

LVM: overview

Table of Contents

- [LVM: overview](#)
 - [Introduction](#)
 - [Avantages de LVM](#)
 - [Inconvénients de LVM](#)
- [Utilisation basique de lvm](#)
 - [Notions et vocabulaire](#)
 - [Glossaire](#)
 - [Volume physique](#)
 - [Groupe de volumes](#)
 - [Volume logique](#)
 - [Notion d'« extent »](#)
- [Commandes utiles](#)
 - [Monter une partition](#)
 - [Vérifier une partition](#)
 - [Afficher les informations](#)
 - [Créer/supprimer un volume](#)
 - [Suppression d'un volume](#)
 - [Redimensionnement](#)
 - [lvextend : augmenter](#)
 - [lvresize : augmenter ou réduire](#)
- [Snapshot](#)
 - [Utilisation du snapshot pour cloner un serveur](#)
 - [Autre usage des snapshots : utilisation du merge](#)

Introduction

LVM (**L**ogical **V**olume **M**anager, ou gestionnaire de volumes logiques en français) permet la création et la gestion de volumes logiques 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.

Depuis Intrepid, il existe le paquet `system-config-lvm` dans les dépôts Universe, qui propose une interface graphique.

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

Si un des volumes physiques devient HS, alors c'est l'ensemble des volumes logiques qui utilisent ce volume physique qui sont perdus. Pour éviter ce désastre, il faudra utiliser LVM sur des disques raid par exemple.

Utilisation basique de lvm

Installer le paquet `lvm2`.

Notions et vocabulaire

Glossaire

abréviation	anglais	français	description
VG	Volume Group	Groupe de Volumes	
LV	Logical Volume	Volume Logique	une "partition" dans un groupe de volumes
PV	Physical Volume	Volume Physique	
PE	Physical Extent	Etendue Physique	un tout petit morceau d'un groupe de volumes

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.

avant de pouvoir définir le volume physique, il est nécessaire qu'une partition existe.

Groupe de volumes

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.



On peut y mettre à peu près tout ... mais il vaut mieux éviter d'y mettre:

* **/boot**. Techniquement, ça doit fonctionner à peu près correctement depuis Grub 2, mais cela est encore sensible suivant la configuration (et perdre la capacité de démarrer est ... gênant), pour le moment il est vivement conseillé d'avoir au moins une partition de l'ancien temps avec /boot (ou / si le dossier /boot n'est pas séparé) pour éviter les ennuis.

* **/boot/grub** car cela peut poser des problèmes pour le démarrage du système. En effet, grub ne sait pas encore (novembre 2015) écrire dans un fichier localisé dans un tel volume. Or grub mémorise le résultat du démarrage courant dans le fichier /boot/grub/grubenv de manière à modifier le démarrage qui suit un démarrage défaillant.

Notion d'« extent »

Un extent, ou « physical extent » 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.

Commandes utiles

Monter une partition

- Obtenir la liste des groupes logiques

```
lvm vgscan
```



cette commande devrait créer les nodes dans le répertoire /dev (sous \dev directement ou dans \dev\mapper selon la distribution) s'ils n'existent pas (en mode rescue par exemple) relancer la commande avec `lvm vgscan --mknodes`

* Obtenir la liste des partitions

```
lvm lvs
```

- Rendre la partition disponible

```
lvm lvchange -ay /dev/VolGroup01/LogVol00
```

- Monter la partition

```
mount /dev/VolGroup01/LogVol00 /media/user/point_de_montage
```

Exemple

```
user@ubuntu:~$ sudo lvm vgscan
Reading all physical volumes. This may take a while...
Found volume group "Groupe_lvm" using metadata type lvm2
user@ubuntu:~$ sudo lvm lvs
  LV      VG      Attr      LSize      Pool Origin Data%
Move Log Copy% Convert
home Groupe_lvm -wi-ao--- 500.00g
root  Groupe_lvm -wi-a---- 20.00g
swap Groupe_lvm -wi-a---- 4.00g

user@ubuntu:~$ sudo lvm lvchange -ay /dev/Groupe_lvm/home
user@ubuntu:~$ sudo mkdir /media/user/montage
user@ubuntu:~$ sudo mount /dev/Groupe_lvm /home /media/user/montage
```

Vérifier une partition

* Rendre la partition disponible

```
lvm lvchange -ay /dev/VolGroup01/LogVol00
```

- Lancer fsck

```
sudo fsck -f -y /dev/VolGroup01/LogVol00
```

Afficher les informations

- **lvs** (en liste) ou **lvdisplay** (plus complet) pour afficher des informations à propos des LV. lvs seul va lister tous les volumes logiques du système.
- **vgs** (en liste) ou **vgdisplay** (plus complet) pour afficher des informations à propos des VG.

Lorsqu'on fait vgs, on obtient ce genre d'information :

```
VG      #PV #LV #SN Attr   VSize VFree
grokska 1  24  1 wz--n- 1,80t 1,09t
```

- **PV** indique le nombre de volume physique (ici, un)
- **LV** le nombre de volume logique
- **VSize** la taille totale du groupe de volume sur le disque dur
- **VFree** indique combien d'espace n'est pas utilisé dans le VG par des LV : c'est donc ce qu'on peut exploiter.

Créer/supprimer un volume

En résumé, :

- créer un PV

```
pvcreeat /dev/sda3
```

- créer un VG

```
vgcreate NOM_VG /dev/sda3
```

- créer les LV

```
lvcreate -n NOM_LV -L TAILLE NOM_VG
```

- **NOM_LV**: nom qu'on donne au volume logique
- **TAILLE**: taille, par exemple 500m, 10g ou 1t
- **NOM_VG**: nom du groupe de volume dans lequel le LV va être.

Si le volume est à utiliser pour du stockage, il faut le formater et lui attribuer un système de fichier :

```
mkfs.ext4 /dev/NOM_VG/NOM_LV
```

Vérifier que tout va bien en la montant :

```
mkdir /mnt/lv
```

```
mount /dev/NOM_VG/NOM_LV /mnt/lv
```

L'ajouter dans le système où elle sera montée (donc depuis une VM si ce n'est pas depuis le système hôte, et dans ce cas ce sera /dev/xvdXX), dans /etc/fstab pour que ce soit monté au démarrage :

```
/dev/NOM_VG/NOM_LV /mnt/lv ext4 defaults 0 2
```

Suppression d'un volume

Pour supprimer un volume, s'assurer qu'il est démonté, puis simplement :

```
lvremove /dev/NOM_VG/NOM_LV
```

Redimensionnement

Vérifier la place occupée sur le volume et restant sur le disque avant.

Il y a deux commandes possibles : `lvresize` qui a des risques non négligeable de créer des soucis, et `lvextend` qui demande plus de manipulation mais est plus sécurisé.

Il faut que la taille du volume soit en rapport avec la table de partition du volume.

lvextend : augmenter

On commence par augmenter la taille du volume. On peut soit dire combien on veut ajouter (`-size`, soit donner tout de suite la taille finale à laquelle on veut arriver (`-size`) :

```
lvextend --size +40G /dev/vg0/foo
```

Ou pour donner la taille finale voulue :

```
lvextend --size 120G /dev/vg0/foo
```

Un message “successfully resized” doit apparaître.

Vérifier la nouvelle taille avec `lvdisplay`

```
lvdisplay /dev/vg0/foo
```

Vérifier ensuite qu'il n'y a pas de souci avec le volume en question, sinon l'étape suivante va foirer :

```
fsck -f /dev/mapper/vg0-foo
```



fsck préfère passer par `/dev/mapper`.

Lancer ensuite la prise en compte du redimensionnement :

```
resize2fs /dev/vg0/foo
```

lvresize : augmenter ou réduire

`lvresize` est plus puissant car il permet d'étendre ou de réduire. Mais si on réduit en dessous du dernier bit utilisé sur le volume, on va avoir un volume corrompu).

Il faut d'abord réduire la taille du système de fichier, puis réduire la partition elle-même, sans quoi la partition se corrompt.

Si la partition est en ext2, 3 ou 4, on utilise `resize2fs`.

Pour connaître la taille minimale de la partition :

```
resize2fs -P /nom/de/la/partition
```

Pour réduire à 50G par exemple :

```
resize2fs /nom/de/la/partition 50G
```

Utiliser `lvresize` (dans l'exemple, on réduit à 100Go) :

```
lvresize -r -L 100g /dev/mapper/svg-ca
```

En théorie, avec l'option `-r`, pas besoin de faire `resize2fs` avant, `lvresize` s'occupera de redimensionner le système de fichier puis ensuite la partition. L'option `-L` indique la taille finale que l'on souhaite obtenir.



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-Vol` sont synonymes. Dans la pratique, il est conseillé quand même d'utiliser plutôt la forme `/dev/mvg/Vol1`, certaines commandes ne passeront pas autrement.

Snapshot

L'un des gros intérêt de LVM est le snapshot. Cela va prendre l'instantané d'un volume à un moment donné ; on peut ensuite revenir à cet état antérieur.

Quand on fais un snapshot, il le lie au disque d'origine et y note les modifications effectuées. Le disque continue à vivre sa vie, mais tout ce qui est modifié dessus est noté sur le snapshot. Quand on supprime le snapshot, c'est comme supprimer ses notes, elles sont perdues, on ne peut plus revenir en arrière. Quand on veux revenir au point de départ, il va appliquer toutes les modifications à l'envers sur le disque d'origine, en utilisant les données du snapshot, puis le snapshot étant devenu inutile, il le supprime aussi.

Avant de créer un snapshot, il faut s'assurer qu'il aura ce qu'il faut d'espace. Pas besoin de le faire aussi gros que le volume d'origine ! Le plus simple est de vérifier la taille du volume d'origine avec `df` ou `ncdu`.

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda2      30G   2.1G   26G   8% /
```

Donc, ici, prévoir un snapshot de 10Go sera largement assez pour démarrer.

```
lvcreate -L 10g -s -n lv_test_20110617 /dev/volume_groupe/lv_test
```

Va créer un snapshot du LV `lv_test` à la taille de 10Go qui va avoir comme nom `lv_test_20110617`. La taille d'utilisation du snapshot évolue avec l'utilisation. Si ce snapshot se retrouve rempli à 100%, il devient alors inutilisable (état "INACTIVE") mais pas d'inquiétude car il n'y a pas d'impact pour le LV d'origine⁴) Le snapshot se remplit à mesure que l'état initial du disque est modifié : il enregistre tout ce qui a été fait, pour le défaire et revenir à l'état antérieur si on lui demande. L'espace peut donc se remplir !

Lors de la création d'un volume logique, l'option `-s` crée un snapshot d'un volume existant.

Un snapshot conserve toutes les modifications apportées au LV d'origine, afin de pouvoir les rejouer à l'envers, soit à la volée si on utilise le snapshot comme périphérique, soit lors d'un merge. Ainsi il est vide au départ et se remplit au fur et à mesure que des modifications sont apportées au LV d'origine.

Si le volume logique est utilisé pour une VM, il vaut mieux éteindre la VM, faire le snapshot, puis rallumer la VM. Cela évitera d'avoir des incohérences lié à ce qui est stocké dans la RAM, le cache disque etc... De même, si une écriture se fait durant le snapshot, cela pourrait amener des incohérences.

Cela reste un risque faible (le snapshot est relativement rapide) mais éteindre la VM évitera de potentiels problèmes.

Utilisation du snapshot pour cloner un serveur

On va voir comment cloner liria pour en faire spofu, et envoyer spofu sur un autre serveur. en utilisant des snapshots pour réduire le temps d'inactivité. Le cas pratique est la copie de Liria pour faire Spofu.

La création du snapshot sert pour deux raisons :

- La copie d'un disque en train d'être modifié risque fortement d'amener à des incohérences. Par exemple, un fichier créé alors que le répertoire qui le contient est déjà copié ne sera pas accessible.
- La copie d'une partition complète est longue. Le snapshot permet de réduire ce temps d'arrêt.

La création du snapshot prenant quelques secondes, le serveur est arrêté moins d'une minute.

Copier un serveur est très long, ce n'est pas forcément la façon la plus rapide de procéder, mais parfois, c'est utile.

On commence par éteindre liria, faire son snapshot, puis copier ce dernier (sur le même disque dur) avant de relancer liria. Copier avec dd avant de relancer liria, sinon le snapshot va enregistrer les modifications et c'est le bazar. Copier tout le volume ; même si, techniquement, un snapshot plus petit (avec uniquement les données écrites) devrait aussi marcher, nos tests n'ont pas été concluants. Créer le volume logique sur le serveur de destination avant de commencer le transfert, sinon il peut y avoir des erreurs pour "manque de place" (même s'il y a la place).

```
# screen -DR
# lvcreate -s -L 10G -n liria-disk-snap /dev/groska/liria-disk
# dd if=/dev/groska/liria-disk-snap of=/dev/groska/spofu-disk
```

Sur le serveur de destination (nuxru), on prépare l'arrivée de spofu en créant également le LV destiné au swap.

```
screen -DR
lvcreate -L 50G -n spofu-disk groska
lvcreate -L 4G -n spofu-swap groska
mkswap /dev/nuxru/spofu-swap
```

Sur le serveur d'origine (groska) :

```
screen -DR
lvcreate -s -L 50G -n liria-disk-snap /dev/groska/liria-disk
lvcreate -L 50G -n spofu-disk groska
dd if=/dev/groska/liria-disk-snap of=/dev/groska/spofu-disk
```

Ensuite, on vérifie que les deux serveurs peuvent communiquer via ssh, on installe pv si ce n'est pas fait (cela permet de suivre la progression de l'échange).

La commande suivante est à exécuter depuis le serveur primaire vers le serveur secondaire (toujours sous screen ou tmux pour qu'une déconnexion ssh ne casse pas la commande) :

```
dd if=/dev/groska/spofu-disk bs=4096 | pv | ssh root@192.168.0.20 dd  
of=/dev/nuxru/spofu-disk bs=4096
```

Il faut ensuite vérifier que le transfert s'est bien passé et donc comparer le checksum des deux volumes :

```
md5sum /dev/groska/spofu-disk  
md5sum /dev/nuxru/spofu-disk
```

Le swap a été créé plus haut.

Autre usage des snapshots : utilisation du merge

Une autre utilisation possible du snapshot est de tester quelque chose et de revenir en arrière si ça se passe mal. Dans ce cas, on va commencer comme à l'étape 1. Après avoir relancée la VM dont on a fait un snapshot, on peut faire les modifications et tester. Si tout va bien, on efface simplement le snapshot, sinon il faut revenir en arrière.

On arrête donc la VM, puis on merge le snapshot, afin d'appliquer à l'envers toutes les modifications effectuées. Cette étape peut être un peu longue, en tout cas plus que la création ou la suppression du snapshot.

```
# lvconvert --merge /dev/groska/liria-disk-snap
```

Le snapshot, devenu inutile, est détruit dans l'opération.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=debian:lvm-overview>

Last update: **2025/02/19 10:59**

