le LVM
- `lvmdiskscan` : scanner l'ensemble des disques et partitions pour éditer une description de l'espace
- `vgscan` : création de `/etc/lvmtab` et `/etc/lvmtab.d`

### Gestion des volumes physiques

- `pvchange` : changer les attributs d'un PV
- `pvcreeate` : création d'un PV
- `pvdata` : afficher des informations de debug
- `pvdisplay` : afficher des informations d'un PV
- `pvsan` : lister tous les PV existant sur tous les disques

### Gestion des groupes de volumes

- `vgcgbbackup` : sauvegarder la VGDA
- `vgcgbrestore` : restaurer la VGDA
- `vgchange` : changer les attributs d'un VG
- `vgck` : vérification de la VGDA
- `vgcreate` : créer un VG
- `vgdisplay` : voir les informations
- `vgexport` : désactiver un VG pour pouvoir extraire les PV
- `vgimport` : activer et déclarer un VG sur le système
- `vgextend` : ajouter un ou plusieurs PV dans un VG
- `vgmerge` : fusionner deux VG
- `vgmknodes` : recréer `/dev/nom_volume` et le fichier spécial group
- `vgreduce` : extraire un ou plusieurs PV d'un VG
- `vgremove` : supprimer un VG
- `vgrename` : renommer un VG

### Gestion des volumes logiques

- `lvcreate` : création d'un VL
- `lvchange` : modification des attributs d'un VL
- `lvdisplay` : voir les informations d'un VL
- `lvextend` : augmenter la taille d'un VL
- `lvreduce` : réduire la taille d'un VL
- `lvremove` : supprimer un VL

- lvrename : renommer un VL
- lvscan : recherche de tous les VL existant

## Utilisation pratique du LVM dans la gestion de l'espace >>

Vous disposez d'un groupe de volume dans lequel vous n'avez plus d'espace disponible. Pour pouvoir augmenter un des volumes logiques, il va donc falloir ajouter un disque au groupe de volume. Comme pour tout élément du LVM, la première étape consiste à le transformer en volume physique ("pvcreate"). Puis on va insérer le nouveau volume physique dans le groupe de volume au moyen de la commande "vgextend". C'est tout ! Plus complexe, vous pouvez, pour quelle que raison que ce soit, envisager de déplacer un groupe de volume d'une machine à une autre. Si une sauvegarde de données est toujours conseillée par sécurité (mais comme vous êtes prévoyant vous en disposez de toute façon ;)), l'opération va consister à désactiver le groupe de volume à déplacer, enregistrer les métadatas le concernant. Puis on va réinjecter ces métadatas sur la machine destinataire et réactiver le groupe de volume sur cette machine. Les données seront alors accessibles de la même façon que sur l'ancienne machine (sous réserve de conserver les mêmes points de montage). Ci-dessous les étapes effectuées :

- 1. désactivation du groupe de volume ("datas") :  
# vgchange n datas
- 2. sauvegarde des métadatas du groupe de volume :  
# vgcgbackup datas
- 3. retrait du groupe de volume de la configuration système :  
# vgexport datas
- 4. arrêt de la première machine, transfert des disques dans la deuxième et redémarrage des machines
- 5. restauration des métadatas concernant ce groupe de volume sur la nouvelle machine :  
# vgcfgrestore -f datas.conf /dev/hda3<br /> /> # vgcfgrestore -f datas.conf /dev/hda4 si les volumes physiques sont hda3 et hda4.
- 6. déclaration du groupe de volume sur la machine :  
# vgimport datas /dev/hda3 /dev/hda4
- 7. activation du groupe de volume importé :  
# vgchange y datas

Il ne vous reste plus qu'à modifier en conséquence /etc/fstab pour le montage automatique des systèmes de fichiers contenus dans le groupe de volume.

## Evolution du LVM : LVM 2 >>

Avec la sortie du noyau 2.6, le LVM a été revu et corrigé et la version LVM2 est disponible de base avec ce noyau. Il est possible de l'utiliser avec un noyau 2.4.x moyennant recompilation de ce même noyau.

## Le device-mapper

Le driver du LVM a été complètement réécrit, ce qui lui procure encore plus d'efficacité et de souplesse. La grande nouveauté réside dans l'utilisation de ce driver, utilisé pour gérer la couche d'abstraction nécessaire dans le cadre de la gestion de volumes logiques. Cette couche d'abstraction a pour fonction principale de réaliser le mapping résultant de l'agrégat par bandes (stripping) des périphériques physiques utilisés. Le device-mapper définit les nouveaux périphériques de bloc composés de tranches de secteurs de périphériques physiques existant. Le mapping réalisé prend la forme suivante : <start > <length > <target > [ <target args...>] Les targets peuvent être de plusieurs natures :

- linear : c'est le cas le plus couramment utilisé dans le LVM. Les arguments nécessaires seront alors le device utilisé et le secteur de début.
- striped : on utilisera cette cible lorsque l'on réalise du stripping avec le LVM. Les arguments seront alors le nombre de stripes et leur taille, puis les paires device name / secteurs.
- error : toutes les I/O sur les secteurs ainsi marquées sont définies en erreur.
- snapshot : permet de réaliser des snapshots asynchrones grâce au LVM.
- mirror : permet d'implémenter les éléments nécessaires à l'exécution de la commande pvmove.

## Utilisation d'un arbre binaire

Pour réaliser le mapping, un **arbre binaire** a été utilisé, ceci afin de rendre la lecture de la table plus rapide et donc le LVM plus efficace.

## Une plus grande configurabilité

---

LVM2 peut fonctionner sans ajout de fichier de configuration mais l'emploi de celui-ci permet d'optimiser ses performances. Un certain nombre d'éléments peuvent ainsi être paramétrés :

- les devices à utiliser pour réaliser le LVM, ce qui permet d'éviter des scans de périphériques inutiles et qui nuisent à la performance (ex : lecteur de CD-ROM). A cela, on ajoute la gestion d'un système de cache contenant ces informations qui permet d'accroître encore plus l'efficacité.
- possibilité de déterminer l'emplacement des fichiers spéciaux des groupes volumes
- possibilité de disposer de logs configurables (taille, contenu, emplacement...)
- paramétrage des backups de la configuration existante et de l'archivage des anciennes configuration du LVM (métadatas)
- définition du type de LVM employé par défaut (1 ou 2), même s'il est possible dans la compilation de n'inclure que la gestion du LVM2, dans le cas où la gestion de la compatibilité descendante n'est pas nécessaire.
- définition du nombre de copies de secours des métadatas sur un volume physique, un peu à l'image des copies des superblocs sur le système de fichiers de type ext2.

## Compatibilité LVM1 / LVM2

---

Il est possible d'utiliser conjointement LVM1 et 2 sur un même système, et/ou de convertir du LVM 1 en 2 et inversement. On peut effectivement utiliser sur un même système les deux versions de LVM, à condition que ce ne soit pas dans le même groupe de volumes. Les commandes LVM ont en effet un commutateur supplémentaire, -M, qui permet de faire ce choix. (-M 1 ou -M 2) D'autre part, la commande `vgconvert` permet la conversion des métadatas pour migrer de LVM1 à LVM2

## Autre

---

LVM2 permet d'assouplir encore plus la gestion de l'espace, dans la manipulation des volumes logiques. La commande `lvcreate` par exemple donne maintenant la possibilité de choisir le périphérique voire la tranche de PE à utiliser.

## En conclusion... >>

---

Voilà donc un outil de plus qui fait qu'un système Linux peut être véritablement efficace et optimiser la disponibilité d'un serveur en production. L'outil LVM peut également être utilisé avantageusement sur un poste personnel, pour s'éviter les opérations fastidieuses liées à la gestion de l'espace disque.

## Liens utiles >>

---

- HOWTO LVM : <http://tldp.org/HOWTO/LVM-HOWTO>
- La principale mailing-liste d'aide : <http://lists.sistina.com/mailman/listinfo>
- Trouver les sources de LVM : [http://www.sistina.com/products\\_lvm.htm](http://www.sistina.com/products_lvm.htm)
- Un autre projet de gestion de volumes logiques, EVMS : <http://evms.sourceforge.net>

Cette page est issue de la documentation 'pré-wiki' de Léa a été convertie avec HTML::WikiConverter. Elle fut créée par Anne le 29/10/2004.

## Copyright >>

---

Copyright © 29/10/2004, Anne



Ce document est publié sous licence **Creative Commons**  
*Attribution, Partage à l'identique, Contexte non commercial 2.0 :*  
**<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>**

---