



Guide utilisateur Linux

Gestion des systèmes de fichiers, partitions et volumes LVM

Principes et exemples pratiques

LASSERRE Jean baptiste

Gestion des systèmes de fichiers, partitions et volumes LVM

Guide utilisateur Linux – Principes et exemples pratiques

Table des matières

1	À propos	4
2	Partitionnement classique et volumes LVM.....	5
3	Principe de création et agrandissement d'un système de fichiers	10
3.1	Création d'un système de fichiers	10
3.1.1	Partitionnement classique.....	10
3.1.1.1	En partant d'un disque dur vierge.....	10
3.1.1.2	En partant d'un disque dur déjà partitionné et avec de l'espace libre	10
3.1.2	Partitionnement avec LVM.....	11
3.1.2.1	En partant d'un disque dur vierge.....	11
3.1.2.2	En partant d'un disque dur déjà partitionné et avec de l'espace libre	11
3.2	Agrandissement d'un système de fichiers.....	12
3.2.1	Partitionnement classique.....	12
3.2.2	Partitionnement avec LVM.....	13
3.2.2.1	Cas où LVM est placé directement sur le disque	13
3.2.2.2	Cas où LVM est placé sur une partition qu'il est possible d'agrandir	13
3.2.2.3	Cas où LVM est placé sur une partition qu'il n'est pas possible d'agrandir	14
4	Cas pratique : création d'un volume LVM et d'un système de fichiers.....	17
4.1	Utilisation d'un disque vierge.....	17
4.2	Utilisation d'un disque partitionné	20
5	Cas pratique : agrandissement d'un système de fichiers.....	25
5.1	Ajout de ressources de stockage à une machine virtuelle	25
5.1.1	Agrandissement d'un disque.....	25
5.1.2	Ajout d'un nouveau disque	26
5.2	Agrandissement d'une partition classique.....	27
5.3	Agrandissement d'un volume logique.....	30
5.3.1	Cas n°1 – Disque sans partition	31
5.3.2	Cas n°2 – Disque avec partition extensible	34
5.3.3	Cas n°3 – Disque avec partition étendue extensible	40
5.3.4	Cas n°4 – Disque avec partition non extensible	47



6	Cas pratique : réorganisation des volumes LVM.....	52
6.1	Affichage de la répartition des extents des volumes physiques	52
6.2	Regroupement de groupes de volumes	53
6.3	Séparation d'un groupe de volumes en deux	58
6.4	Déplacement d'un volume physique ou logique dans un autre groupe de volumes	63
6.5	Réorganisation des volumes LVM d'une machine	63
7	Mémento des commandes utiles pour LVM	73
8	Sources	75

1 À propos

Ce document est un guide sur la création et l'agrandissement de systèmes de fichiers et des éléments sous-jacents, à savoir les partitions et les volumes LVM. Le point de départ est toujours une action à faire sur le système de fichiers et le guide indique la procédure à suivre selon la situation. L'exhaustivité n'est pas recherchée, chaque situation est unique, mais les cas de base présentés ici permettent de s'adapter à la plupart des scénarios possibles. Chaque action du guide est expliquée de façon théorique et un exemple pas-à-pas est ensuite donné.

2 Partitionnement classique et volumes LVM

Les disques durs forment un ensemble de blocs pouvant stocker des données. L'organisation logique des blocs est assurée par le système de fichiers. Il tient à jour un registre des blocs utilisés afin de retrouver les informations stockées. Un ensemble de blocs ne peut être géré que par un seul système de fichiers. Il est cependant possible de créer des partitions afin de créer des ensembles de blocs indépendants.

Un disque est donc composé d'une ou plusieurs partitions, sur lesquelles est installé un système de fichiers.

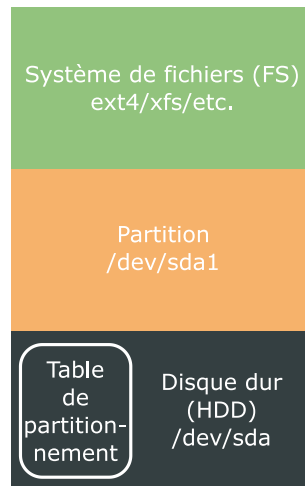


Figure 1 – Utilisation classique d'un disque dur : partitionnement et formatage

La liste des partitions est stockée dans la table de partitionnement, tout au début du disque. Les partitions sont soumises à des contraintes techniques très fortes :

- quatre partitions primaires maximum (limite historique, contournée aujourd'hui) ;
- les partitions sont physiquement placées les unes derrière les autres sur le disque ;
- une partition occupe un espace contigu (impossible d'en agrandir une s'il n'y a pas d'espace libre attenant) ;
- un système de fichiers est directement lié à une partition (il est soumis aux mêmes limites physiques).

Pour aller au-delà des limitations physiques des partitions, il est possible d'utiliser une couche d'abstraction entre le disque dur/partition (matériel) et le système de fichiers (logiciel). Sous Linux, [LVM](#) fournit cette couche d'abstraction en permettant d'agréger plusieurs disques et partitions en disques logiques qui ne sont plus soumis à des contraintes matérielles, ce qui est très utile lors de l'utilisation de machines virtuelles. LVM est un outil très complet qui permet également de faire du RAID logiciel et même de déplacer des données sur de nouveaux supports physiques, à chaud et en toute transparence. Tout ne sera pas abordé ici, ce document se focalisant sur la gestion des volumes de données (création, agrandissement, déplacement).

La couche d'abstraction LVM s'utilise soit sur une partition, soit directement sur le disque (qui ne possède plus de table de partitionnement).

Le principe de fonctionnement de LVM est le suivant :

1. sur le disque ou une de ses partitions, un volume physique (PV) est créé, indiquant que l'unité de stockage est gérée par LVM ;
2. un ou plusieurs volumes physiques sont regroupés dans un groupe de volumes (VG), cela permet d'agréger plusieurs disques et partitions en un seul disque logique ;
3. un ou plusieurs volumes logiques (LV) sont créés dans le groupe de volumes (VG), ce sont des équivalents des partitions, utilisables par un système de fichiers.

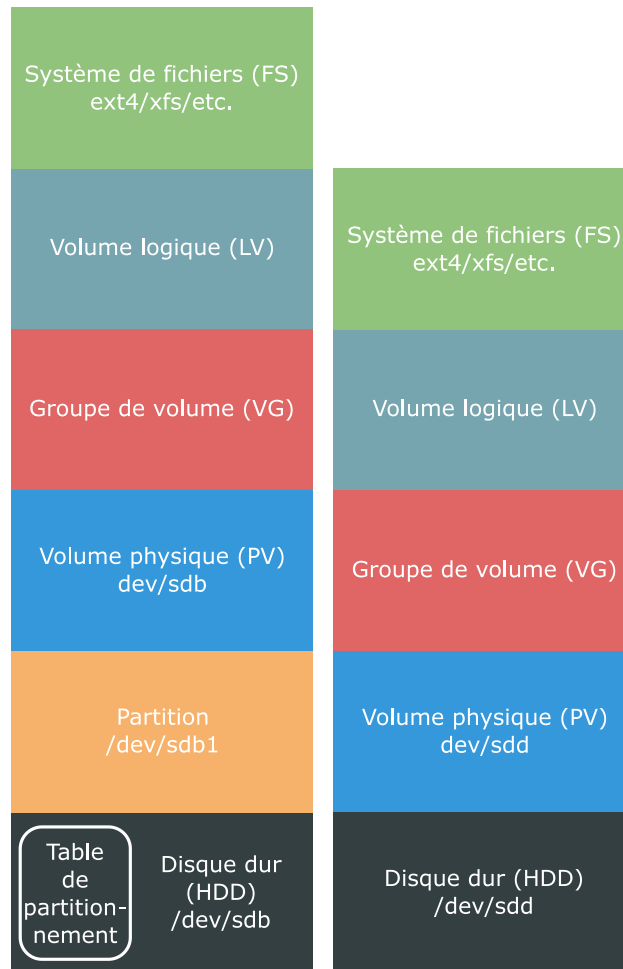


Figure 2 – Utilisation de LVM sur un disque partitionné ou non

Ce fonctionnement en trois niveaux offre une grande souplesse. Pour agrandir la taille d'un système de fichiers, il suffit d'agrandir en amont celle du volume logique sous-jacent. Cette dernière n'est contrainte que par l'espace restant sur le groupe de volumes. S'il y a suffisamment d'espace non alloué, le volume logique peut être agrandi. Sinon il suffit d'ajouter un disque ou une partition au groupe de volumes, ou bien agrandir un disque ou une partition existante. La modification est directement prise en compte. C'est LVM qui gère l'écriture des données sur les différents volumes physiques.

Il est tout à fait possible de mixer partitionnement classique et volumes LVM sur une même machine ou un même disque. La figure ci-dessous résume les différents cas possibles et permet de visualiser la flexibilité apportée par LVM.

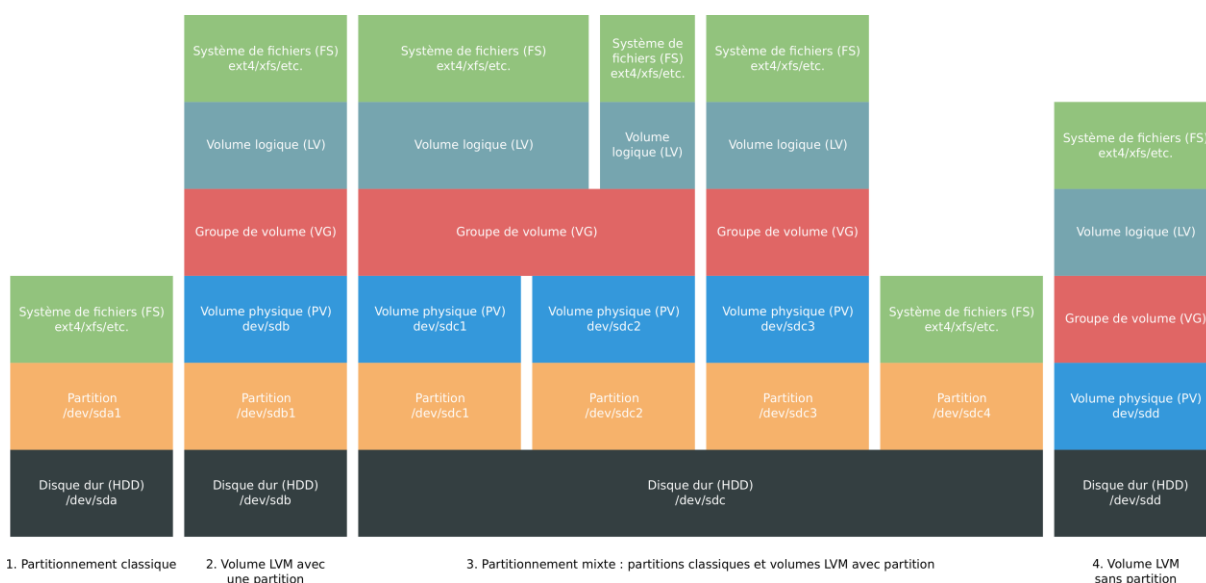


Figure 3 – Synthétique des différents cas de partitionnement classiques et LVM

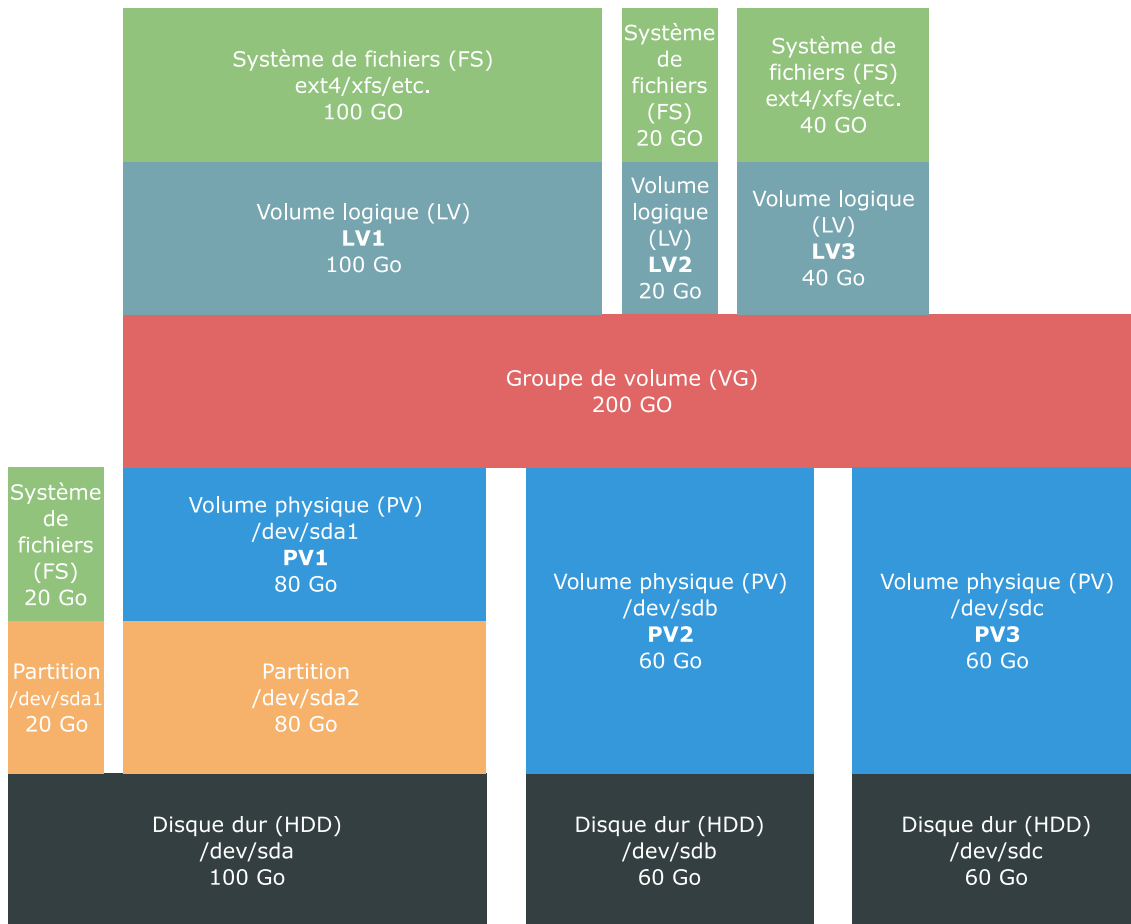
Note : Un PV et un LV appartiennent toujours à un et un seul VG. Un VG est une liste de PV et contient un à plusieurs LV.

Attention : La plupart des distributions Linux ont besoin que la partition montée dans `/boot` soit formatée de façon classique.

Quand une unité de stockage (disque ou partition) est gérée par LVM, elle est décomposée en blocs de données appelés « extents ». Ces blocs de données ont tous la même taille et sont interchangeables, ce qui permet de les rendre indépendants du support physique sous-jacent. Au niveau du groupe de volumes (VG), le stockage est donc vu comme un ensemble d’extents. À chaque fois qu’un volume logique (LV) est créé ou agrandi, LVM va allouer un certain nombre d’extents libres à ce volume. Ainsi, pour chaque volume physique d’un VG, LVM conserve une liste indiquant quels sont les extents alloués et à quels volumes logiques ils sont alloués.

La figure FIGURE 4 ci-dessous illustre le fonctionnement des extents, en prenant pour exemple une machine possédant trois volumes physiques (PV) regroupés dans un même groupe de volumes (VG), dans lequel trois volumes logiques (LV) ont été créés. Le groupe de volumes a une taille totale de 200 Go répartie en 50 extents de 4 Go chacun. La figure montre la façon dont LVM a réparti les extents entre chaque volume logique et la dissociation complète entre les systèmes de fichiers et le matériel de la machine.





Configuration de la machine exemple

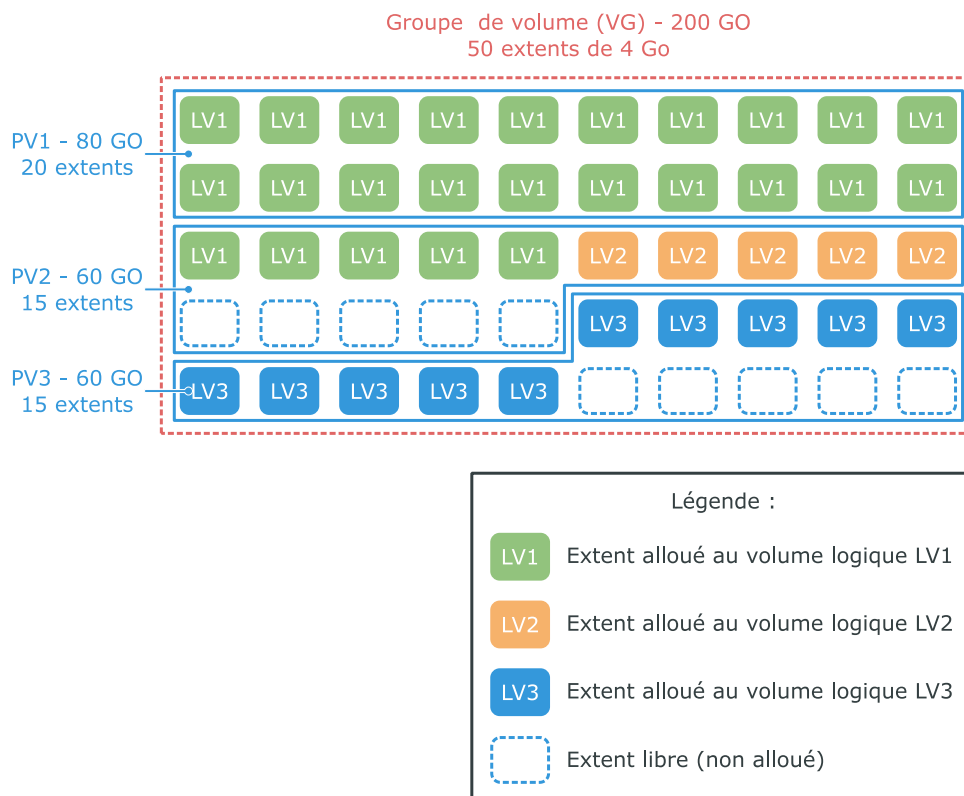


Figure 4 – Exemple d'allocation des extents par LVM

L'utilisateur peut laisser LVM choisir les PV à utiliser pour l'allocation des extents, mais il peut aussi indiquer explicitement combien d'extents allouer à chaque PV. Et il est également possible par la suite de déplacer les extents à chaud d'un volume physique vers un autre et même de réorganiser complètement les PV, VG et LV. Chose quasiment impensable avec des partitions classiques.

Cette organisation en extents offre donc une grande souplesse et est un des mécanismes faisant la force de LVM.

3 Principe de création et agrandissement d'un système de fichiers

3.1 Création d'un système de fichiers

3.1.1 Partitionnement classique

3.1.1.1 En partant d'un disque dur vierge

Les actions suivantes sont nécessaires avant l'utilisation d'un disque dur vierge :

1. création d'une table de partitionnement (MBR ou GPT) ;
2. création d'une partition ;
3. création d'un système de fichiers sur la partition.

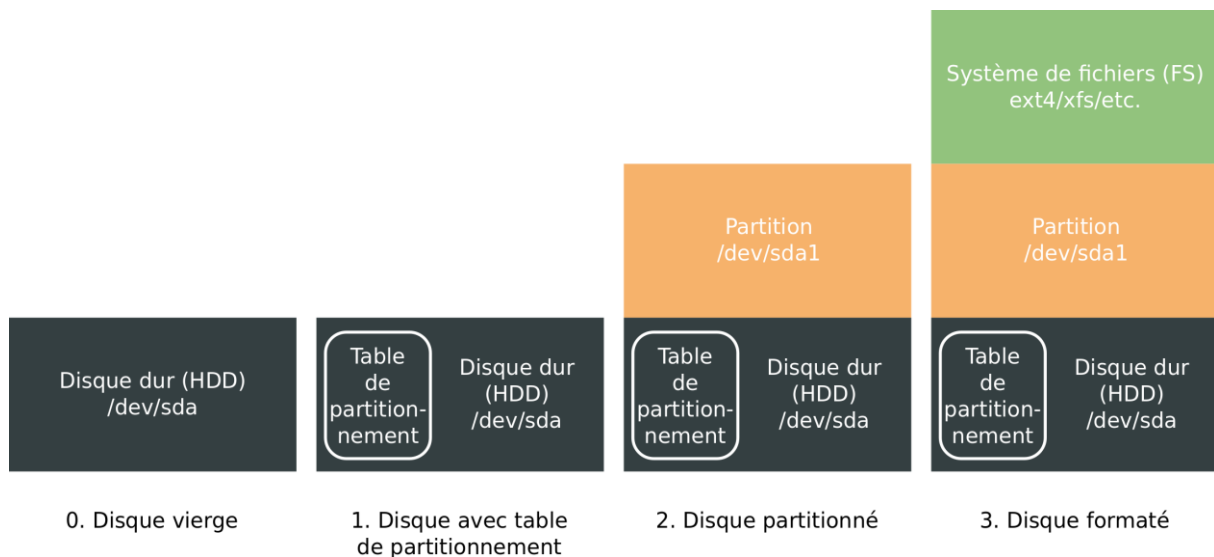


Figure 5 – Formatage classique d'un disque dur vierge

3.1.1.2 En partant d'un disque dur déjà partitionné et avec de l'espace libre

Un disque possédant déjà une table de partitionnement et de l'espace non alloué permet de créer facilement une nouvelle partition :

1. création d'une partition ;
2. création d'un système de fichiers sur la partition créée.

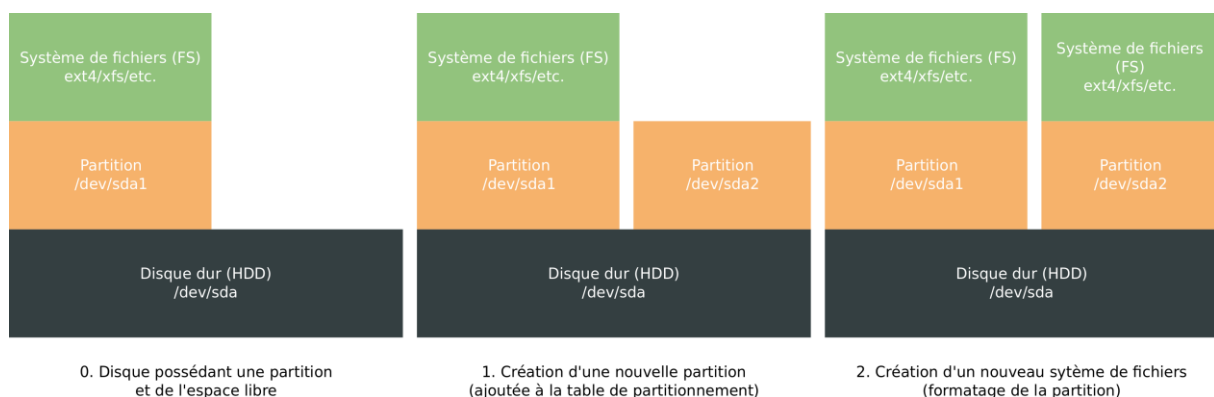


Figure 6 – Ajout et formatage d'une partition classique

3.1.2 Partitionnement avec LVM

Si le disque dur est encore vierge, il est recommandé de l'utiliser directement avec LVM, sans créer de partition. Cela facilitera ultérieurement l'agrandissement du disque.

Il reste malgré tout possible de créer une partition occupant le disque entier et d'utiliser LVM dessus (c'est même un cas courant), mais ce cas de figure n'est pas abordé. Il est cependant très proche du cas où le disque est séparé en plusieurs partitions.

3.1.2.1 En partant d'un disque dur vierge

La préparation d'un disque vierge pour une utilisation exclusive par LVM est très simple :

1. création d'un volume physique (PV) sur le disque ;
2. création d'un groupe de volumes (VG) contenant le volume créé ;
3. création d'un volume logique (LV) sur le groupe de volumes créé ;
4. création d'un système de fichiers sur le volume logique créé.

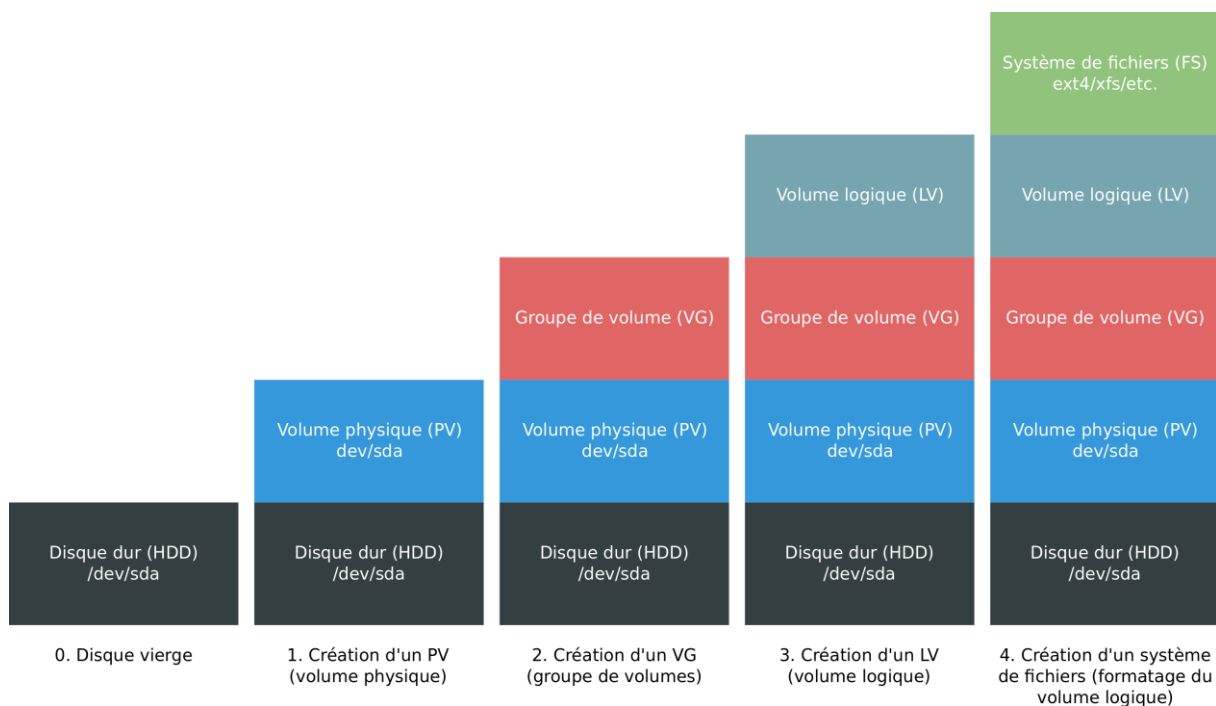


Figure 7 – Formatage avec LVM d'un disque dur vierge

3.1.2.2 En partant d'un disque dur déjà partitionné et avec de l'espace libre

La souplesse de LVM lui permet de cohabiter avec le système de partitionnement classique en fonctionnant au-dessus d'une partition :

1. création d'une partition ;
2. création d'un volume physique (PV) sur la partition créée ;
3. création d'un groupe de volumes (VG) contenant le volume créé ;
4. création d'un volume logique (LV) sur le groupe de volumes créé ;
5. création d'un système de fichiers sur le volume logique créé.

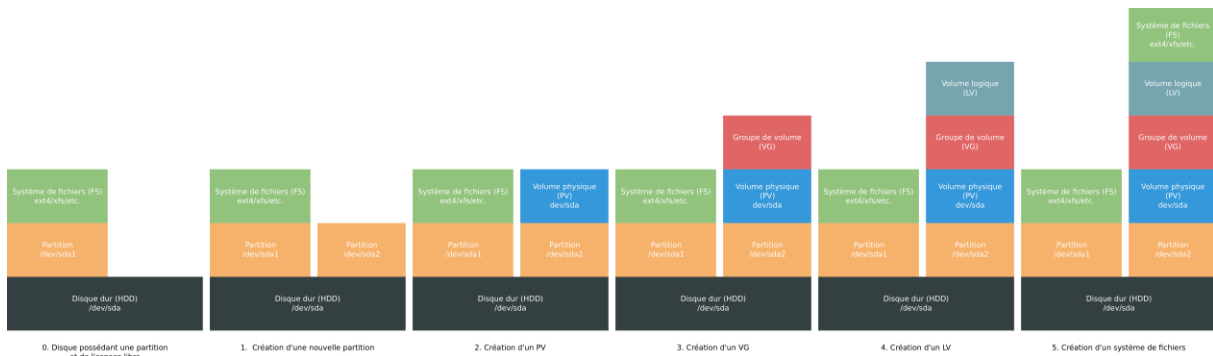


Figure 8 – Ajout et formatage d'une partition avec LVM

3.2 Agrandissement d'un système de fichiers

L'agrandissement d'un système de fichiers nécessite d'avoir de l'espace de stockage non alloué sur un des disques de la machine. Dans le cas des machines virtuelles, il est possible d'agrandir la taille du disque, pour les machines physiques il faut soit avoir de l'espace soit ajouter un nouveau disque.

Avec un partitionnement classique, il faut de l'espace libre immédiatement en suivant de la partition. En utilisant LVM, il est possible d'agrandir/ajouter un disque ou bien agrandir/ajouter une partition.

3.2.1 Partitionnement classique

Une partition ne peut pas déborder sur deux disques, pour être agrandie, il faut de l'espace disponible sur le disque. La figure suivante montre les différents cas possibles. Si la partition à agrandir est suivie d'espace libre puis d'une autre partition, il sera possible de l'agrandir dans la limite de cet espace libre.

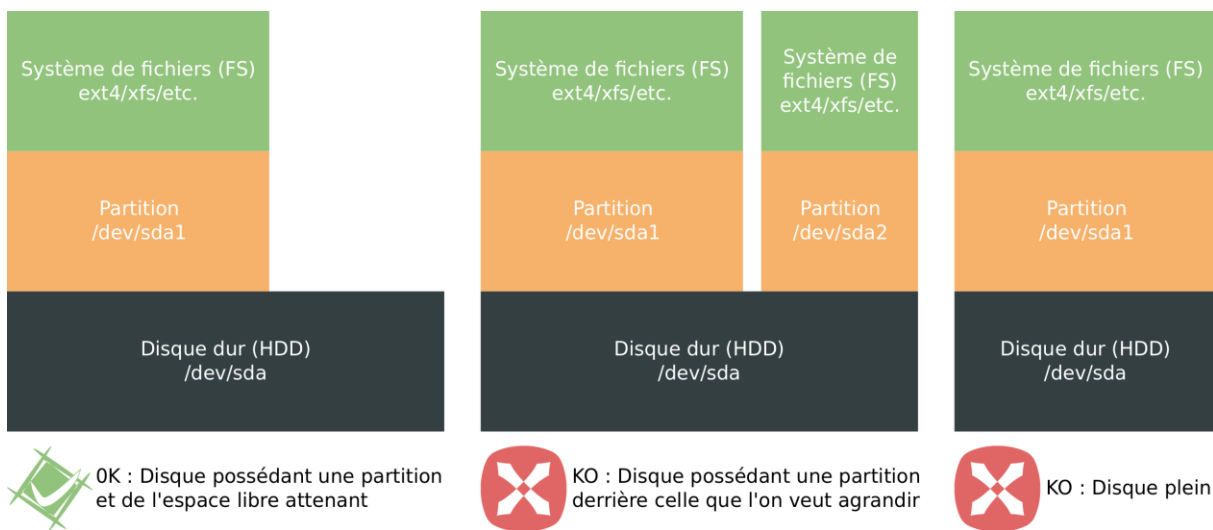


Figure 9 – Cas dans lesquels il est ou non possible d'agrandir une partition

Il n'est pas possible d'agrandir directement une partition, il est nécessaire de la supprimer et d'en créer une nouvelle. Les partitions étant un simple enregistrement dans la table de partitionnement, leur suppression ne fait que les retirer de la table, sans affecter les données. En recréant une partition débutant exactement au même endroit et avec une taille plus importante, les données sont de nouveau accessibles. Les étapes nécessaires :

1. suppression de la partition ;
2. création d'une partition débutant au même endroit et ayant une taille plus importante ;
3. agrandissement du système de fichiers.

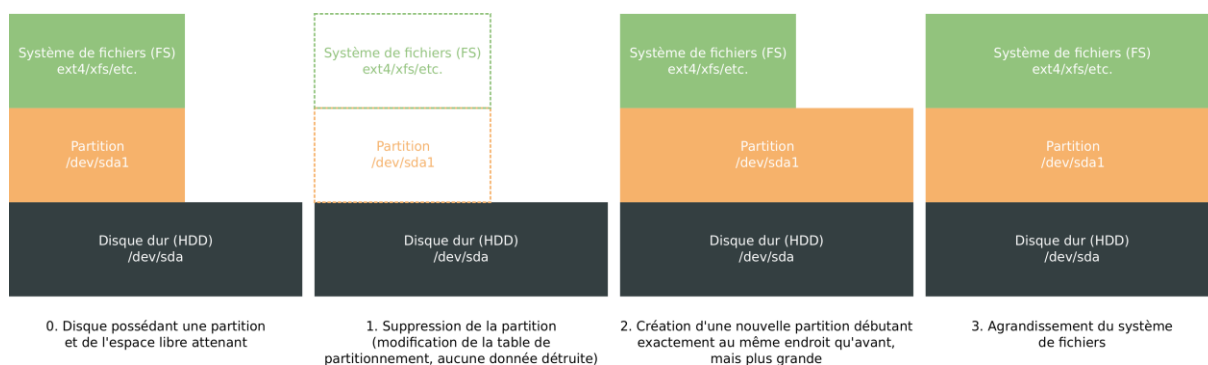


Figure 10 – Agrandissement d'un système de fichiers avec un partitionnement classique

3.2.2 Partitionnement avec LVM

3.2.2.1 Cas où LVM est placé directement sur le disque

L'utilisation de LVM sur un disque dur entier permet de prendre en compte rapidement la modification de la taille du disque, c'est un mécanisme natif :

1. agrandissement de la taille du volume physique (PV) ;
2. agrandissement de la taille du volume logique (LV) ;
3. agrandissement de la taille du système de fichiers.

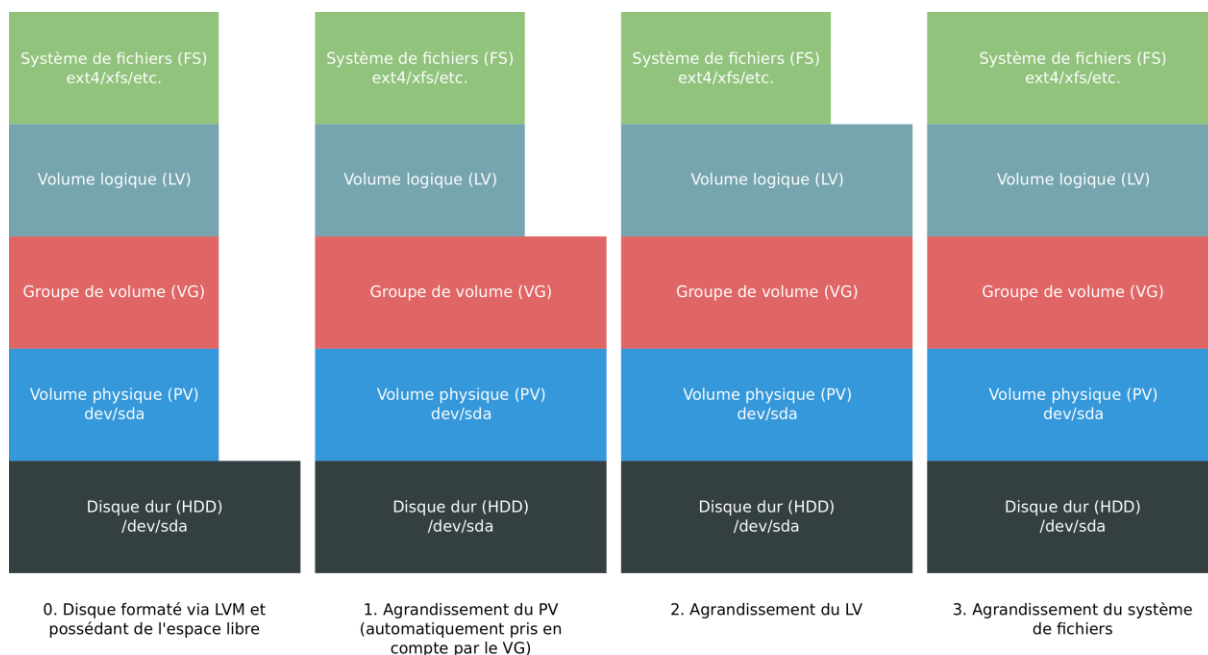


Figure 11 – Agrandissement d'un système de fichiers avec LVM sans partition

Note : Le groupe de volumes (VG) étant simplement une liste de volumes physiques, la taille du VG est toujours égale à la somme des tailles des PV, toute augmentation de volumétrie est directement prise en compte.

3.2.2.2 Cas où LVM est placé sur une partition qu'il est possible d'agrandir

Le cas où LVM est utilisé dans une partition est le plus défavorable, il faut réaliser l'agrandissement de la partition puis de la partie LVM. Dans ce cas, LVM n'apporte rien et au contraire, complexifie les opérations :

1. suppression de la partition ;
2. création d'une partition débutant au même endroit et ayant une taille plus importante ;
3. agrandissement de la taille du volume physique (PV) ;
4. agrandissement de la taille du volume logique (LV) ;
5. agrandissement de la taille du système de fichiers.

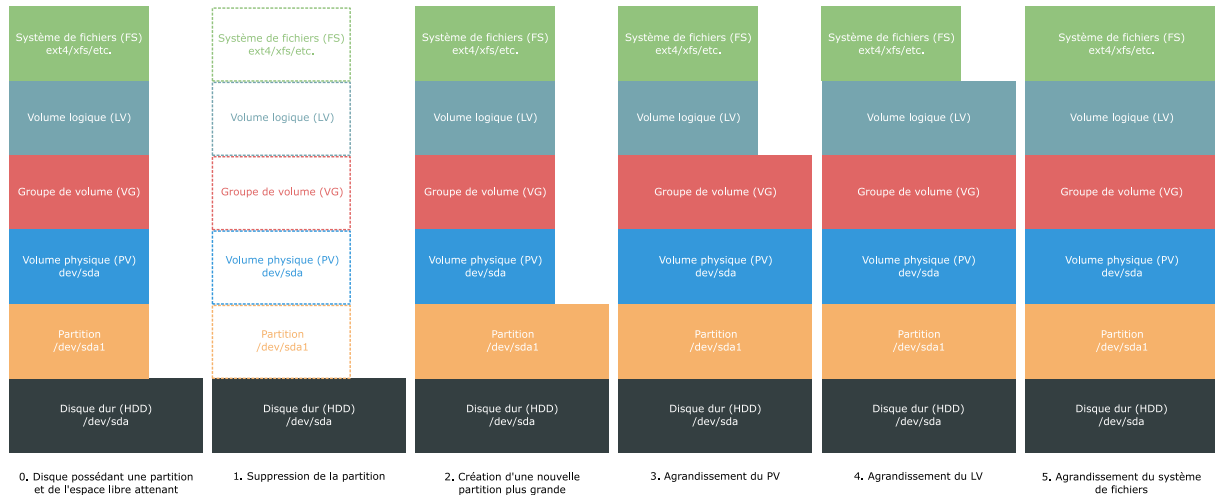


Figure 12 – Agrandissement d'un système de fichiers avec LVM avec partition

3.2.2.3 Cas où LVM est placé sur une partition qu'il n'est pas possible d'agrandir

Une alternative au redimensionnement de la partition sous-jacente à LVM est l'ajout de nouveaux disques ou partitions :

1. création d'une nouvelle partition ou ajout d'un nouveau disque ;
2. création d'un volume physique (PV) sur le nouveau disque ou la nouvelle partition ;
3. ajout du volume physique au groupe de volumes (VG) ;
4. agrandissement de la taille du volume logique (LV) ;
5. agrandissement de la taille du système de fichiers.

S'il n'y a pas d'espace libre en suivant de la partition, il n'est pas possible de l'agrandir. Dans le cas d'un partitionnement classique, cette contrainte est bloquante et empêche tout agrandissement du système de fichiers (sauf en se risquant au déplacement des partitions). Avec LVM, ce n'est pas problématique, s'il y a une partition derrière celle à agrandir, il suffit d'en créer une nouvelle plus loin sur le disque, ou bien d'utiliser un autre disque. Les figures suivantes montrent chacune des deux solutions.

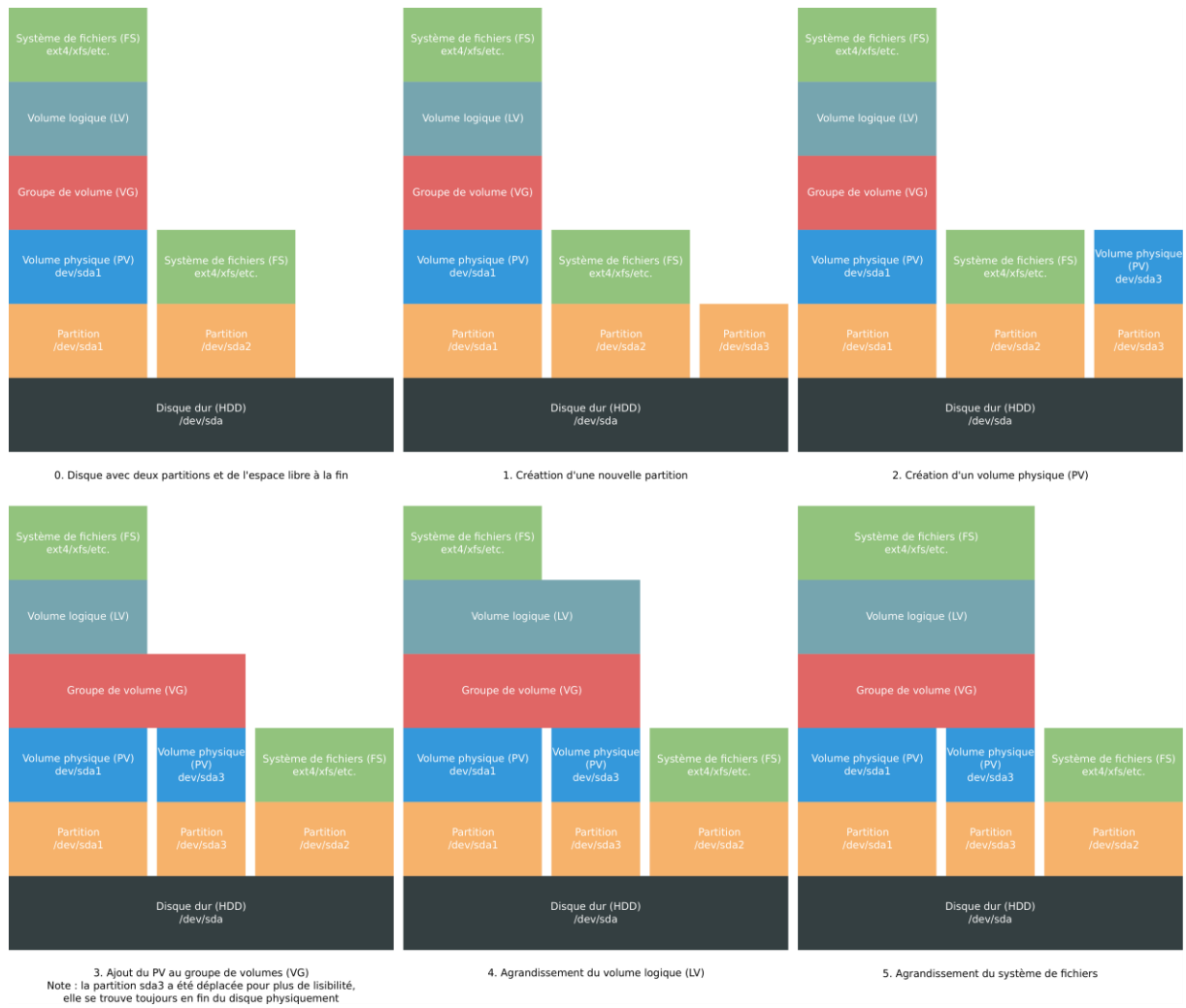


Figure 13 – Agrandissement d'un système de fichiers avec LVM avec une nouvelle partition

Dans la figure ci-dessus, la partition *sda1* se trouve au début du disque et *sda3* à la fin. Elles ont été mises côte à côte sur le schéma pour une meilleure lisibilité.



Figure 14 – Agrandissement d'un système de fichiers avec LVM avec un nouveau disque

4 Cas pratique : création d'un volume LVM et d'un système de fichiers

Plusieurs cas d'utilisation de LVM sont traités dans ce document, le but étant de traiter les situations les plus fréquentes dans la réalité. Pour chaque exemple, un résumé des commandes importantes est donné suivi du détail de la mise en œuvre pas à pas avec les vérifications utiles à chaque étape.

Attention : Les exemples de ce document ont été réalisés à partir d'une machine CentOS 7. Quelques différences peuvent exister selon la distribution et la version des outils. Les commandes sont à adapter à la situation, en ayant une attention particulière dans le choix du disque (*sda*, *sdb*, etc.) et de la partition (*sda1*, *sda2*, etc.) sur laquelle les commandes sont exécutées. En cas d'erreur, une perte de données est possible.

Attention : Toujours avoir une sauvegarde ou un snapshot de la machine avant toute opération.

4.1 Utilisation d'un disque vierge

4.1.1 Description du cas

La machine de l'exemple possède un disque système *sda* de 50 Go et un disque vierge *sdb* de 10 Go. Le disque *sdb* va être configuré pour une utilisation exclusive via LVM, sans table de partitionnement.

4.1.2 Résumé des commandes

1. Créer un nouveau PV : `pvcreate /dev/sdX`
2. Créer un nouveau VG : `vgcreate vgName /dev/sdX`
3. Créer un nouveau LV : `lvcreate -n LvName -l 100%FREE vgName`
4. Créer un système de fichiers :
 - Cas ext4 : `mkfs.ext4 /dev/vgName/LvName`
 - Cas xfs : `mkfs.xfs /dev/vgName/LvName`
5. Configurer le montage automatique au démarrage : `mkdir /mountPoint && echo '/dev/vgName/LvName /mountPoint xfs|ext4 defaults 0 0' >> /etc/fstab`

4.1.3 Détail

1. Afficher les partitions présentes sur le disque avec `lsblk`, le disque *sdb* a une taille de 10 Go et est encore vierge :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1               8:1    0   500M  0 part /boot
└─sda2               8:2    0  49,5G  0 part
   ├─centos-swap     253:0    0    2G   0 lvm  [SWAP]
   └─centos-root     253:1    0  47,5G  0 lvm  /
sdb                  8:16    0   10G  0 disk
sr0                  11:0    1 1024M  0 rom
```

2. Créer un volume physique avec `pvcreate` afin d'indiquer à LVM de gérer le disque *sdb* :

```
root@lucario:~# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created.
```

- Afficher les volumes physiques avec `pvs`, le volume `sdb` apparaît, sa taille est bien égale à celle du disque :

```
root@lucario:~# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sda2   centos  lvm2 a--  <49,51g 44,00m
/dev/sdb    lvm2   ---  10,00g 10,00g
```

- Créer un groupe de volumes nommé `vgDemo` avec `vgcreate` et y inclure le volume physique `sdb` :

```
root@lucario:~# vgcreate vgDemo /dev/sdb
Volume group "vgDemo" successfully created
```

- Afficher les groupes de volume avec `vgs`, le groupe de volumes `vgDemo` apparaît :

```
root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize  VFree
centos   1  2  0 wz--n- <49,51g 44,00m
vgDemo   1  0  0 wz--n- <10,00g <10,00g
```

- Créer un volume logique nommé `lvDemo` avec `lvcreate` et lui indiquer d'utiliser tout l'espace libre sur le groupe de volumes `vgDemo` :

```
root@lucario:~# lvcreate -n lvDemo -l 100%FREE vgDemo
Logical volume "lvDemo" created.
```

- Afficher les volumes logiques avec `lvs`, le volume logique `lvDemo` apparaît avec une taille de 10 Go :

```
root@lucario:~# lvs
LV          VG      Attr              LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sy
nc Convert
root        centos  -wi-ao----- 47,46g
swap        centos  -wi-ao-----  2,00g
lvDemo     vgDemo  -wi-a-----  <10,00g
```

- Vérifier avec `lsblk` que le groupe de volumes et le volume logique créés sont correctement pris en compte :

```
root@lucario:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0  50G  0 disk
├─sda1           8:1    0  500M  0 part /boot
└─sda2           8:2    0  49,5G  0 part
   ├─centos-swap 253:0   0    2G   0 lvm  [SWAP]
   └─centos-root 253:1   0  47,5G  0 lvm  /
sdb              8:16   0   10G  0 disk
└─vgDemo-lvDemo 253:2   0   10G  0 lvm
sr0             11:0    1 1024M  0 rom
```

9. Créer un système de fichiers sur le volume logique `lvDemo`. Adapter la commande `mkfs` au système de fichiers souhaité, ici `xfs` :

```
root@lucario:~# mkfs.xfs /dev/vgDemo/lvDemo
meta-data=/dev/vgDemo/lvDemo      isize=512    agcount=4, agsize=655104 blks
      =                               sectsz=512    attr=2, projid32bit=1
      =                               crc=1        finobt=0, sparse=0
data      =                               bsize=4096  blocks=2620416, imaxpct=25
      =                               sunit=0     swidth=0 blks
naming    =version 2                 bsize=4096  ascii-ci=0 ftype=1
log       =internal log              bsize=4096  blocks=2560, version=2
      =                               sectsz=512  sunit=0 blks, lazy-count=1
realtime  =none                      extsz=4096  blocks=0, rtextents=0
```

Astuce : Le chemin vers un volume logique est toujours de la forme `/dev/<vgName>/<lvName>`.

10. Vérifier avec `lsblk -f` que le système de fichiers est correctement reconnu :

```
root@lucario:~# lsblk -f
NAME            FSTYPE      LABEL UUID                                 MOU
NTPOINT
sda
├─sda1          xfs          81834042-fffe-449e-8f2a-1f840ab70dd9 /bo
ot
└─sda2          LVM2_member AyBns1-sOKo-AYyT-jN1Q-WMvM-aain-kda73w
   ├─centos-swap swap         74fddea9-1591-43a5-b09c-bff997f2198c [SW
AP]
   └─centos-root xfs          353f25d5-45d2-478f-bbe9-b17e322ef707 /
sdb             LVM2_member OQaKo5-Rp3Z-6mtn-LXVR-jNXt-gdk8-ZNyWSR
└─vgDemo-lvDemo xfs          7597716e-b556-4ecd-b045-9fc4f42aed57
sr0
```

11. (Facultatif) Préparer le montage automatique du système de fichiers avec le point de montage `/data` :

- a. Créer le répertoire accueillant le point de montage :

```
root@lucario:~# mkdir /data
```

- b. Configurer le montage automatique dans le fichier `/etc/fstab` :

```
root@lucario:~# echo '/dev/vgDemo/lvDemo /data xfs defaults 0 0' >>
/etc/fstab
```

- c. Monter le système de fichiers avec `mount` et vérifier avec `lsblk` que tout est correct :

```
root@lucario:~# mount -a

root@lucario:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
[...]
```

```
/dev/mapper/vgDemo-lvDemo on /data type xfs (rw,relatime,seclabel,att
r2,inode64,noquota)
```

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0    50G  0 disk
├─sda1                8:1    0   500M  0 part /boot
├─sda2                8:2    0  49,5G  0 part
│   └─centos-swap     253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root     253:1    0  47,5G  0 lvm  /
sdb                  8:16    0    10G  0 disk
└─vgDemo-lvDemo     253:2    0    10G  0 lvm  /data
sr0                 11:0    1 1024M  0 rom
```

4.2 Utilisation d'un disque partitionné

4.2.1 Description du cas

La machine de l'exemple possède un disque de 75 Go contenant plusieurs partitions :

- une partition `sda1` (/boot) de 500 Mo ;
- une partition `sda2` (/) de 49,5 Go ;
- un espace non alloué de 25 Go.

Le fait d'avoir une ou plusieurs partitions ne change en rien la méthode. La machine de test a été configurée avec deux partitions, dont une gérée via LVM. Chaque machine est différente et la configuration de leurs disques aussi. La seule chose importante est d'avoir de l'espace non alloué pour créer une partition supplémentaire.

Dans l'exemple, une troisième partition va être créée sur ce disque, nommée `sda3`.

4.2.2 Résumé des commandes

1. Créer une partition sur le disque :

```
root@lucario:~# fdisk /dev/sdX
n
p
<partNumber>
<startSector>
<endSector>
t
<partNumber>
8e
w
```

2. Scanner les partitions : `partprobe`
3. Créer un nouveau PV : `pvcreate /dev/sdX`
4. Créer un nouveau VG : `vgcreate vgName /dev/sdX`
5. Créer un nouveau LV : `lvcreate -n LvName -l 100%FREE vgName`
6. Créer un système de fichiers :
 - Cas ext4 : `mkfs.ext4 /dev/vgName/LvName`
 - Cas xfs : `mkfs.xfs /dev/vgName/LvName`
7. Configurer le montage automatique au démarrage : `mkdir /mountPoint && echo '/dev/vgName/LvName /mountPoint xfs|ext4 defaults 0 0' >> /etc/fstab`

4.2.3 Détail

1. Afficher les partitions présentes sur le disque avec `lsblk` et noter le numéro de la prochaine qui sera créée, `sda3` dans cet exemple :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   75G  0 disk
├─sda1                8:1    0   500M  0 part /boot
├─sda2                8:2    0  49,5G  0 part
│   └─centos-swap     253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root     253:1    0  47,5G  0 lvm  /
└─sr0                11:0    1 1024M  0 rom
```

2. Créer une nouvelle partition `sda3` de type LVM (8e) avec `fdisk`. L'outil fonctionne de façon interactive, les commandes à entrer sont en couleur :

```
root@lucario:~# fdisk /dev/sda
Bienvenue dans fdisk (util-linux 2.23.2).

Les modifications resteront en mémoire jusqu'à écriture.
Soyez prudent avant d'utiliser la commande d'écriture.

Commande (m pour l'aide) : n
Type de partition :
  p primaire (2 primaire(s), 0 étendue(s), 2 libre(s))
  e étendue
Sélection (p par défaut) : p
Numéro de partition (3,4, 3 par défaut) : 3
Premier secteur (104857600-157286399, 104857600 par défaut) :
Utilisation de la valeur 104857600 par défaut
Dernier secteur, +secteur ou +taille{K,M,G} (104857600-157286399, 157286399
par défaut) :
Utilisation de la valeur 157286399 par défaut
La partition 3 de type Linux et de taille 25 GiB est configurée

Commande (m pour l'aide) : t
Numéro de partition (1-3, 3 par défaut) : 3
Code Hexa (taper L pour afficher tous les codes) :8e
Type de partition « Linux » modifié en « Linux LVM »

Commande (m pour l'aide) : w
La table de partitions a été altérée.

Appel d'ioctl() pour relire la table de partitions.

Attention : la table de partitions n'a pas pu être relue : erreur 16 : Périphérique ou ressource occupé.
Le noyau continue à utiliser l'ancienne table. La nouvelle sera utilisée lors du prochain démarrage ou après avoir exécuté partprobe(8) ou kpartx(8).
Synchronisation des disques.
```

Explications :

- n : nouvelle partition ;
- p : primaire ;
- 3 : numéro de la partition à formater (peut varier) ;

- puis faire attention à bien choisir le premier secteur libre, ce n'est pas toujours le choix par défaut ;
 - t : modifie le type de partition ;
 - 3 : numéro de la partition à modifier, cette étape est sautée s'il n'y en a qu'une ;
 - 8e : partition de type Linux LVM ;
 - w : écrit les changements sur le disque.
3. Scanner les partitions avec `partprobe` pour que celle qui vient d'être créée soit prise en compte :

```
root@lucario:~# partprobe
```

4. Afficher les partitions présentes sur le disque avec `lsblk`, la partition `sda3` est visible et a une taille de 25 Go comme attendu :

```
root@lucario:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0   75G  0 disk
├─sda1            8:1    0  500M  0 part /boot
├─sda2            8:2    0  49,5G  0 part
│   └─centos-swap 253:0    0    2G  0 lvm  [SWAP]
│   └─centos-root 253:1    0  47,5G  0 lvm  /
└─sda3           8:3    0   25G  0 part
sr0             11:0    1 1024M  0 rom
```

5. Créer un volume physique avec `pvcreate` afin d'indiquer à LVM de gérer la partition `sda3` :

```
root@lucario:~# pvcreate /dev/sda3
Physical volume "/dev/sda3" successfully created.
```

6. Afficher les volumes physiques avec `pvs`, le volume `sda3` apparaît, sa taille est bien égale à celle de la partition :

```
root@lucario:~# pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sda2   centos  lvm2 a--   <49,51g 44,00m
/dev/sda3   lvm2   ---   25,00g 25,00g
```

7. Créer un groupe de volumes nommé `vgDemo` avec `vgcreate` et y inclure le volume physique `sda3` :

```
root@lucario:~# vgcreate vgDemo /dev/sda3
Volume group "vgDemo" successfully created
```

8. Afficher les groupes de volume avec `vgs`, le groupe de volumes `vgDemo` apparaît :

```
root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize   VFree
centos  1  2  0 wz--n- <49,51g 44,00m
vgDemo  1  0  0 wz--n- <25,00g <25,00g
```

9. Créer un volume logique nommé *lvDemo* avec `lvcreate`, lui indiquer d'utiliser tout l'espace libre sur le groupe de volumes *vgDemo* :

```
root@lucario:~# lvcreate -n lvDemo -l 100%FREE vgDemo
Logical volume "lvDemo" created.
```

10. Afficher les volumes logiques avec `lvs`, le volume logique *lvDemo* apparaît avec une taille de 25 Go :

```
root@lucario:~# lvs
LV      VG      Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sy
nc Convert
root   centos  -wi-ao----   47,46g
swap   centos  -wi-ao----    2,00g
lvDemo vgDemo  -wi-a----- <25,00g
```

11. Vérifier avec `lsblk` que le groupe de volumes et le volume logique créés sont correctement pris en compte :

```
root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   75G  0 disk
├─sda1                               8:1    0  500M  0 part /boot
├─sda2                               8:2    0  49,5G  0 part
│   ├─centos-swap                    253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root                    253:1    0  47,5G  0 lvm  /
└─sda3                               8:3    0    25G  0 part
    └─vgDemo-lvDemo                  253:2    0    25G  0 lvm
sr0                                  11:0    1 1024M  0 rom
```

12. Créer un système de fichiers sur le volume logique *lvDemo*. Adapter la commande `mkfs` au système de fichiers souhaité, ici `xfs` :

```
root@lucario:~# mkfs.xfs /dev/vgDemo/lvDemo
meta-data=/dev/vgDemo/lvDemo      isize=512    agcount=4, agsize=1638144 blks
      =                               sectsz=512   attr=2, projid32bit=1
      =                               crc=1       finobt=0, sparse=0
data     =                               bsize=4096  blocks=6552576, imaxpct=25
      =                               sunit=0     swidth=0 blks
naming   =version 2                   bsize=4096  ascii-ci=0 ftype=1
log      =internal log                bsize=4096  blocks=3199, version=2
      =                               sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                        extsz=4096  blocks=0, rtextents=0
```

Astuce : Le chemin vers un volume logique est toujours de la forme `/dev/<vgName>/<lvName>`.

13. Vérifier avec `lsblk -f` que le système de fichiers est correctement reconnu :

```
root@lucario:~# lsblk -f
```

NAME	FSTYPE	LABEL	UUID	M
OUNTPOINT				
sda				
├sda1	xf		81834042-ffffe-449e-8f2a-1f840ab70dd9	/
boot				
├sda2	LVM2_member		AyBns1-sOKo-AYyT-jN1Q-WMvM-aain-kda73w	
├centos-swap	swap		74fddea9-1591-43a5-b09c-bff997f2198c	[
SWAP]				
├centos-root	xf		353f25d5-45d2-478f-bbe9-b17e322ef707	/
├sda3	LVM2_member		gEPVvm-tbHh-UGCC-wGMK-GD2S-Xpby-bbfBju	
├vgDemo-lvDemo	xf		f551b4ad-10c9-40c0-9e29-a1c6dee06913	
sr0				

14. (Facultatif) Préparer le montage automatique du système de fichiers avec le point de montage `/data` :

- a. Créer le répertoire accueillant le point de montage :

```
root@lucario:~# mkdir /data
```

- b. Configurer le montage automatique dans le fichier `/etc/fstab` :

```
root@lucario:~# echo '/dev/vgDemo/lvDemo /data xfs defaults 0 0' >> /etc/fstab
```

- c. Monter le système de fichiers avec `mount` et vérifier avec `lsblk` que tout est correct :

```
root@lucario:~# mount -a

root@lucario:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
[...]
/dev/mapper/vgDemo-lvDemo on /data type xfs (rw,relatime,seclabel,attr2,inode64,noquota)

root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0     0   75G  0 disk
├sda1                 8:1     0  500M  0 part /boot
├sda2                 8:2     0  49,5G  0 part
├centos-swap         253:0    0     2G  0 lvm  [SWAP]
├centos-root         253:1    0  47,5G  0 lvm  /
├sda3                 8:3     0    25G  0 part
├vgDemo-lvDemo      253:2    0    25G  0 lvm  /data
sr0                  11:0    1 1024M  0 rom
```


5 Cas pratique : agrandissement d'un système de fichiers

L'agrandissement d'un système de fichiers passe d'abord par l'ajout de ressources à la machine. Cela peut être l'ajout d'un disque pour une machine physique, ou l'agrandissement d'un disque existant pour une machine virtuelle. Le second cas peut être réalisé à chaud. Une fois les nouvelles ressources prises en compte par le système d'exploitation, il est possible d'agrandir le système de fichiers.

Attention : Les exemples de ce document ont été réalisés à partir d'une machine CentOS 7. Quelques différences peuvent exister selon la distribution et la version des outils. Les commandes sont à adapter à la situation, en ayant une attention particulière dans le choix du disque (*sda*, *sdb*, etc.) et de la partition (*sda1*, *sda2*, etc.) sur laquelle les commandes sont exécutées. En cas d'erreur, une perte de données est possible.

Attention : Toujours avoir une sauvegarde ou un snapshot de la machine avant toute opération. Si un snapshot de la machine virtuelle est pris en prévention des opérations de redimensionnement, il doit être créé **après** l'augmentation de la taille du disque. Dans le cas contraire, le snapshot sera inutilisable. Cette contrainte peut aussi s'appliquer aux sauvegardes, selon l'outil et la méthode utilisés.

5.1 Ajout de ressources de stockage à une machine virtuelle

5.1.1 Agrandissement d'un disque

Dans l'exemple la machine possède un disque *sda* de 50 Go dont la taille va être augmentée à 75 Go.

1. Depuis l'outil de gestion des machines virtuelles, reconfigurer la machine pour augmenter la taille d'un disque existant.
2. Pour que la modification matérielle soit prise en compte sans redémarrer la machine, indiquer au système de scanner les disques physiques.
 - a. Vérifier la taille des disques vue par le système avant le scan, avec la commande `lsblk`. Dans l'exemple, le disque *sda* a une capacité de 50 Go :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0   500M  0 part /boot
└─sda2                8:2    0  49,5G  0 part
   ├─centos-swap      253:0   0     2G  0 lvm  [SWAP]
   └─centos-root      253:1   0  47,5G  0 lvm  /
sr0                  11:0    1 1024M  0 rom
```

- b. Lancer un scan du disque qui a été redimensionné (*sda* dans l'exemple) :

```
root@lucario:~# echo 1 > /sys/class/block/sda/device/rescan
```

- c. Vérifier la nouvelle taille des disques vue par le système, dans l'exemple la nouvelle taille de *sda* est de 75 Go :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
```

```

sda                8:0    0   75G   0 disk
├─sda1             8:1    0  500M   0 part /boot
├─sda2             8:2    0 49,5G   0 part
│   ├─centos-swap 253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root 253:1    0 47,5G   0 lvm  /
sr0                11:0    1 1024M   0 rom

```

5.1.2 Ajout d'un nouveau disque

Dans l'exemple la machine possède initialement un disque *sda* de 50 Go et un second disque *sdb* de 25 Go va être ajouté.

1. Depuis l'outil de gestion des machines virtuelles, reconfigurer la machine pour ajouter un nouveau disque.
2. Pour que la modification matérielle soit prise en compte sans redémarrer la machine, indiquer au système de scanner les disques physiques.
 - a. Vérifier la taille des disques vue par le système avant le scan, avec la commande `lsblk`. Dans l'exemple, seul le disque *sda* d'une capacité de 50 Go est présent :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                 8:0    0   50G   0 disk
├─sda1             8:1    0  500M   0 part /boot
├─sda2             8:2    0 49,5G   0 part
│   ├─centos-swap 253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root 253:1    0 47,5G   0 lvm  /
sr0                11:0    1 1024M   0 rom

```

- b. Rechercher le numéro de bus de l'hôte, « host0 » dans l'exemple :

```

root@lucario:~# grep mpt /sys/class/scsi_host/host?/proc_name
/sys/class/scsi_host/host0/proc_name:mptspi

```

- c. Lancer un scan du bus correspondant (les tirets représentent : contrôleur, canal, lun ; donc « - - - » signifie que tous les contrôleurs, canaux et luns doivent être scannés) :

```

root@lucario:~# echo "- - -" > /sys/class/scsi_host/host0/scan

```

- d. Vérifier la nouvelle taille des disques vue par le système, dans l'exemple le disque *sdb* d'une taille de 25 Go a été ajouté :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                 8:0    0   50G   0 disk
├─sda1             8:1    0  500M   0 part /boot
├─sda2             8:2    0 49,5G   0 part
│   ├─centos-swap 253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root 253:1    0 47,5G   0 lvm  /
sdb                 8:16    0   25G   0 disk
sr0                11:0    1 1024M   0 rom

```

5.2 Agrandissement d'une partition classique

5.2.1 Description du cas

La machine de l'exemple possède un disque système *sda* de 50 Go et un disque secondaire *sdb* de 100 Go contenant une partition unique *sdb1* formatée en xfs.

Le disque *sdb* a été agrandi de 50 Go, agrandissement qui va être répercuté sur la partition et le système de fichiers.

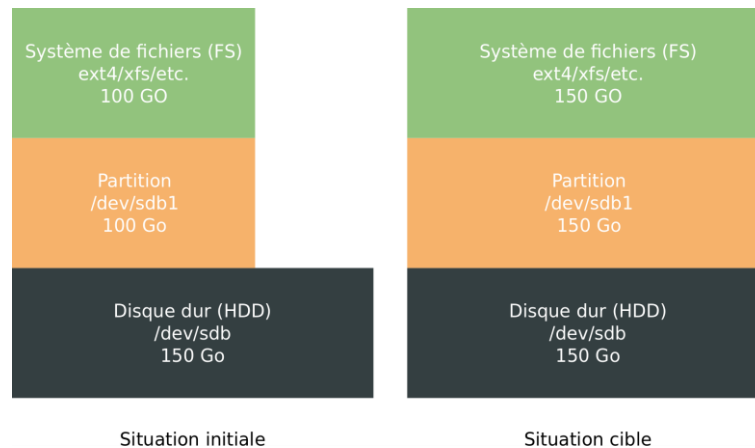


Figure 15 – Agrandissement d'une partition avant/après

Note : Pour agrandir une partition, il est nécessaire d'avoir de l'espace libre attenante sur le disque.

5.2.2 Résumé des commandes

1. Supprimer la partition à agrandir et créer une nouvelle partition débutant au même endroit et ayant une taille plus importante :

```
root@lucario:~# fdisk /dev/sdX
d
<partNumber>
n
p
<partNumber>
<startSector>
<endSector>
w
```

2. Agrandir le système de fichiers :

- Cas ext4 : `resize2fs /dev/sdX`
- Cas xfs : `xfs_growfs /dev/sdX`

5.2.3 Détail

1. Afficher les disques et partitions présents sur la machine avec `lsblk`, le disque *sdb* a une taille de 150 Go et contient une partition *sdb1* de 100 Go, qui va être agrandie :

```
root@lucario:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0   50G  0 disk
```

```

├─sda1      8:1    0  500M  0 part /boot
├─sda2      8:2    0 49,5G  0 part
├─centos-swap 253:0   0   2G  0 lvm [SWAP]
└─centos-root 253:1   0 47,5G  0 lvm /
sdb         8:16    0 150G  0 disk
├─sdb1      8:17    0 100G  0 part /data
sr0         11:0    1 1024M  0 rom

```

2. Agrandir la partition `sdb1` avec `fdisk`. L'outil fonctionne de façon interactive, les commandes à entrer sont en couleur :

Attention : Cette étape consiste à supprimer puis recréer la partition. Une erreur peut entraîner une perte de données et potentiellement empêcher le système de redémarrer (si la partition système est modifiée).

L'exemple est donné dans le cas le plus simple : un disque avec une partition seulement. Si d'autres partitions sont présentes, il faut faire attention à bien choisir le **numéro de la partition, le premier secteur et le dernier secteur**.

```

root@lucario:~# fdisk /dev/sdb
Bienvenue dans fdisk (util-linux 2.23.2).

Les modifications resteront en mémoire jusqu'à écriture.
Soyez prudent avant d'utiliser la commande d'écriture.

Commande (m pour l'aide) : d
Partition 1 sélectionnée
La partition 1 est supprimée

Commande (m pour l'aide) : n
Type de partition :
  p  primaire (0 primaire(s), 0 étendue(s), 4 libre(s))
  e  étendue
Sélection (p par défaut) : p
Numéro de partition (1-4, 1 par défaut) : 1
Premier secteur (2048-314572799, 2048 par défaut) :
Utilisation de la valeur 2048 par défaut
Dernier secteur, +secteur ou +taille{K,M,G} (2048-314572799, 314572799 par d
éfaut) :
Utilisation de la valeur 314572799 par défaut
La partition 1 de type Linux et de taille 150 GiB est configurée

Commande (m pour l'aide) : w
La table de partitions a été altérée.

Appel d'ioctl() pour relire la table de partitions.

Attention : la table de partitions n'a pas pu être relue : erreur 16 : Périph
érique ou ressource occupé.
Le noyau continue à utiliser l'ancienne table. La nouvelle sera utilisée
lors du prochain démarrage ou après avoir exécuté partprobe(8) ou kpartx(8).
Synchronisation des disques.

```

Explications :

- d : supprime la partition, s'il y en a plusieurs, le numéro de la partition est demandé ;
 - n : nouvelle partition ;
 - p : primaire ;
 - 1 : numéro de la partition à formater (peut varier) ;
 - premier secteur : reprendre celui de la partition supprimée ;
 - dernier secteur : indiquer la nouvelle valeur. Si la partition est la dernière sur le disque, il s'agit du plus grand secteur proposé. S'il y a une partition derrière, il faut choisir le secteur précédent le premier secteur de la partition suivante ;
 - w : écrit les changements sur le disque.
3. Scanner les partitions avec `partprobe` pour que les modifications sur `sdb1` soient prises en compte :

```
root@lucario:~# partprobe
```

4. Afficher les disques et partitions présents sur la machine avec `lsblk`, la taille de la partition `sdb1` est maintenant de 150 Go :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0  500M  0 part /boot
├─sda2                8:2    0 49,5G  0 part
│   └─centos-swap     253:0    0    2G  0 lvm  [SWAP]
│   └─centos-root     253:1    0 47,5G  0 lvm  /
sdb                  8:16    0  150G  0 disk
└─sdb1                8:17    0  150G  0 part /data
sr0                  11:0    1 1024M  0 rom
```

5. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers de `sdb1` a toujours une taille de 100 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,1G   46G   5% /
devtmpfs                908M     0   908M   0% /dev
tmpfs                   920M     0   920M   0% /dev/shm
tmpfs                   920M   8,8M   911M   1% /run
tmpfs                   920M     0   920M   0% /sys/fs/cgroup
/dev/sdb1               100G   33M  100G   1% /data
/dev/sda1               497M  251M  246M  51% /boot
tmpfs                   184M     0   184M   0% /run/user/0
```

6. Agrandir le système de fichiers de `sdb1` :

- Pour xfs :

```
root@lucario:~# xfs_growfs /dev/sdb1
meta-data=/dev/sdb1          isize=512    agcount=4, agsize=6553536
blks
      =                       sectsz=512   attr=2, projid32bit=1
      =                       crc=1      finobt=0 spinodes=0
```

```

data      =                               bsize=4096  blocks=26214144, imaxpct=2
5
          =                               sunit=0    swidth=0 blks
naming    =version 2                       bsize=4096  ascii-ci=0  ftype=1
log       =internal                         bsize=4096  blocks=12799, version=2
          =                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                             extsz=4096  blocks=0,  rtextents=0
data blocks changed from 26214144 to 39321344

```

- Pour ext4 :

```
root@lucario:~# resize2fs /dev/sdb1
```

7. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers de `sdb1` a maintenant une taille de 150 Go :

```

root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,1G   46G   5% /
devtmpfs              908M   0   908M   0% /dev
tmpfs                 920M   0   920M   0% /dev/shm
tmpfs                 920M   8,8M   911M   1% /run
tmpfs                 920M   0   920M   0% /sys/fs/cgroup
/dev/sdb1             150G   33M  150G   1% /data
/dev/sda1             497M  251M  246M  51% /boot
tmpfs                 184M   0   184M   0% /run/user/0

```

5.3 Agrandissement d'un volume logique

La création de volumes LVM peut se faire soit sur une partition, soit sur un disque entier. Selon le cas, la procédure d'agrandissement du volume diffère. La commande `lsblk` permet de déterminer la façon dont les disques ont été configurés :

```

root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   55G  0 disk
├─sda1                               8:1    0   476M  0 part /boot
└─sda2                               8:2    0  49,5G  0 part
   ├─vgRoot-lvRoot                   249:0    0   13G  0 lvm /
   ├─vgRoot-lvSwap                   249:1    0    8G  0 lvm [SWAP]
   ├─vgRoot-lvOracleProduct          249:2    0   20G  0 lvm /oracle/product
   └─vgRoot-lvADR                    249:3    0    5G  0 lvm /oracle/diag
sdb                                  8:16    0   18G  0 disk
└─vgOradata-lvOradata               249:4    0   12G  0 lvm /oradata
sdc                                  8:32    0    6G  0 disk
└─vgOrafra-lvOrafra                 249:5    0    6G  0 lvm /oralog
sr0                                  11:0    1 1024M  0 rom

```

Dans l'exemple ci-dessus, le disque `sda` est le disque système a été partitionné de la façon suivante :

- une partition `sda1` utilisée pour le démarrage du système (`/boot`) ;
- une partition `sda2` utilisée par LVM et contenant le volume principal (`/`), celui de swap et d'autres volumes.

Le disque système est toujours partitionné. En revanche, les deux autres disques *sdb* et *sdc* n'ont aucune partition.

Le cas le plus simple est celui où le volume LVM est créé directement au niveau du disque, le volume peut alors être directement agrandi.

Quand un disque a été partitionné (même avec une seule partition), certaines précautions sont nécessaires pour augmenter la taille des volumes LVM. La première étape consiste en l'agrandissement de la partition sous-jacente ou en la création d'une nouvelle partition.

5.3.1 Cas n°1 – Disque sans partition

5.3.1.1 Description du cas

La machine de l'exemple possède un disque système *sda* de 50 Go et un disque secondaire *sdb* de 100 Go géré par LVM et contenant un système de fichiers xfs de 50 Go.

Le disque *sdb* a été agrandi de 50 Go, agrandissement qui va être répercuté sur le système de fichiers.

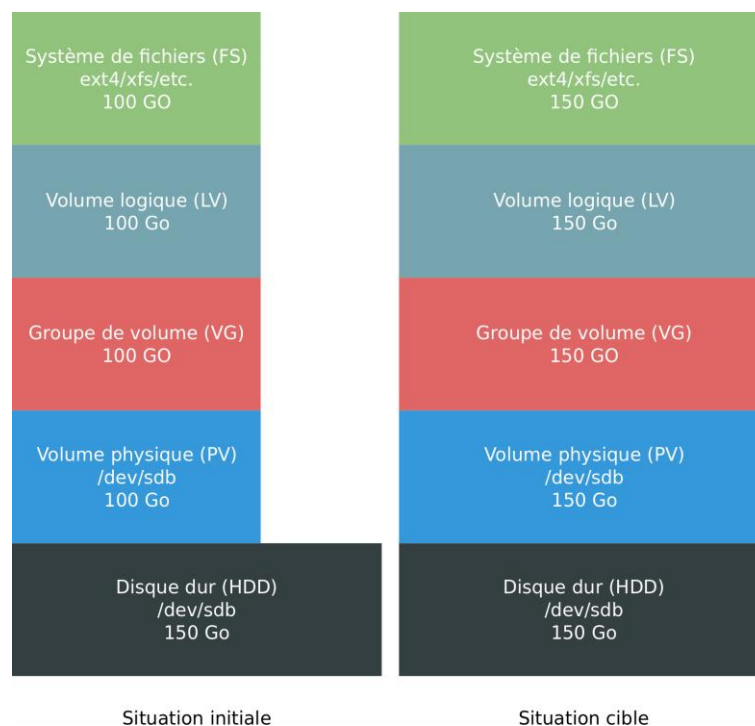


Figure 16 – Agrandissement d'un système de fichiers sur LVM avant/après

5.3.1.2 Résumé des commandes

1. Agrandir le PV : `pvresize /dev/sdX`
2. Agrandir le LV : `lvextend /dev/vgName/LvName /dev/sdX`
3. Agrandir le FS :
 - Cas ext4 : `resize2fs /dev/vgName/LvName`
 - Cas xfs : `xfs_growfs /dev/vgName/LvName`

5.3.1.3 Détail

1. Afficher les disques et partitions présents sur la machine avec `lsblk`, le disque *sdb* a une taille de 150 Go et contient un volume LVM de 100 Go, qui va être agrandi :

```
root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
```

```

sda                8:0    0   50G   0 disk
├─sda1              8:1    0   500M  0 part /boot
├─sda2              8:2    0  49,5G  0 part
│   ├─centos-swap   253:0    0    2G   0 lvm  [SWAP]
│   └─centos-root   253:1    0  47,5G  0 lvm  /
sdb                 8:16    0  150G   0 disk
└─vgDemo-lvDemo    253:2    0  100G   0 lvm  /data
sr0                 11:0    1 1024M  0 rom

```

- Afficher les volumes physiques avec `pvs`, le volume `sdb` a une taille de 100 Go avant l'opération :

```

root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2   centos lvm2 a--  <49,51g 44,00m
/dev/sdb    vgDemo lvm2 a--  <100,00g  0

```

- Agrandir le volume physique avec `pvresize`, il est agrandi de façon à occuper l'ensemble de l'espace disponible sur le disque :

```

root@lucario:~# pvresize /dev/sdb
Physical volume "/dev/sdb" changed
1 physical volume(s) resized or updated / 0 physical volume(s) not resized

```

- Afficher les volumes physiques avec `pvs`, le volume `sdb` a une taille de 150 Go, dont 50 Go de libres :

```

root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2   centos lvm2 a--  <49,51g 44,00m
/dev/sdb    vgDemo lvm2 a--  <150,00g 50,00g

```

L'augmentation de la taille est directement prise en compte au niveau du groupe de volumes associé. Cela peut se vérifier en listant les volumes avec `vgs` :

```

root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize   VFree
centos  1  2  0 wz--n- <49,51g 44,00m
vgDemo  1  1  0 wz--n- <150,00g 50,00g

```

- Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a une taille de 100 Go avant son agrandissement :

```

root@lucario:~# lvs
LV      VG      Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sy
nc Convert
root    centos -wi-ao----   47,46g
swap    centos -wi-ao----    2,00g
lvDemo  vgDemo -wi-ao----  <100,00g

```

- Agrandir le volume logique avec `lvextend`. Il suffit d'indiquer à LVM le nom du volume logique à agrandir et le volume physique sur lequel réaliser l'extension. Par défaut tout l'espace disponible sur le volume physique est alloué :


```

root@lucario:~# lvextend /dev/vgDemo/lvDemo /dev/sdb
Size of logical volume vgDemo/lvDemo changed from <100,00 GiB (25599 extent
s) to <150,00 GiB (38399 extents).
Logical volume vgDemo/lvDemo successfully resized.

```

7. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a désormais une taille de 150 Go :

```

root@lucario:~# lvs
LV      VG      Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sy
nc Convert
root    centos -wi-ao----   47,46g
swap    centos -wi-ao----    2,00g
lvDemo  vgDemo -wi-ao----  <150,00g

```

8. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume `lvDemo` a toujours une taille de 100 Go :

```

root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,2G   46G   5% /
devtmpfs                908M         0  908M   0% /dev
tmpfs                   920M         0  920M   0% /dev/shm
tmpfs                   920M     8,8M  911M   1% /run
tmpfs                   920M         0  920M   0% /sys/fs/cgroup
/dev/mapper/vgDemo-lvDemo 100G     33M  100G   1% /data
/dev/sda1               497M    251M  246M  51% /boot
tmpfs                   184M         0  184M   0% /run/user/0

```

9. Agrandir le système de fichiers de `lvDemo` :

- Pour xfs :

```

root@lucario:~# xfs_growfs /dev/vgDemo/lvDemo
meta-data=/dev/mapper/vgDemo-lvDemo isize=512    agcount=4, agsize=65533
44 blks
         =                               sectsz=512   attr=2, projid32bit=1
         =                               crc=1       finobt=0 spinodes=0
data     =                               bsize=4096  blocks=26213376, imaxpct=2
5
         =                               sunit=0    swidth=0 blks
naming   =version 2                       bsize=4096  ascii-ci=0 ftype=1
log      =internal                        bsize=4096  blocks=12799, version=2
         =                               sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                             extsz=4096  blocks=0, rtextents=0
data blocks changed from 26213376 to 39320576

```

- Pour ext4 :

```

root@lucario:~# resize2fs /dev/vgDemo/lvDemo

```

10. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume `lvDemo` a maintenant une taille de 150 Go :

```

root@lucario:~# df -h

```

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
/dev/mapper/centos-root	48G	2,2G	46G	5%	/
devtmpfs	908M	0	908M	0%	/dev
tmpfs	920M	0	920M	0%	/dev/shm
tmpfs	920M	8,8M	911M	1%	/run
tmpfs	920M	0	920M	0%	/sys/fs/cgroup
/dev/mapper/vgDemo-lvDemo	150G	33M	150G	1%	/data
/dev/sda1	497M	251M	246M	51%	/boot
tmpfs	184M	0	184M	0%	/run/user/0

11. Effectuer une vérification complémentaire avec `lsblk` :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1               8:1    0   500M  0 part /boot
└─sda2               8:2    0  49,5G  0 part
   ├─centos-swap     253:0    0    2G   0 lvm  [SWAP]
   └─centos-root     253:1    0  47,5G  0 lvm  /
sdb                  8:16    0  150G  0 disk
└─vgDemo-lvDemo     253:2    0  150G  0 lvm  /data
sr0                  11:0    1 1024M  0 rom

```

5.3.2 Cas n°2 – Disque avec partition extensible

5.3.2.1 Description du cas

La machine de l'exemple possède un disque système `sda` de 50 Go et un disque secondaire `sdb` de 100 Go contenant une partition unique `sdb1` utilisée comme volume LVM formaté en xfs.

Le disque `sdb` a été agrandi de 50 Go, agrandissement qui va être répercuté sur la partition et le système de fichiers.



Figure 17 – Agrandissement d'une partition et d'un volume LVM avant/après

5.3.2.2 Résumé des commandes

1. Supprimer la partition à agrandir et créer une nouvelle partition débutant au même endroit et ayant une taille plus importante :

```
root@lucario:~# fdisk /dev/sdX
d
<partNumber>
n
p
<partNumber>
<startSector>
<endSector>
t
<partNumber>
8e
w
```

2. Scanner les partitions : `partprobe`
3. Agrandir le PV : `pvresize /dev/sdX`
4. Agrandir le LV : `lvextend /dev/vgName/LvName /dev/sdX`
5. Agrandir le FS :
 - Cas ext4 : `resize2fs /dev/vgName/LvName`
 - Cas xfs : `xfs_growfs /dev/vgName/LvName`

5.3.2.3 Détail

1. Afficher les disques et partitions présents sur la machine avec `lsblk`, le disque `sdb` a une taille de 150 Go et contient une partition `sdb1` de 100 Go, qui va être agrandie :

```
root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   50G  0 disk
├─sda1                               8:1    0  500M  0 part /boot
├─sda2                               8:2    0 49,5G  0 part
│   ├─centos-swap                    253:0    0    2G  0 lvm  [SWAP]
│   └─centos-root                    253:1    0 47,5G  0 lvm  /
sdb                                  8:16    0  150G  0 disk
├─sdb1                               8:17    0  100G  0 part
│   └─vgDemo-lvDemo                 253:2    0  100G  0 lvm  /data
sr0                                  11:0    1 1024M  0 rom
```

2. Agrandir la partition `sdb1` avec `fdisk`. L'outil fonctionne de façon interactive, les commandes à entrer sont en couleur :

Attention : Cette étape consiste à supprimer puis recréer la partition. Une erreur peut entraîner une perte de données et potentiellement empêcher le système de redémarrer (si la partition système est modifiée).

L'exemple est donné dans le cas le plus simple : un disque avec une partition seulement. Si d'autres partitions sont présentes, il faut faire attention à bien choisir le **numéro de la partition, le premier secteur et le dernier secteur**.

```
root@lucario:~# fdisk /dev/sdb
Bienvenue dans fdisk (util-linux 2.23.2).

Les modifications resteront en mémoire jusqu'à écriture.
Soyez prudent avant d'utiliser la commande d'écriture.

Commande (m pour l'aide) : d
Partition 1 sélectionnée
La partition 1 est supprimée

Commande (m pour l'aide) : n
Type de partition :
  p  primaire (0 primaire(s), 0 étendue(s), 4 libre(s))
  e  étendue
Sélection (p par défaut) : p
Numéro de partition (1-4, 1 par défaut) :
Premier secteur (2048-314572799, 2048 par défaut) :
Utilisation de la valeur 2048 par défaut
Dernier secteur, +secteur ou +taille{K,M,G} (2048-314572799, 314572799 par d
éfaut) :
Utilisation de la valeur 314572799 par défaut
La partition 1 de type Linux et de taille 150 GiB est configurée

Commande (m pour l'aide) : t
Partition 1 sélectionnée
```

```
Code Hexa ( taper L pour afficher tous les codes ) :8e
Type de partition « Linux » modifié en « Linux LVM »
```

```
Commande ( m pour l'aide ) : w
La table de partitions a été altérée.
```

Appel d'ioctl() pour relire la table de partitions.

Attention : la table de partitions n'a pas pu être relue : erreur 16 : Périphérique ou ressource occupé.
Le noyau continue à utiliser l'ancienne table. La nouvelle sera utilisée lors du prochain démarrage ou après avoir exécuté `partprobe(8)` ou `kpartx(8)`. Synchronisation des disques.

Explications :

- d : supprime la partition, s'il y en a plusieurs, le numéro de la partition est demandé ;
 - n : nouvelle partition ;
 - p : primaire ;
 - 1 : numéro de la partition à formater (peut varier) ;
 - premier secteur : reprendre celui de la partition supprimée ;
 - dernier secteur : indiquer la nouvelle valeur. Si la partition est la dernière sur le disque, il s'agit du plus grand secteur proposé. S'il y a une partition derrière, il faut choisir le secteur précédent le premier secteur de la partition suivante ;
 - t : modifie le type de partition ;
 - 1 : numéro de la partition à modifier, cette étape est sautée s'il n'y en a qu'une ;
 - 8e : partition de type Linux LVM ;
 - w : écrit les changements sur le disque.
3. Scanner les partitions avec `partprobe` pour que les modifications sur `sdb1` soient prises en compte :

```
root@lucario:~# partprobe
```

4. Afficher les disques et partitions présents sur la machine avec `lsblk`, la taille de la partition `sdb1` est maintenant de 150 Go et le volume LVM n'a pas été modifié :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0  500M  0 part /boot
├─sda2                8:2    0 49,5G  0 part
├─centos-swap        253:0   0    2G   0 lvm  [SWAP]
└─centos-root        253:1   0 47,5G  0 lvm  /
sdb                  8:16   0  150G  0 disk
├─sdb1                8:17   0  150G  0 part
└─vgDemo-lvDemo      253:2   0  100G  0 lvm  /data
sr0                  11:0    1 1024M  0 rom
```

5. Afficher les volumes physiques avec `pvs`, le volume `sdb` a une taille de 100 Go avant l'opération :

```
root@lucario:~# pvs
```

PV	VG	Fmt	Attr	PSize	PFree
/dev/sda2	centos	lvm2	a--	<49,51g	44,00m
/dev/sdb1	vgDemo	lvm2	a--	<100,00g	0

6. Agrandir le volume physique avec `pvresize`, il est agrandi de façon à occuper l'ensemble de l'espace disponible sur la partition :

```
root@lucario:~# pvresize /dev/sdb1
Physical volume "/dev/sdb1" changed
1 physical volume(s) resized or updated / 0 physical volume(s) not resized
```

7. Afficher les volumes physiques avec `pvs`. Le volume `sdb1` a une taille de 150 Go, dont 50 Go de libres :

PV	VG	Fmt	Attr	PSize	PFree
/dev/sda2	centos	lvm2	a--	<49,51g	44,00m
/dev/sdb1	vgDemo	lvm2	a--	<150,00g	50,00g

L'augmentation de la taille est directement prise en compte au niveau du groupe de volumes associé. Cela peut se vérifier en listant les volumes avec `vgs` :

VG	#PV	#LV	#SN	Attr	VSize	VFree
centos	1	2	0	wz--n-	<49,51g	44,00m
vgDemo	1	1	0	wz--n-	<150,00g	50,00g

8. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a une taille de 100 Go avant son agrandissement :

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sy
root	centos	-wi-ao----	47,46g								
swap	centos	-wi-ao----	2,00g								
lvDemo	vgDemo	-wi-ao----	<100,00g								

9. Agrandir le volume logique avec `lvextend`. Il suffit d'indiquer à LVM le nom du volume logique à agrandir et le volume physique sur lequel réaliser l'extension. Par défaut tout l'espace disponible sur le volume physique est alloué :

```
root@lucario:~# lvextend /dev/vgDemo/lvDemo /dev/sdb1
Size of logical volume vgDemo/lvDemo changed from <100,00 GiB (25599 extents) to <150,00 GiB (38399 extents).
Logical volume vgDemo/lvDemo successfully resized.
```

10. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a désormais une taille de 150 Go :

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sy
root	centos	-wi-ao----	47,46g								
swap	centos	-wi-ao----	2,00g								

```
lvDemo vgDemo -wi-ao---- <150,00g
```

11. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume *lvDemo* a toujours une taille de 100 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,2G   46G   5% /
devtmpfs                908M     0   908M   0% /dev
tmpfs                   920M     0   920M   0% /dev/shm
tmpfs                   920M   8,8M   911M   1% /run
tmpfs                   920M     0   920M   0% /sys/fs/cgroup
/dev/mapper/vgDemo-lvDemo 100G   33M  100G   1% /data
/dev/sda1               497M  251M  246M  51% /boot
tmpfs                   184M     0   184M   0% /run/user/0
```

12. Agrandir le système de fichiers de *lvDemo* :

- Pour xfs :

```
root@lucario:~# xfs_growfs /dev/vgDemo/lvDemo
meta-data=/dev/mapper/vgDemo-lvDemo isize=512    agcount=4, agsize=65533
44 blks
         =                       sectsz=512    attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096  blocks=26213376, imaxpct=2
5
         =                       sunit=0      swidth=0 blks
naming   =version 2             bsize=4096  ascii-ci=0 ftype=1
log      =internal             bsize=4096  blocks=12799, version=2
         =                       sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                 extsz=4096  blocks=0, rtextents=0
data blocks changed from 26213376 to 39320576
```

- Pour ext4 :

```
root@lucario:~# resize2fs /dev/vgDemo/lvDemo
```

13. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume *lvDemo* a maintenant une taille de 150 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,2G   46G   5% /
devtmpfs                908M     0   908M   0% /dev
tmpfs                   920M     0   920M   0% /dev/shm
tmpfs                   920M   8,8M   911M   1% /run
tmpfs                   920M     0   920M   0% /sys/fs/cgroup
/dev/mapper/vgDemo-lvDemo 150G   33M  150G   1% /data
/dev/sda1               497M  251M  246M  51% /boot
tmpfs                   184M     0   184M   0% /run/user/0
```

14. Effectuer une vérification complémentaire avec `lsblk` :

```
root@lucario:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	50G	0	disk	
└sda1	8:1	0	500M	0	part	/boot
└sda2	8:2	0	49,5G	0	part	
└centos-swap	253:0	0	2G	0	lvm	[SWAP]
└centos-root	253:1	0	47,5G	0	lvm	/
sdb	8:16	0	150G	0	disk	
└sdb1	8:17	0	150G	0	part	
└vgDemo-lvDemo	253:2	0	150G	0	lvm	/data
sr0	11:0	1	1024M	0	rom	

5.3.3 Cas n°3 – Disque avec partition étendue extensible

5.3.3.1 Description du cas

La machine de l'exemple possède un disque de 100 Go contenant quatre partitions :

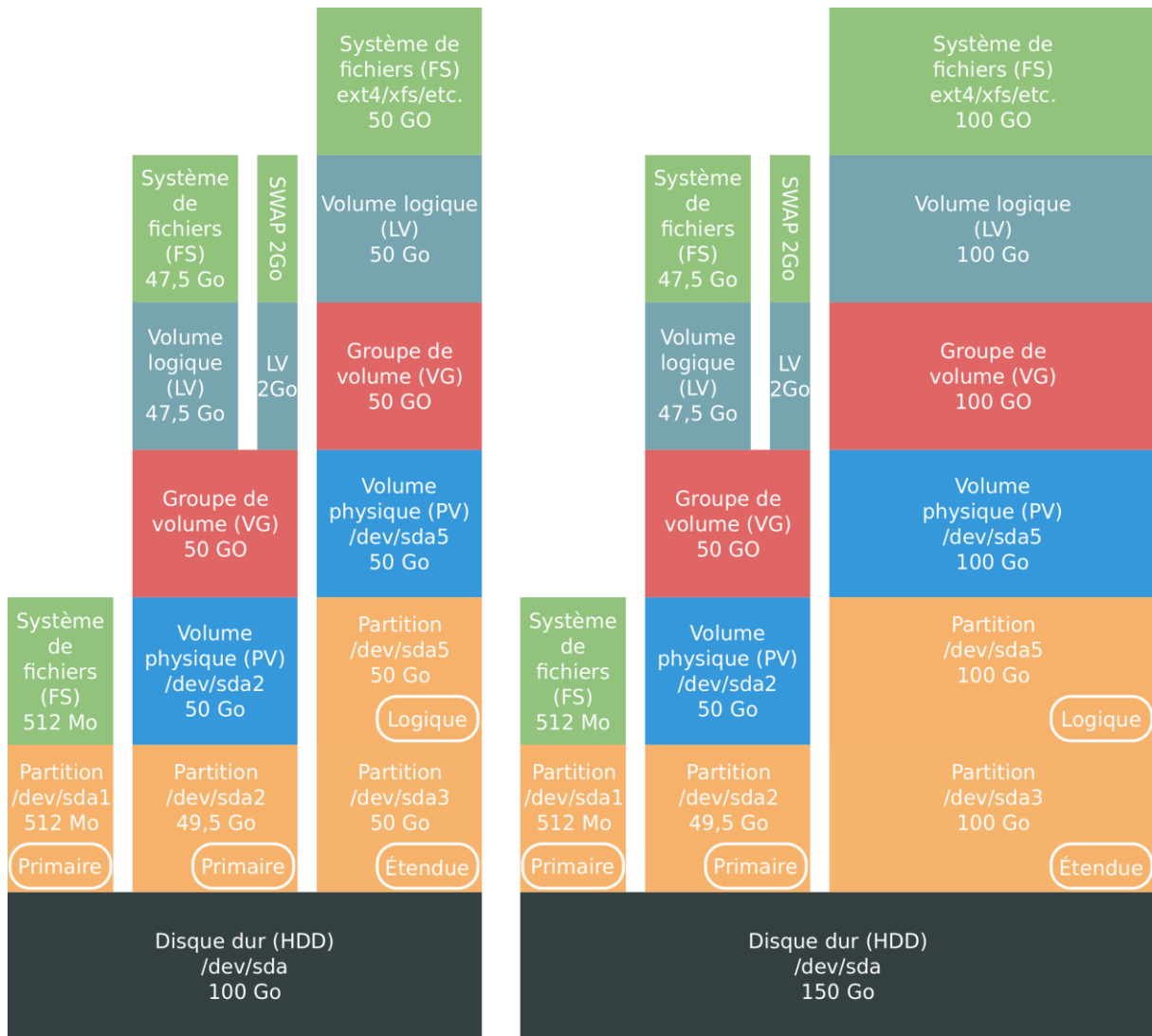
- une partition primaire *sda1* de 512 Mo utilisée pour le démarrage du système (/boot) ;
- une partition primaire *sda2* de 49,5 Go utilisée par LVM et contenant le volume principal (/) et celui de swap ;
- une partition étendue *sda3* de 50 Go ;
- une partition logique *sda5* de 50 Go utilisée par LVM et contenant le système de fichiers à agrandir.

Note : Un disque ne peut contenir que quatre partitions primaires, référencées dans la table de partitionnement. Pour utiliser davantage de partitions, il est nécessaire de créer une partition étendue. Il s'agit d'un type particulier de partition, qui est référencée dans la table de partitionnement et qui occupe donc un des quatre emplacements disponibles. Une partition étendue ne contient pas de système de fichiers, mais est un conteneur pouvant accueillir des partitions logiques. Les partitions logiques sont référencées au début de la partition étendue et leur taille est limitée à celle de la partition étendue sous-jacente.

Le disque *sda* a été agrandi de 50 Go, agrandissement qui va être répercuté sur la partition *sda5* et le système de fichiers qu'elle contient.

L'agrandissement d'une partition logique est très proche de celui d'une partition primaire, mais les contraintes sont plus fortes. Pour agrandir un volume, il faut d'abord agrandir tous les conteneurs sous-jacents, ce qui signifie dans le cas présent qu'il faut commencer par agrandir la partition étendue, puis la partition logique qu'elle contient afin de pouvoir agrandir par la suite la partie LVM et système de fichiers. Comme deux partitions doivent être agrandies, il faut de l'espace libre attendant à la partition étendue et qu'à l'intérieur de celle-ci, la partition à agrandir ne soit suivie d'aucune autre.

Les contraintes sont donc similaires à celles montrées dans la figure **FIGURE 9 – CAS DANS LESQUELS IL EST OU NON POSSIBLE D'AGRANDIR UNE PARTITION** page 12. Et le mode opératoire est similaire à celui montré dans la figure **FIGURE 12 – AGRANDISSEMENT D'UN SYSTEME DE FICHIERS AVEC LVM AVEC PARTITION** page 14.



Situation initiale

Situation cible

Figure 18 – Agrandissement d'une partition étendue et d'un volume LVM avant/après

5.3.3.2 Résumé des commandes

1. Supprimer la partition étendue à agrandir et créer une nouvelle partition étendue débutant au même endroit et ayant une taille plus importante, puis créer une nouvelle partition logique débutant au même endroit que la précédente et ayant une taille plus importante :

```

root@lucario:~# fdisk /dev/sdX
d
<extendedPartNumber>
n
e
<extendedPartNumber>
<extendedPartStartSector>
<extendedPartEndSector>
n
l
<LogicalPartNumber>
<LogicalPartStartSector>
<LogicalPartEndSector>

```

```
t
<LogicalPartNumber>
8e
w
```

2. Scanner les partitions : `partprobe`
3. Agrandir le PV : `pvresize /dev/sdX`
4. Agrandir le LV : `lvextend /dev/vgName/LvName /dev/sdX`
5. Agrandir le FS :
 - Cas ext4 : `resize2fs /dev/vgName/LvName`
 - Cas xfs : `xfs_growfs /dev/vgName/LvName`

5.3.3.3 Détail

1. Afficher les disques et partitions présents sur la machine avec `lsblk`, le disque `sda` a une taille de 150 Go et contient une partition étendue `sda3` et une partition logique `sda5` de 100 Go, qui vont être agrandies :

```
root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  150G  0 disk
├─sda1                               8:1    0   500M  0 part /boot
├─sda2                               8:2    0  49,5G  0 part
│   ├─centos-swap                    253:0    0    2G  0 lvm  [SWAP]
│   └─centos-root                    253:1    0  47,5G  0 lvm  /
├─sda3                               8:3    0    1K  0 part
├─sda5                               8:5    0   50G  0 part
│   └─vgDemo-lvDemo                 253:2    0   50G  0 lvm  /data
sr0                                  11:0    1 1024M  0 rom
```

Comme la commande `lsblk` n'indique pas le type de partition (primaire, étendue, logique), le lien entre `sda3` et `sda5` n'est pas visible.

Pour compléter les informations, il est possible d'utiliser la commande `fdisk`. Elle n'indique pas les relations entre les partitions, mais indique si l'une d'elles est de type étendu et affiche leur emplacement sur le disque.

```
root@lucario:~# fdisk -l /dev/sda

Disque /dev/sda : 161.1 Go, 161061273600 octets, 314572800 secteurs
Unités = secteur de 1 x 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x0009fb86

Périphérique Amorçage Début          Fin            Blocs          Id. Système
/dev/sda1    *           2048          1026047        512000        83  Linux
/dev/sda2           1026048      104857599      51915776      8e  Linux LVM
/dev/sda3           104857600    209715199      52428800      5  Extended
/dev/sda5           104859648    209715199      52427776      8e  Linux LVM
```

La sortie de la commande indique que `sda3` est de type « Extended ». De plus, l'information sur les secteurs de début et de fin permet de confirmer que `sda5` est bien contenue dans `sda3`.

2. Agrandir les partitions `sda3` et `sda5` avec `fdisk`. L'outil fonctionne de façon interactive, les commandes à entrer sont en couleur :

Attention : Cette étape consiste à supprimer puis recréer la partition. Une erreur peut entraîner une perte de données et potentiellement empêcher le système de redémarrer (si la partition système est modifiée).

Il faut faire attention à bien choisir le **numéro de la partition, le premier secteur et le dernier secteur**.

```
root@lucario:~# fdisk /dev/sda
Bienvenue dans fdisk (util-linux 2.23.2).

Les modifications resteront en mémoire jusqu'à écriture.
Soyez prudent avant d'utiliser la commande d'écriture.

Commande (m pour l'aide) : d
Numéro de partition (1-3,5, 5 par défaut) : 3
La partition 3 est supprimée

Commande (m pour l'aide) : n
Type de partition :
  p  primaire (2 primaire(s), 0 étendue(s), 2 libre(s))
  e  étendue
Sélection (p par défaut) : e
Numéro de partition (3,4, 3 par défaut) : 3
Premier secteur (104857600-314572799, 104857600 par défaut) : 104857600
Dernier secteur, +secteur ou +taille{K,M,G} (104857600-314572799, 314572799
par défaut) : 314572799
La partition 3 de type Extended et de taille 100 GiB est configurée

Commande (m pour l'aide) : n
Type de partition :
  p  primaire (2 primaire(s), 1 étendue(s), 1 libre(s))
  l  logique (numéroté à partir de 5)
Sélection (p par défaut) : l
Ajout de la partition logique 5
Premier secteur (104859648-314572799, 104859648 par défaut) : 104859648
Dernier secteur, +secteur ou +taille{K,M,G} (104859648-314572799, 314572799
par défaut) : 314572799
La partition 5 de type Linux et de taille 100 GiB est configurée

Commande (m pour l'aide) : t
Numéro de partition (1-3,5, 5 par défaut) : 5
Code Hexa (taper L pour afficher tous les codes) :8e
Type de partition « Linux » modifié en « Linux LVM »

Commande (m pour l'aide) : w
La table de partitions a été altérée.

Appel d'ioctl() pour relire la table de partitions.

Attention : la table de partitions n'a pas pu être relue : erreur 16 : Périphérique ou ressource occupé.
```

Le noyau continue à utiliser l'ancienne table. La nouvelle sera utilisée lors du prochain démarrage ou après avoir exécuté `partprobe(8)` ou `kpartx(8)`. Synchronisation des disques.

Explications :

- d : supprime une partition ;
 - 3 : numéro de la partition à supprimer (*sda3* étant de type étendue, cela supprime aussi toutes les partitions logiques qu'elle contient) ;
 - n : nouvelle partition ;
 - e : étendue ;
 - 3 : numéro de la partition à formater (peut varier) ;
 - premier secteur : reprendre celui de la partition supprimée ;
 - dernier secteur : indiquer la nouvelle valeur. Si la partition est la dernière sur le disque, il s'agit du plus grand secteur proposé. S'il y a une partition derrière, il faut choisir le secteur précédent le premier secteur de la partition suivante ;
 - n : nouvelle partition ;
 - l : logique ;
 - 5 : numéro de la partition logique (peut varier) ;
 - premier secteur : reprendre celui de la partition supprimée ;
 - dernier secteur : indiquer la nouvelle valeur. Si la partition est la dernière sur le disque, il s'agit du plus grand secteur proposé. S'il y a une partition derrière, il faut choisir le secteur précédent le premier secteur de la partition suivante ;
 - t : modifie le type de partition ;
 - 5 : numéro de la partition à modifier, cette étape est sautée s'il n'y en a qu'une ;
 - 8e : partition de type Linux LVM ;
 - w : écrit les changements sur le disque.
3. Scanner les partitions avec `partprobe` pour que les modifications sur *sda3* et *sda5* soient prises en compte :

```
root@lucario:~# partprobe
```

4. Afficher les disques et partitions présents sur la machine avec `lsblk`, la taille de la partition *sda5* est maintenant de 100 Go et le volume LVM n'a pas été modifié :

```
root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0  150G  0 disk
├─sda1                8:1    0   500M  0 part /boot
├─sda2                8:2    0  49,5G  0 part
│   ├─centos-swap     253:0    0    2G  0 lvm  [SWAP]
│   └─centos-root     253:1    0  47,5G  0 lvm  /
├─sda3                8:3    0   512B  0 part
└─sda5                8:5    0  100G  0 part
   └─vgDemo-lvDemo    253:2    0    50G  0 lvm  /data
sr0                  11:0    1 1024M  0 rom
```

5. Afficher les volumes physiques avec `pvs`, le volume *sda5* a une taille de 50 Go avant l'opération :

```
root@lucario:~# pvs
```

PV	VG	Fmt	Attr	PSize	PFree
/dev/sda2	centos	lvm2	a--	<49,51g	44,00m
/dev/sda5	vgDemo	lvm2	a--	<50,00g	0

6. Agrandir le volume physique avec `pvresize`, il est agrandi de façon à occuper l'ensemble de l'espace disponible sur la partition :

```
root@lucario:~# pvresize /dev/sda5
Physical volume "/dev/sda5" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
```

7. Afficher les volumes physiques avec `pvs`. Le volume `sda5` a une taille de 100 Go, dont 50 Go de libres :

```
root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2  centos  lvm2 a--  <49,51g 44,00m
/dev/sda5  vgDemo  lvm2 a--  <100,00g 50,00g
```

L'augmentation de la taille est directement prise en compte au niveau du groupe de volumes associé. Cela peut se vérifier en listant les volumes avec `vgs` :

```
root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize   VFree
centos  1  2  0 wz--n- <49,51g 44,00m
vgDemo  1  1  0 wz--n- <100,00g 50,00g
```

8. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a une taille de 50 Go avant son agrandissement :

```
root@lucario:~# lvs
LV      VG      Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Syn
c Convert
root   centos -wi-ao----- 47,46g
swap   centos -wi-ao----- 2,00g
lvDemo vgDemo -wi-ao----- <50,00g
```

9. Agrandir le volume logique avec `lvextend`. Il suffit d'indiquer à LVM le nom du volume logique à agrandir et le volume physique sur lequel réaliser l'extension. Par défaut tout l'espace disponible sur le volume physique est alloué :

```
root@lucario:~# lvextend /dev/vgDemo/lvDemo /dev/sda5
Size of logical volume vgDemo/lvDemo changed from <50,00 GiB (12799 extents)
) to <100,00 GiB (25599 extents).
Logical volume vgDemo/lvDemo successfully resized.
```

10. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a désormais une taille de 100 Go :

```
root@lucario:~# lvs
LV      VG      Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Syn
nc Convert
root   centos -wi-ao----- 47,46g
swap   centos -wi-ao----- 2,00g
```

```
lvDemo vgDemo -wi-ao---- <100,00g
```

11. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume *lvDemo* a toujours une taille de 50 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,3G   46G   5% /
devtmpfs                908M     0   908M   0% /dev
tmpfs                   920M     0   920M   0% /dev/shm
tmpfs                   920M   8,8M   911M   1% /run
tmpfs                   920M     0   920M   0% /sys/fs/cgroup
/dev/sda1              497M   251M   246M  51% /boot
/dev/mapper/vgDemo-lvDemo 50G    33M   50G   1% /data
tmpfs                  184M     0   184M   0% /run/user/0
```

12. Agrandir le système de fichiers de *lvDemo* :

- Pour xfs :

```
root@lucario:~# xfs_growfs /dev/vgDemo/lvDemo
meta-data=/dev/mapper/vgDemo-lvDemo isize=512    agcount=4, agsize=32765
44 blks
         =                               sectsz=512   attr=2, projid32bit=1
         =                               crc=1      finobt=0 spinodes=0
data     =                               bsize=4096 blocks=13106176, imaxpct=2
5
         =                               sunit=0    swidth=0 blks
naming   =version 2                      bsize=4096 ascii-ci=0 ftype=1
log      =internal                       bsize=4096 blocks=6399, version=2
         =                               sectsz=512 sunit=0 blks, lazy-count=1
realtime =none                            extsz=4096 blocks=0, rtextents=0
data blocks changed from 13106176 to 26213376
```

- Pour ext4 :

```
root@lucario:~# resize2fs /dev/vgDemo/lvDemo
```

13. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume *lvDemo* a maintenant une taille de 100 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,3G   46G   5% /
devtmpfs                908M     0   908M   0% /dev
tmpfs                   920M     0   920M   0% /dev/shm
tmpfs                   920M   8,8M   911M   1% /run
tmpfs                   920M     0   920M   0% /sys/fs/cgroup
/dev/sda1              497M   251M   246M  51% /boot
/dev/mapper/vgDemo-lvDemo 100G    33M  100G   1% /data
tmpfs                  184M     0   184M   0% /run/user/0
```

14. Effectuer une vérification complémentaire avec `lsblk` :

```
root@lucario:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	150G	0	disk	
├─sda1	8:1	0	500M	0	part	/boot
├─sda2	8:2	0	49,5G	0	part	
│ └─centos-swap	253:0	0	2G	0	lvm	[SWAP]
│ └─centos-root	253:1	0	47,5G	0	lvm	/
├─sda3	8:3	0	512B	0	part	
└─sda5	8:5	0	100G	0	part	
└─vgDemo-lvDemo	253:2	0	100G	0	lvm	/data
sr0	11:0	1	1024M	0	rom	

5.3.4 Cas n°4 – Disque avec partition non extensible

5.3.4.1 Description du cas

La machine de l'exemple possède un disque système *sda* de 50 Go et un disque secondaire *sdb* de 100 Go contenant deux partitions :

- une partition *sdb1* de 100 Go utilisée par le volume LVM qui doit être agrandi à 150 Go ;
- une partition *sdb2* de 50 Go et formatée en xfs, sans utiliser LVM.

Dans cette situation, il est impossible d'agrandir la partition *sdb1* car elle est suivie par une autre partition. Plusieurs possibilités :

- déplacer la partition *sdb2*, ce qui est complexe ;
- ajouter une partition *sdb3* en suivant de *sdb2*, cela est possible s'il reste de la place sur le disque, mais empêche alors tout redimensionnement de *sdb2* ;
- ajouter un nouveau disque dédié à LVM et l'utiliser pour agrandir le VG et ainsi le système de fichiers, c'est une bonne solution, bien qu'elle « complexifie » la configuration en répartissant les données sur deux disques. Il est cependant possible par la suite de déplacer toutes les données sur le nouveau disque pour résoudre le souci (action expliquée dans la section 6.5 page 63).

C'est la troisième possibilité qui est montrée ici. Un disque *sdc* de 50 Go a été créé pour permettre l'agrandissement du système de fichiers.

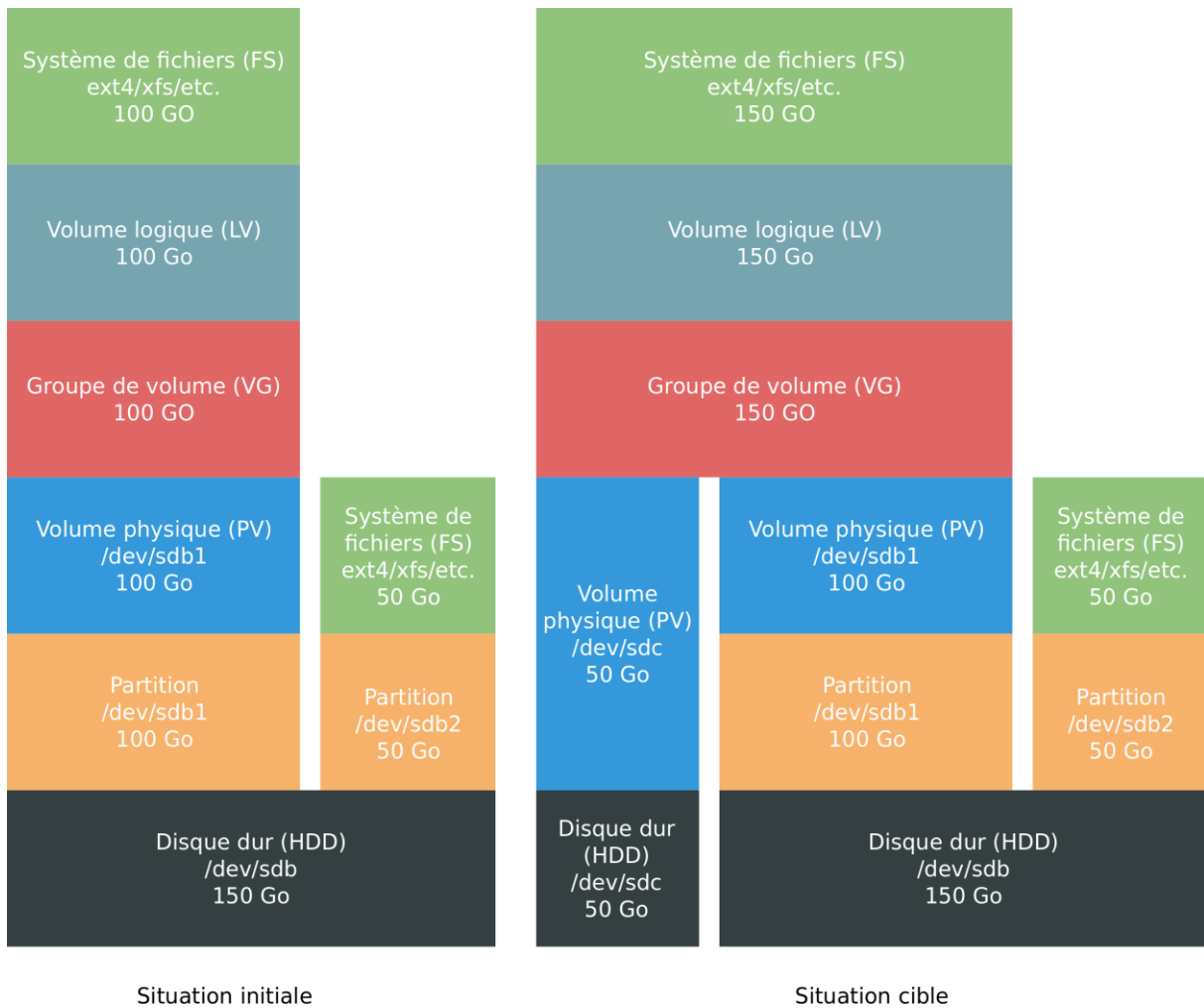


Figure 19 – Agrandissement d'un volume LVM avec un nouveau disque avant/après

5.3.4.2 Résumé des commandes

1. Créer un nouveau PV : `pvcreate /dev/sdX`
2. Intégrer le PV au VG : `vgextend vgName /dev/sdX`
3. Agrandir le LV : `lvextend /dev/vgName/LvName /dev/sdX`
4. Agrandir le FS :
 - Cas ext4 : `resize2fs /dev/vgName/LvName`
 - Cas xfs : `xfs_growfs /dev/vgName/LvName`

5.3.4.3 Détail

1. Afficher les disques et partitions présents sur la machine avec `lsblk`, le disque `sdb` a une taille de 150 Go et contient une partition `sdb1` de 100 Go utilisée par LVM ainsi qu'une partition `sdb2` utilisée pour autre chose. Le disque `sdc` de 50 Go est encore vierge et va être utilisé :

```

root@lucario:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0   50G  0 disk
├─sda1           8:1    0   500M  0 part /boot
├─sda2           8:2    0  49,5G  0 part
├─centos-swap    253:0   0     2G   0 lvm  [SWAP]
└─centos-root    253:1   0  47,5G  0 lvm  /
sdb              8:16   0  150G  0 disk
├─sdb1           8:17   0  100G  0 part

```



```

└─vgDemo-lvDemo 253:2    0 100G 0 lvm /data
└─sdb2          8:18    0  50G 0 part /storage
sdc            8:32    0  50G 0 disk
sr0           11:0     1 1024M 0 rom

```

2. Créer un volume physique avec `pvcreate` afin d'indiquer à LVM de gérer le disque `sdc` :

```

root@lucario:~# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created.

```

3. Afficher les volumes physiques avec `pvs`, le volume `sdc` apparaît, sa taille est bien égale à celle du disque :

```

root@lucario:~# pvs
PV          VG      Fmt  Attr PSize   PFree
/dev/sda2   centos lvm2  a--  <49,51g 44,00m
/dev/sdb1   vgDemo lvm2  a--  <100,00g 0
/dev/sdc    lvm2   ---   50,00g 50,00g>

```

4. Intégrer le nouveau volume physique au groupe de volumes à agrandir avec `vgextend` :

```

root@lucario:~# vgextend vgDemo /dev/sdc
Volume group "vgDemo" successfully extended

```

5. Vérifier avec `pvs` la bonne intégration du volume physique `sdc` au groupe de volumes et avec `vgs` la nouvelle taille du groupe de volumes :

```

root@lucario:~# pvs
PV          VG      Fmt  Attr PSize   PFree
/dev/sda2   centos lvm2  a--  <49,51g 44,00m
/dev/sdb1   vgDemo lvm2  a--  <100,00g 0
/dev/sdc    vgDemo lvm2  a--  <50,00g <50,00g

root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize   VFree
centos  1  2  0 wz--n- <49,51g 44,00m
vgDemo  2  1  0 wz--n- 149,99g <50,00g

```

6. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a une taille de 100 Go avant son agrandissement :

```

root@lucario:~# lvs
LV      VG      Attr          LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sy
nc Convert
root    centos -wi-ao----   47,46g
swap    centos -wi-ao----    2,00g
lvDemo  vgDemo -wi-ao----  <100,00g

```

7. Agrandir le volume logique avec `lvextend`. Il suffit d'indiquer à LVM le nom du volume logique à agrandir et le volume physique sur lequel réaliser l'extension. Par défaut tout l'espace disponible sur le volume physique est alloué :

```

root@lucario:~# lvextend /dev/vgDemo/lvDemo /dev/sdc

```

```
Size of logical volume vgDemo/lvDemo changed from <100,00 GiB (25599 extent
s) to 149,99 GiB (38398 extents).
Logical volume vgDemo/lvDemo successfully resized.
```

8. Afficher les volumes logiques avec `lvs`, le volume `lvDemo` a désormais une taille de 150 Go :

```
root@lucario:~# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Syn
c Convert
root   centos -wi-ao---- 47,46g
swap   centos -wi-ao----  2,00g
lvDemo vgDemo -wi-ao---- 149,99g
```

9. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume `lvDemo` a toujours une taille de 100 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
/dev/mapper/centos-root 48G   2,2G   46G   5% /
devtmpfs               908M   0    908M   0% /dev
tmpfs                  920M   0    920M   0% /dev/shm
tmpfs                  920M   8,8M   911M   1% /run
tmpfs                  920M   0    920M   0% /sys/fs/cgroup
/dev/sda1              497M   251M   246M   51% /boot
tmpfs                  184M   0    184M   0% /run/user/0
/dev/mapper/vgDemo-lvDemo 100G   33M   100G   1% /data
/dev/sdb2              50G    33M   50G    1% /storage
```

10. Agrandir le système de fichiers de `lvDemo` :

- Pour xfs :

```
root@lucario:~# xfs_growfs /dev/vgDemo/lvDemo
meta-data=/dev/mapper/vgDemo-lvDemo isize=512    agcount=4, agsize=65533
44 blks
         =                               sectsz=512   attr=2, projid32bit=1
         =                               crc=1       finobt=0 spinodes=0
data     =                               bsize=4096  blocks=26213376, imaxpct=2
5
         =                               sunit=0     swidth=0 blks
naming   =version 2                       bsize=4096  ascii-ci=0 ftype=1
log      =internal                       bsize=4096  blocks=12799, version=2
         =                               sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                             extsz=4096  blocks=0, rtextents=0
data blocks changed from 26213376 to 39320576
```

- Pour ext4 :

```
root@lucario:~# resize2fs /dev/vgDemo/lvDemo
```

11. Afficher la taille des systèmes de fichiers avec `df -h`, le système de fichiers du volume `lvDemo` a maintenant une taille de 150 Go :

```
root@lucario:~# df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
```

/dev/mapper/centos-root	48G	2,2G	46G	5%	/
devtmpfs	908M	0	908M	0%	/dev
tmpfs	920M	0	920M	0%	/dev/shm
tmpfs	920M	8,8M	911M	1%	/run
tmpfs	920M	0	920M	0%	/sys/fs/cgroup
/dev/sda1	497M	251M	246M	51%	/boot
tmpfs	184M	0	184M	0%	/run/user/0
/dev/mapper/vgDemo-lvDemo	150G	33M	150G	1%	/data
/dev/sdb2	50G	33M	50G	1%	/storage

12. Effectuer une vérification complémentaire avec `lsblk` :

```

root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   50G  0 disk
├─sda1                               8:1    0  500M  0 part /boot
└─sda2                               8:2    0 49,5G  0 part
   ├─centos-swap                     253:0    0    2G  0 lvm  [SWAP]
   └─centos-root                     253:1    0 47,5G  0 lvm  /
sdb                                  8:16    0  150G  0 disk
├─sdb1                              8:17    0  100G  0 part
│  └─vgDemo-lvDemo                 253:2    0  150G  0 lvm  /data
└─sdb2                              8:18    0    50G  0 part /storage
sdc                                  8:32    0    50G  0 disk
└─vgDemo-lvDemo                 253:2    0  150G  0 lvm  /data
sr0                                  11:0    1 1024M  0 rom

```

Note : Il est possible d'aller plus loin en agrandissant le disque *sdc* pour y déplacer entièrement le contenu de *sdb1* et ainsi regrouper tout le volume logique *lvDemo* sur un seul disque. Cela permettrait de supprimer la partition *sdb1*.

En revanche, cela laisserait un « trou » dans le disque *sdb*, avec de l'espace non utilisé et donc perdu. Comme la partition *sdb2* est une partition classique, son déplacement est complexe, risqué et ne peut pas se faire à chaud. Il serait donc difficile d'arriver à une situation propre. Et même si cela est possible, on ne pourrait pas gagner d'espace car il est déconseillé de réduire la taille d'un disque. Ainsi, même avec *sdb2* déplacée en début du disque *sdb*, il y aurait de l'espace non utilisé à la fin du disque (il pourrait être utilisé pour une nouvelle partition ou pour agrandir *sdb2*).

Dans le cas actuel, il est donc préférable de conserver les partitions telles qu'elles sont.

6 Cas pratique : réorganisation des volumes LVM

6.1 Affichage de la répartition des extents des volumes physiques

Les volumes physiques (PV) ajoutés à un groupe de volumes (VG) sont découpés en extents, des blocs de données qui sont ensuite alloués aux différents volumes logiques (LV) appartenant à ce groupe. La figure FIGURE 4 – EXEMPLE D'ALLOCATION DES EXTENTS PAR LVM page 8 illustre ce concept.

Deux commandes sont utiles pour connaître en temps réel la répartition des extents d'un volume physique :

- `lsblk` : pour chaque PV, cette commande affiche les LV ayant au moins un extent sur ce PV, cela donne une vision d'ensemble ;
- `pvdisplay -m` : affiche de façon détaillée quels sont les extents libres d'un PV et quels sont ceux alloués aux différents LV.

Ces commandes sont utiles pour mieux appréhender les machines à la configuration disque assez touffue voire désorganisée, comme la machine de l'exemple ci-dessous.

La commande `lsblk` permet d'identifier tous les volumes logiques (type = lvm) ainsi que le volume physique parent.

```
root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0                                  2:0      1   4K  0 disk
sda                                  8:0      0 150G  0 disk
├─sda1                               8:1      0 250M  0 part /boot
├─sda2                               8:2      0 17,6G  0 part
│   ├─rootvg-lv_01 (dm-0)           252:0    0    3G  0 lvm /
│   ├─rootvg-lv_var (dm-4)          252:4    0    2G  0 lvm /var
│   ├─rootvg-lv_home (dm-5)         252:5    0    3G  0 lvm /home
│   ├─rootvg-lv_tmp (dm-6)          252:6    0    2G  0 lvm /tmp
│   └─rootvg-lv_usr (dm-7)          252:7    0   3,9G  0 lvm /usr
├─sda3                               8:3      0   3,9G  0 part [SWAP]
├─sda4                               8:4      0    1K  0 part
├─sda5                               8:5      0 28,3G  0 part
│   ├─datavg-lv_u01 (dm-1)          252:1    0 64,7G  0 lvm /u01
│   ├─datavg-lv_u02 (dm-2)          252:2    0   4,9G  0 lvm /u02
│   └─datavg-lv_u03 (dm-3)          252:3    0 25,9G  0 lvm /u03
├─sdb                               8:16     0   20G  0 disk
│   └─datavg-lv_u03 (dm-3)          252:3    0 25,9G  0 lvm /u03
├─sdc                               8:32     0   50G  0 disk
│   └─datavg-lv_u01 (dm-1)          252:1    0 64,7G  0 lvm /u01
└─sr0                              11:0     1 1024M  0 rom
```

Le retour de la commande `lsblk` illustre bien le fait qu'un LV est affiché comme enfant d'un PV si et seulement si une partie de ce LV est réellement stockée sur le PV. Par exemple, les volumes physiques `sda5` et `sdc` appartiennent au même groupe de volumes, mais seul `sda5` est affiché comme parent du volume logique `lv_u003`.

La configuration LVM de cette machine :

Volume physique (PV)	Groupe de volumes (VG)	Volumes logiques (LV)
sda2	rootvg	lv_01

		lv_var lv_home lv_tmp lv_usr
sda5	datavg	lv_u01 lv_u02 lv_u03
sdb		lv_u03
sdc		lv_u01

Cette machine possède de nombreux LV, éparpillés sur plusieurs VG et PV, avec des disques dédiés à LVM, des partitions et des partitions étendues. C'est un exemple réel, montrant bien qu'aucun guide ne peut anticiper tous les cas possibles (cette machine cochant à peu près toutes les cases !).

Le volume physique *sda5* est intéressant, il est possible d'afficher le détail de sa répartition avec la commande `pvdisplay -m` :

```
root@lucario:~# pvdisplay /dev/sda5 -m
--- Physical volume ---
PV Name                /dev/sda5
VG Name                datavg
PV Size                28,27 GiB / not usable 4,00 MiB
Allocatable           yes
PE Size                4,00 MiB
Total PE               7236
Free PE                730
Allocated PE           6506
PV UUID                dUmPwC-Rri0-oVtH-Yx9z-rk3Y-5f2M-Fru1fL

--- Physical Segments ---
Physical extent 0 to 3749:
  Logical volume       /dev/datavg/lv_u01
  Logical extents      0 to 3749
Physical extent 3750 to 4999:
  Logical volume       /dev/datavg/lv_u02
  Logical extents      0 to 1249
Physical extent 5000 to 6505:
  Logical volume       /dev/datavg/lv_u03
  Logical extents      0 to 1505
Physical extent 6506 to 7235:
  FREE
```

Le retour de la commande montre la taille des extents (4 Mo), le nombre total d'extents (7236) et ceux qui sont libres (730). Ensuite, le détail de la répartition de chaque extent entre les volumes logiques est donnée.

6.2 Regroupement de groupes de volumes

6.2.1 Description du cas

La machine de l'exemple possède un disque système *sda* et deux disques secondaires *sdb* et *sdc* utilisés par LVM de la façon suivante :

- deux volumes physiques (PV) : *sdb* et *sdc* ;

- deux groupes de volumes (VG) : *vgDemo1* et *vgDemo2*, contenant chacun le PV correspondant ;
- deux volumes logiques (LV) : *lvDemo1* et *lvDemo2*, chacun associé au VG correspondant.

L'objectif est de regrouper les deux groupes de volumes. Le groupe *vgDemo2* va disparaître et tous les PV qu'il contient vont être intégrés au groupe *vgDemo1*. Dans le cas de l'exemple, cela n'a aucun intérêt. Mais cela est utile pour partager les ressources entre deux groupes de volumes. Par exemple, si le disque *sdb* est plein et non extensible, alors qu'il reste de la place (c.-à-d. des extents non alloués) sur le disque *sdc*, regrouper les deux groupes permet d'exploiter cette espace libre.

Note : L'opération se fait sans perte de données et sans déplacement. Elle ne fait que modifier le VG associé au PV.

Note : La fusion de deux VG n'est possible que si la taille de leurs extents est identique. En effet cette valeur est définie dans le VG pour l'ensemble des PV qu'il contient. Tous les PV d'un VG doivent donc avoir la même taille pour leurs extents.

Note : Cette opération ne peut pas se faire à chaud, les LV du groupe de volumes à fusionner doivent être inactifs et les systèmes de fichiers qu'ils contiennent doivent être démontés. En effet, le chemin vers le périphérique est de la forme « `/dev/vgName/lvName` », il contient le nom du VG. Comme ce dernier change, les instructions pour monter le volume sont à modifier aussi.

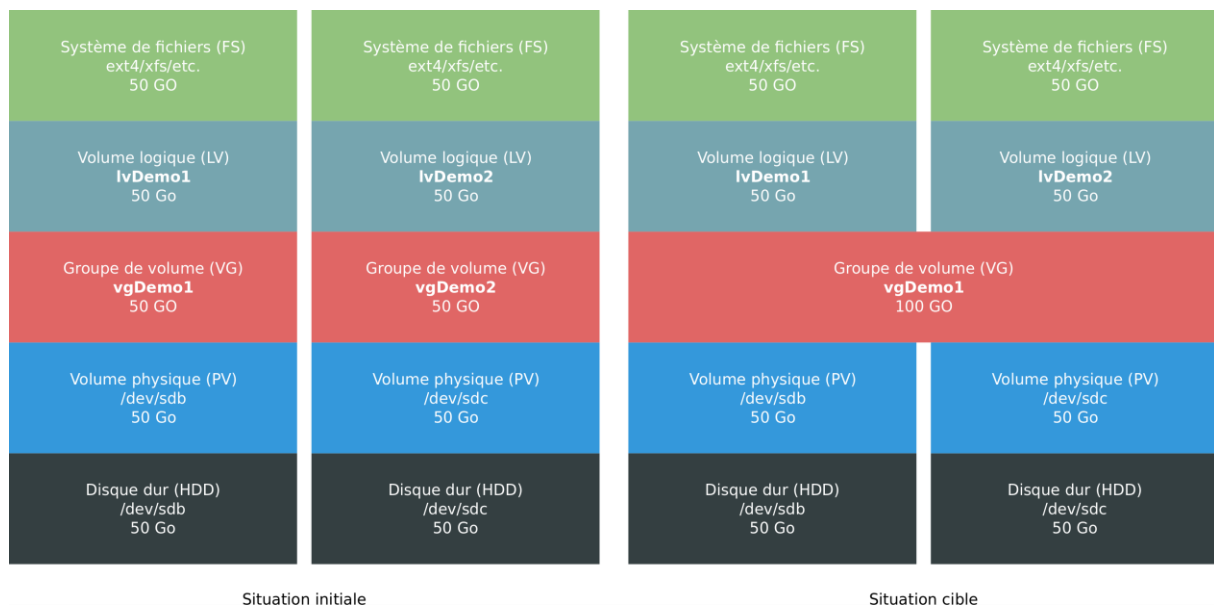


Figure 20 – Regroupement de deux groupes de volumes avant/après

6.2.2 Résumé des commandes

1. Prérequis : taille des extents identique entre les VG, LV impactés et points de montages tous identifiés.
2. Démonter tous les points de montage des LV à migrer : `umount /mountPath`
3. Désactiver les LV à migrer : `vgchange -a n vgName2`
4. Fusionner les VG (le second dans le premier) : `vgmerge vgName1 vgName2`
5. Réactiver les LV migrés : `vgchange -a y vgName1`

6. Modifier les instructions de montage dans le fichier `/etc/fstab`, remplacer « `/dev/vgName2` » par « `/dev/vgName1` ».
7. Monter tous les systèmes de fichiers : `mount -a`

6.2.3 Détail

1. Analyser la configuration de la machine avec les commandes `lsblk`, `pvs`, `vgs` et `lvs`.
 - a. La commande `lsblk` donne une vision d'ensemble de la situation. Les disques `sdb` et `sdc` sont utilisés par LVM et contiennent les volumes logiques `lvDemo1` et `lvDemo2` qui sont montés, respectivement, sur `/data` et `/storage` :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0    50G  0 disk
├─sda1                8:1    0   500M  0 part /boot
├─sda2                8:2    0  49,5G  0 part
│   └─centos-swap     253:0    0     2G  0 lvm  [SWAP]
│       └─centos-root 253:1    0  47,5G  0 lvm  /
sdb                  8:16    0    50G  0 disk
└─vgDemo1-lvDemo1    253:2    0    50G  0 lvm  /data
sdc                  8:32    0    50G  0 disk
└─vgDemo2-lvDemo2    253:3    0    50G  0 lvm  /storage
sr0                  11:0    1 1024M  0 rom
  
```

- b. La commande `pvs` liste les volumes physiques et le groupe de volumes associé. Les volumes physiques `sdb` et `sdc` sont affectés respectivement aux groupes de volumes `vgDemo1` et `vgDemo2` :

```

root@lucario:~# pvs
PV                VG          Fmt Attr PSize  PFree
/dev/sda2         centos      lvm2 a--  <49,51g 44,00m
/dev/sdb          vgDemo1     lvm2 a--  <50,00g  0
/dev/sdc          vgDemo2     lvm2 a--  <50,00g  0
  
```

- c. La commande `vgs` liste les groupes de volumes présents. Les groupes de volumes `vgDemo1` et `vgDemo2` sont présents :

```

root@lucario:~# vgs
VG          #PV #LV #SN Attr   VSize  VFree
centos      1  2  0 wz--n- <49,51g 44,00m
vgDemo1     1  1  0 wz--n- <50,00g  0
vgDemo2     1  1  0 wz--n- <50,00g  0
  
```

- d. La commande `lvs` liste les volumes logiques et le groupe de volumes associé. Les volumes `lvDemo1` et `lvDemo2` sont affectés respectivement aux groupes de volumes `vgDemo1` et `vgDemo2` :

```

root@lucario:~# lvs
LV          VG          Attr          LSize  Pool Origin Data%  Meta%  Move L
og Cpy%Sync Convert
root        centos      -wi-ao----- 47,46g
swap        centos      -wi-ao-----  2,00g
lvDemo1     vgDemo1     -wi-ao----- <50,00g
lvDemo2     vgDemo2     -wi-ao----- <50,00g
  
```

La situation de l'exemple est donc la suivante :

- les VG *vgDemo1* et *vgDemo2* vont être fusionnés, entraînant la suppression du VG *vgDemo2* ;
 - le VG *vgDemo2* possède un seul PV *sdC* qui sera intégré au VG *vgDemo1* ;
 - le VG *vgDemo2* contient un LV *lvDemo2* dont le système de fichiers est monté dans */storage*, il devra être démonté le temps de l'opération et le fichier */etc/fstab* modifié pour reparamétrer le montage du système de fichiers.
2. Vérifier que les extents des deux groupes de volumes à fusionner ont bien la même taille avec `vgdisplay`, elle est ici de 4 Mo pour les deux :

```
root@lucario:~# vgdisplay vgDemo1
--- Volume group ---
VG Name                vgDemo1
System ID
Format                 lvm2
Metadata Areas        1
Metadata Sequence No  2
VG Access              read/write
VG Status              resizable
MAX LV                0
Cur LV               1
Open LV               1
Max PV                0
Cur PV               1
Act PV                1
VG Size               <50,00 GiB
PE Size               4,00 MiB
Total PE              12799
Alloc PE / Size      12799 / <50,00 GiB
Free PE / Size       0 / 0
VG UUID               bmv0z-sv6J-MUaW-tFjp-XUOZ-3IqP-AT0mPR

root@lucario:~# vgdisplay vgDemo2
--- Volume group ---
VG Name                vgDemo2
System ID
Format                 lvm2
Metadata Areas        1
Metadata Sequence No  2
VG Access              read/write
VG Status              resizable
MAX LV                0
Cur LV               1
Open LV               1
Max PV                0
Cur PV               1
Act PV                1
VG Size               <50,00 GiB
PE Size               4,00 MiB
Total PE              12799
Alloc PE / Size      12799 / <50,00 GiB
Free PE / Size       0 / 0
VG UUID               u6n8PK-uXKH-jH5k-oHtA-A0mf-HEeJ-urhqCr
```

3. Démontez tous les systèmes de fichiers liés au groupe de volumes qui va être supprimé avec `umount`. Ici, le groupe *vgDemo2* contient un seul volume logique *lvDemo2* monté sur */storage* :


```
root@lucario:~# umount /storage
```

- Désactiver tous les volumes logiques liés au groupe de volumes qui va être supprimé avec `vgchange`. Ici, le groupe `vgDemo2` contient un seul volume logique `lvDemo2` :

```
root@lucario:~# vgchange -a n vgDemo2
0 logical volume(s) in volume group "vgDemo2" now active
```

- Effectuer un test de fusion des groupes de volumes `vgDemo1` et `vgDemo2` avec `vgmerge -vt`. La commande doit indiquer un succès de l'opération :

```
root@lucario:~# vgmerge -vt vgDemo1 vgDemo2
TEST MODE: Metadata will NOT be updated and volumes will not be (de)activ
ated.
Checking for volume group "vgDemo1"
Checking for volume group "vgDemo2"
Test mode: Skipping archiving of volume group.
Test mode: Skipping archiving of volume group.
Writing out updated volume group
Test mode: Skipping backup of volume group.
Volume group "vgDemo2" successfully merged into "vgDemo1"
Test mode: Wiping internal cache
Wiping internal VG cache
```

- Fusionner les groupes de volumes `vgDemo1` et `vgDemo2` avec `vgmerge` :

```
root@lucario:~# vgmerge vgDemo1 vgDemo2
Volume group "vgDemo2" successfully merged into "vgDemo1"
```

- Vérifier la fusion avec `pvs`, `vgs` et `lvs`. Le volume physique `sdc` et le volume logique `lvDemo2` sont maintenant liés au groupe `vgDemo1`. Et le groupe de volumes `vgDemo2` n'existe plus :

```
root@lucario:~# pvs
PV          VG      Fmt Attr PSize  PFree
/dev/sda2   centos  lvm2 a-- <49,51g 44,00m
/dev/sdb    vgDemo1 lvm2 a-- <50,00g  0
/dev/sdc    vgDemo1 lvm2 a-- <50,00g  0
root@lucario:~# lvs
LV          VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy
%Sync Convert
root       centos  -wi-ao---- 47,46g
swap       centos  -wi-ao----  2,00g
lvDemo1    vgDemo1 -wi-ao---- <50,00g
lvDemo2    vgDemo1 -wi----- <50,00g
root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize  VFree
centos   1  2  0 wz--n- <49,51g 44,00m
vgDemo1  2  2  0 wz--n- 99,99g  0
```

- Effectuer un scan des volumes logiques avec `lvscan`, cela montre que le volume logique `lvDemo2` est toujours inactif :

```
root@lucario:~# lvscan
```

```

ACTIVE          '/dev/centos/swap' [2,00 GiB] inherit
ACTIVE          '/dev/centos/root' [47,46 GiB] inherit
ACTIVE          '/dev/vgDemo1/lvDemo1' [<50,00 GiB] inherit
inactive        '/dev/vgDemo1/lvDemo2' [<50,00 GiB] inherit

```

9. Réactiver tous les volumes logiques migrés avec `vgchange` :

```

root@lucario:~# vgchange -a y vgDemo1
2 logical volume(s) in volume group "vgDemo1" now active

```

10. Effectuer un scan des volumes logiques avec `lvscan`, ils sont désormais tous actifs :

```

root@lucario:~# lvscan
ACTIVE          '/dev/centos/swap' [2,00 GiB] inherit
ACTIVE          '/dev/centos/root' [47,46 GiB] inherit
ACTIVE          '/dev/vgDemo1/lvDemo1' [<50,00 GiB] inherit
ACTIVE          '/dev/vgDemo1/lvDemo2' [<50,00 GiB] inherit

```

11. Modifier le paramétrage du montage des systèmes de fichiers migrés dans le fichier `/etc/fstab`. Dans l'exemple, la ligne suivante est à modifier :

```

/dev/vgDemo2/lvDemo2 /storage xfs defaults 0 0

```

Nouvelle version :

```

/dev/vgDemo1/lvDemo2 /storage xfs defaults 0 0

```

12. Monter tous les systèmes de fichiers avec `mount` :

```

root@lucario:~# mount -a

```

13. Vérifier la situation avec `lsblk`. Le disque `sdc` contient toujours le volume logique `lvDemo2`, appartenant désormais au groupe `vgDemo1` et dont le système de fichiers est monté correctement dans `/storage` :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1               8:1    0   500M  0 part /boot
├─sda2               8:2    0  49,5G  0 part
│   ├─centos-swap    253:0   0    2G   0 lvm  [SWAP]
│   └─centos-root    253:1   0  47,5G  0 lvm  /
sdb                  8:16   0   50G  0 disk
├─vgDemo1-lvDemo1   253:2   0   50G  0 lvm  /data
sdc                  8:32   0   50G  0 disk
├─vgDemo1-lvDemo2   253:3   0   50G  0 lvm  /storage
sr0                 11:0    1 1024M  0 rom

```

6.3 Séparation d'un groupe de volumes en deux

6.3.1 Description du cas

La machine de l'exemple possède un disque système `sda` et deux disques secondaires `sdb` et `sdc` utilisés par LVM de la façon suivante :

- deux volumes physiques (PV) : *sdb* et *sd*c ;
- un groupe de volumes (VG) : *vgDemo1*, contenant les deux PV *sdb* et *sd*c ;
- deux volumes logiques (LV) : *lvDemo1* et *lvDemo2*, tous deux contenus dans le VG *vgDemo1*.

L'objectif est de séparer les deux volumes logiques dans des groupes de volumes dédiés *vgDemo1* et *vgDemo2*. Tous les PV contenant des données de *lvDemo2* vont donc être transférés vers le groupe *vgDemo2*.

La commande `vgsplit` va être utilisée pour cela. Elle permet de déplacer des volumes logiques d'un groupe à un autre, en créant un nouveau groupe si besoin. Il est possible d'indiquer soit une liste de PV à déplacer, soit un LV. Comme un LV ne peut appartenir qu'à un seul VG, s'il est déplacé, tous les PV sous-jacents le seront. De même, un PV ne peut appartenir qu'à un seul VG. S'il est déplacé, tous les LV s'appuyant sur ce PV doivent être déplacés.

Note : L'opération se fait sans perte de données et sans déplacement. Elle ne fait que modifier le VG associé au PV.

Note : La séparation d'un VG (ou le déplacement de PV/LV) vers un VG déjà existant n'est possible que si la taille de leurs extents est identique. En effet cette valeur est définie dans le VG pour l'ensemble des PV qu'il contient. Tous les PV d'un VG doivent donc avoir la même taille pour leurs extents.

Note : Cette opération ne peut pas se faire à chaud, les LV à migrer doivent être inactifs et les systèmes de fichiers qu'ils contiennent doivent être démontés. En effet, le chemin vers le périphérique est de la forme « `/dev/vgName/lvName` », il contient le nom du VG. Comme ce dernier change, les instructions pour monter le volume sont à modifier aussi.

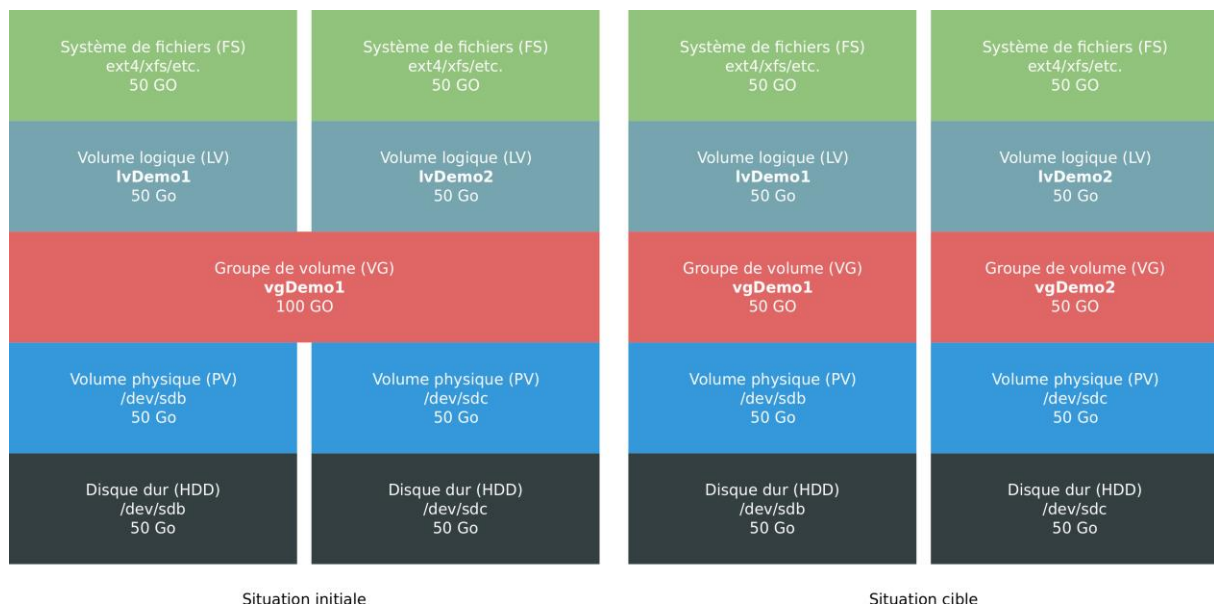


Figure 21 – Séparation d'un groupe de volumes en deux avant/après

6.3.2 Résumé des commandes

1. Prérequis : taille des extents identique entre les VG, LV impactés et points de montages tous identifiés.

2. Démonter tous les points de montage des LV à migrer : `umount /mountPath`
3. Désactiver les LV à migrer : `lvchange -a n /dev/vgName1/LvName`
4. Diviser le VG (en en créant un nouveau) : `vsplit -n LvName vgName1 vgName2`
5. Réactiver les LV migrés : `vgchange -a y vgName2`
6. Modifier les instructions de montage dans le fichier `/etc/fstab`, remplacer « `/dev/vgName1/lvName` » par « `/dev/vgName2/lvName` ».
7. Monter tous les systèmes de fichiers : `mount -a`

6.3.3 Détail

1. Analyser la configuration de la machine avec les commandes `lsblk`, `pvs`, `vgs` et `lvs`.
 - a. La commande `lsblk` donne une vision d'ensemble de la situation. Les disques `sdb` et `sdc` sont utilisés par LVM et contiennent les volumes logiques `lvDemo1` et `lvDemo2` qui sont montés, respectivement, sur `/data` et `/storage` :

```

root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0   50G  0 disk
├─sda1                               8:1      0  500M  0 part /boot
└─sda2                               8:2      0  49,5G  0 part
   ├─centos-swap                     253:0    0    2G   0 lvm  [SWAP]
   └─centos-root                     253:1    0  47,5G  0 lvm  /
sdb                                  8:16     0   50G  0 disk
├─vgDemo1-lvDemo1                  253:2    0   50G  0 lvm  /data
sdc                                  8:32     0   50G  0 disk
├─vgDemo1-lvDemo2                  253:3    0   50G  0 lvm  /storage
sr0                                  11:0     1 1024M  0 rom

```

- b. La commande `pvs` liste les volumes physiques et le groupe de volumes associé. Les volumes physiques `sdb` et `sdc` sont affectés au groupe de volumes `vgDemo1` :

```

root@lucario:~# pvs
PV          VG          Fmt  Attr  PSize  PFree
/dev/sda2  centos     lvm2 a--   <49,51g 44,00m
/dev/sdb   vgDemo1    lvm2 a--   <50,00g  0
/dev/sdc   vgDemo1    lvm2 a--   <50,00g  0

```

- c. La commande `vgs` liste les groupes de volumes présents. Le groupe de volumes `vgDemo1` est présent :

```

root@lucario:~# vgs
VG          #PV #LV #SN Attr   VSize  VFree
centos      1  2  0 wz--n- <49,51g 44,00m
vgDemo1    2  2  0 wz--n-  99,99g  0

```

- d. La commande `lvs` liste les volumes logiques et le groupe de volumes associé. Les volumes `lvDemo1` et `lvDemo2` sont affectés au groupe de volumes `vgDemo1` :

```

root@lucario:~# lvs
LV          VG          Attr   LSize  Pool Origin Data%  Meta%  Move L
og Cpy%Sync Convert
root       centos     -wi-ao---- 47,46g
swap       centos     -wi-ao----  2,00g
lvDemo1    vgDemo1    -wi-ao---- <50,00g

```

```
lvDemo2 vgDemo1 -wi-ao---- <50,00g
```

La situation de l'exemple est donc la suivante :

- le VG *vgDemo1* va être séparé en deux VG *vgDemo1* et *vgDemo2*, en déplaçant sur le nouveau VG le volume logique *lvDemo2* ;
 - le LV *lvDemo2* est associé à un seul PV *sdC* qui lui est dédié, ce PV sera intégré au VG *vgDemo2* ;
 - le LV *lvDemo2* contient un système de fichiers est monté dans */storage*, il devra être démonté le temps de l'opération et le fichier */etc/fstab* modifié pour reparamétrer le montage du système de fichiers.
2. Démontez tous les systèmes de fichiers liés aux volumes logiques qui seront déplacés avec `umount`. Ici, le volume *lvDemo2* est monté sur */storage* :

```
root@lucario:~# umount /storage/
```

3. Désactivez tous les volumes logiques qui seront déplacés avec `lvchange` :

```
root@lucario:~# lvchange -a n /dev/vgDemo1/lvDemo2
```

4. Effectuez un test de division du groupe de volumes *vgDemo1* avec `vgsplit -vt`. La commande doit indiquer un succès de l'opération :

```
root@lucario:~# vgsplit -n lvDemo2 vgDemo1 vgDemo2 -vt
TEST MODE: Metadata will NOT be updated and volumes will not be (de)activated.
Checking for volume group "vgDemo1"
Checking for new volume group "vgDemo2"
Test mode: Skipping archiving of volume group.
Writing out updated volume groups
Test mode: Skipping archiving of volume group.
Test mode: Skipping backup of volume group.
Test mode: Skipping backup of volume group.
Test mode: Skipping backup of volume group.
New volume group "vgDemo2" successfully split from "vgDemo1"
Test mode: Wiping internal cache
Wiping internal VG cache
```

5. Divisez le groupe de volumes *vgDemo1* avec `vgsplit` :

```
root@lucario:~# vgsplit -n lvDemo2 vgDemo1 vgDemo2
New volume group "vgDemo2" successfully split from "vgDemo1"
```

6. Vérifiez la division avec `pvs`, `vgs` et `lvs`. Le volume physique *sdC* et le volume logique *lvDemo2* sont maintenant liés au groupe *vgDemo2* qui a été créé durant l'opération :

```
root@lucario:~# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sda2  centos  lvm2 a--  <49,51g 44,00m
/dev/sdb    vgDemo1 lvm2 a--  <50,00g  0
/dev/sdc    vgDemo2 lvm2 a--  <50,00g  0
root@lucario:~# vgs
VG          #PV #LV #SN Attr   VSize  VFree
```

```

centos    1    2    0 wz--n- <49,51g 44,00m
vgDemo1  1    1    0 wz--n- <50,00g    0
vgDemo2  1    1    0 wz--n- <50,00g    0
root@lucario:~# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy
%Sync Convert
root    centos  -wi-ao---- 47,46g
swap    centos  -wi-ao----  2,00g
lvDemo1 vgDemo1 -wi-ao---- <50,00g
lvDemo2 vgDemo2 -wi-a----- <50,00g

```

7. Effectuer un scan des volumes logiques avec `lvscan`, cela montre que le volume logique `lvDemo2` est toujours inactif :

```

root@lucario:~# lvscan
ACTIVE          '/dev/centos/swap' [2,00 GiB] inherit
ACTIVE          '/dev/centos/root' [47,46 GiB] inherit
ACTIVE          '/dev/vgDemo1/lvDemo1' [<50,00 GiB] inherit
inactive        '/dev/vgDemo2/lvDemo2' [<50,00 GiB] inherit

```

8. Réactiver tous les volumes logiques migrés avec `lvchange` :

```

root@lucario:~# lvchange -a y /dev/vgDemo2/lvDemo2

```

9. Effectuer un scan des volumes logiques avec `lvscan`, ils sont désormais tous actifs :

```

root@lucario:~# lvscan
ACTIVE          '/dev/centos/swap' [2,00 GiB] inherit
ACTIVE          '/dev/centos/root' [47,46 GiB] inherit
ACTIVE          '/dev/vgDemo1/lvDemo1' [<50,00 GiB] inherit
ACTIVE          '/dev/vgDemo2/lvDemo2' [<50,00 GiB] inherit

```

10. Modifier le paramétrage du montage des systèmes de fichiers migrés dans le fichier `/etc/fstab`. Dans l'exemple, la ligne suivante est à modifier :

```

/dev/vgDemo1/lvDemo2 /storage xfs defaults 0 0

```

Nouvelle version :

```

/dev/vgDemo2/lvDemo2 /storage xfs defaults 0 0

```

11. Monter tous les systèmes de fichiers avec `mount` :

```

root@lucario:~# mount -a

```

12. Vérifier la situation avec `lsblk`. Le disque `sdc` contient toujours le volume logique `lvDemo2`, appartenant désormais au groupe `vgDemo2` et dont le système de fichiers est monté correctement dans `/storage` :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0    50G  0 disk

```

└sda1	8:1	0	500M	0	part	/boot
└sda2	8:2	0	49,5G	0	part	
└└centos-swap	253:0	0	2G	0	lvm	[SWAP]
└└centos-root	253:1	0	47,5G	0	lvm	/
sdb	8:16	0	50G	0	disk	
└vgDemo1-lvDemo1	253:2	0	50G	0	lvm	/data
sdc	8:32	0	50G	0	disk	
└vgDemo2-lvDemo2	253:3	0	50G	0	lvm	/storage
sr0	11:0	1	1024M	0	rom	

6.4 Déplacement d'un volume physique ou logique dans un autre groupe de volumes

La commande `vgsplit` permet de déplacer un volume logique (et les volumes physiques sous-jacents) d'un groupe de volume à un autre. C'est comme cela qu'elle permet de séparer un groupe en deux. D'ailleurs, pour séparer un groupe de volumes, il est nécessaire d'indiquer soit des PV, soit des LV à déplacer.

L'exemple de la section 6.3 SEPARATION D'UN GROUPE DE VOLUMES fonctionne également pour le déplacement d'un volume logique entre deux groupes déjà existants. Il faut simplement s'assurer que les deux groupes ont la même taille d'extents. De plus, comme un LV ne peut appartenir qu'à un seul groupe, s'il est déplacé, tous les PV sous-jacents le seront. Réciproquement, un PV ne peut appartenir qu'à un seul groupe. S'il est déplacé, tous les LV s'appuyant sur ce PV doivent être déplacés.

6.5 Réorganisation des volumes LVM d'une machine

6.5.1 Description du cas

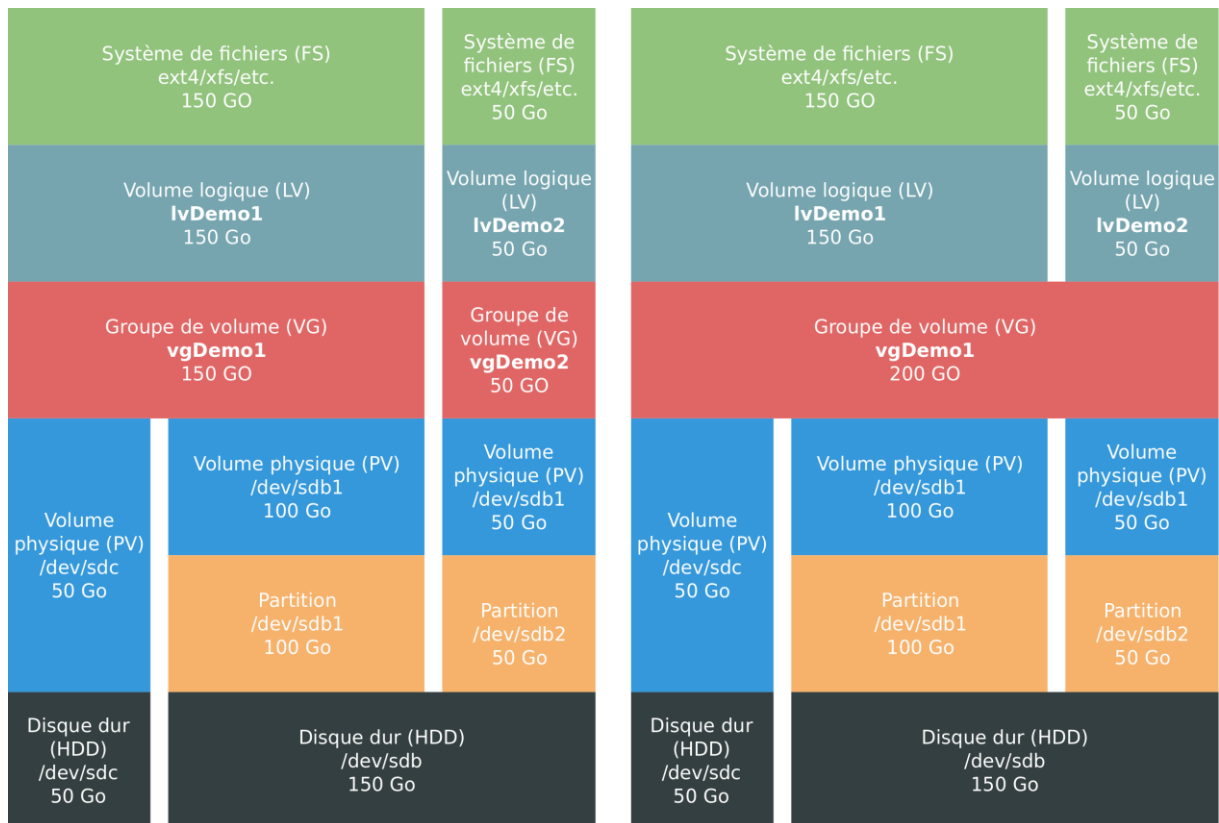
La situation de départ est proche de celle de la fin de la section 5.3.4 où un disque a été ajouté à la machine pour agrandir un système de fichiers.

La machine de l'exemple possède trois disques :

- un disque système *sda* de 50 Go ;
- un disque *sdb* de 150 Go contenant deux partitions :
 - une partition *sdb1* de 100 Go utilisée par le volume LVM *lvDemo1* ;
 - une partition *sdb2* de 50 Go utilisée par le volume LVM *lvDemo2* ;
- un disque *sdc* de 50 Go utilisée par le volume LVM *lvDemo1*.

Note : Dans l'exemple de la section 5.3.4, la partition *sdb2* était formatée en *xf*s sans utiliser LVM, ce qui empêchait une réorganisation efficace. Le problème est résolu dans le cas actuel.

La situation de départ peut connaître deux variantes : les deux volumes logiques regroupés au sein d'un même groupe, ou bien séparés dans des groupes dédiés. Cela n'a aucun impact sur le choix de la solution finale, il est en effet très simple de fusionner ou de séparer deux groupes de volumes (cf. sections 6.2 et 6.3).



Situation initiale 1 : 2 VG

Situation initiale 2 : 1 VG

Figure 22 – Réorganisation des disques d'une machine, situations initiales possibles

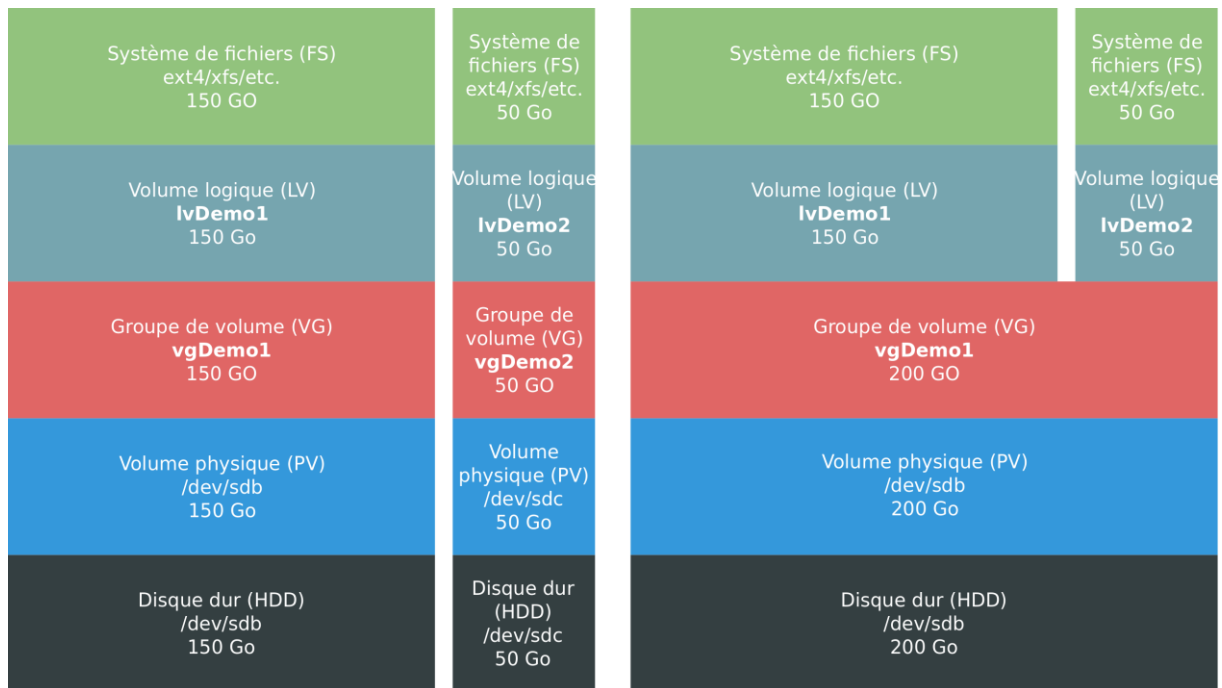
Il y a deux problèmes à résoudre :

- le volume logique **lvDemo1** est réparti sur deux volumes physiques (**sdb1** et **sdc**) ;
- le disque **sdb** est partitionné.

Quelle que soit la variante, les solutions possibles sont les suivantes :

1. regrouper les deux LV dans le même VG et les réunir sur un seul disque non partitionné ;
2. regrouper les deux LV dans le même VG, mais les stocker sur des disques dédiés ;
3. séparer les deux LV dans des VG dédiés et donc les stocker sur des disques dédiés.

Les solutions 1 et 3 sont pertinentes, en revanche la seconde proposition est hybride et, bien que possible, elle est plus difficile à justifier et ne sera pas présentée.



Situation cible 1 : 2 PV, 2 VG

Situation cible 2 : 1 PV, 1 VG

Figure 23 – Réorganisation des disques d'une machine, situations cibles possibles

L'exemple présentera le passage de la situation initiale 1 à la situation cible 1 (2 VG) et de la situation initiale 2 à la situation cible 2 (1 VG).

6.5.2 Séparation en deux groupes de volumes

Le principe est le suivant :

- le disque `sdc` sera agrandi pour y transférer les données de `sdb1` ;
- un disque `sdd` sera ajouté pour y transférer les données de `sdb2` ;
- le disque `sdb` sera ensuite retiré du système.

Procédure :

1. Analyser la configuration de la machine avec les commandes `lsblk`, `pvs`, `vgs` et `lvs`.
 - a. La commande `lsblk` donne une vision d'ensemble de la situation. Les disques `sdb` et `sdc` sont utilisés par LVM et contiennent les volumes logiques `lvDemo1` et `lvDemo2` qui sont montés, respectivement, sur `/data` et `/storage` :

```

root@lucario:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   50G  0 disk
├─sda1                               8:1    0   500M  0 part /boot
└─sda2                               8:2    0  49,5G  0 part
   ├─centos-swap                     253:0    0    2G   0 lvm  [SWAP]
   └─centos-root                     253:1    0  47,5G  0 lvm  /
sdb                                  8:16    0   150G  0 disk
├─sdb1                               8:17    0   100G  0 part
│ └─vgDemo1-lvDemo1                 253:2    0   150G  0 lvm  /data
└─sdb2                               8:18    0    50G   0 part
   └─vgDemo2-lvDemo2                 253:3    0    50G   0 lvm  /storage
sdc                                  8:32    0    50G   0 disk
└─vgDemo1-lvDemo1                   253:2    0   150G  0 lvm  /data

```

```
sr0          11:0      1 1024M  0 rom
```

- b. La commande `pvs` liste les volumes physiques et le groupe de volumes associé. Les volumes physiques `sdb1` et `sdc` sont affectés au groupe de volumes `vgDemo1` et le volume physique `sdb2` au groupe `vgDemo2` :

```
root@lucario:~# pvs
PV          VG          Fmt Attr PSize   PFree
/dev/sda2   centos      lvm2 a--  <49,51g 44,00m
/dev/sdb1   vgDemo1     lvm2 a--  <100,00g  0
/dev/sdb2   vgDemo2     lvm2 a--  <50,00g  0
/dev/sdc    vgDemo1     lvm2 a--  <50,00g  0
```

- c. La commande `vgs` liste les groupes de volumes présents. Les groupes de volumes `vgDemo1` et `vgDemo2` sont présents :

```
root@lucario:~# vgs
VG          #PV #LV #SN Attr   VSize   VFree
centos      1  2  0 wz--n- <49,51g 44,00m
vgDemo1     2  1  0 wz--n- 149,99g  0
vgDemo2     1  1  0 wz--n- <50,00g  0
```

- d. La commande `lvs` liste les volumes logiques et le groupe de volumes associé. Les volumes `lvDemo1` et `lvDemo2` sont affectés respectivement aux groupes de volumes `vgDemo1` et `vgDemo2` :

```
root@lucario:~# lvs
LV          VG          Attr          LSize   Pool Origin Data%  Meta%  Move L
og Cpy%Sync Convert
root       centos      -wi-ao----   47,46g
swap       centos      -wi-ao----    2,00g
lvDemo1    vgDemo1     -wi-ao----  149,99g
lvDemo2    vgDemo2     -wi-ao----   <50,00g
```

- Agrandir le disque `sdc` de 100 Go afin qu'il puisse contenir l'ensemble des données de `lvDemo1` et ajouter un disque `sdd` de 50 Go pour contenir `lvDemo2`.
- Rescanner le disque `sdc` pour prendre en compte sa nouvelle taille :

```
root@lucario:~# echo 1 > /sys/class/block/sdc/device/rescan
```

- Rescanner les disques de la machine pour détecter `sdc` :

```
root@lucario:~# grep mpt /sys/class/scsi_host/host?/proc_name
/sys/class/scsi_host/host0/proc_name:mptspi
root@lucario:~# echo "- - -" > /sys/class/scsi_host/host0/scan
```

- Étendre le volume physique `sdc` :

```
root@lucario:~# pvresize /dev/sdc
Physical volume "/dev/sdc" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
```

6. Vérifier que la modification a bien été prise en compte avec `pvs`. Le volume `sdc` a une taille de 150 Go, dont 100 Go de libres :

```
root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2   centos  lvm2 a--  <49,51g 44,00m
/dev/sdb1   vgDemo1 lvm2 a--  <100,00g 0
/dev/sdb2   vgDemo2 lvm2 a--  <50,00g 0
/dev/sdc    vgDemo1 lvm2 a--  <150,00g 100,00g
```

7. Déplacer les extents du volume physique `sdb1` vers le volume physique `sdc` :

```
root@lucario:~# pvmove /dev/sdb1 /dev/sdc
/dev/sdb1: Moved: 0,01%
/dev/sdb1: Moved: 2,51%
/dev/sdb1: Moved: 5,01%
[...]
/dev/sdb1: Moved: 99,30%
/dev/sdb1: Moved: 100,00%
```

8. Vérifier que le volume physique `sdb1` ne contient plus aucune donnée, c'est-à-dire plus aucun extent alloué :

```
root@lucario:~# pvdisplay /dev/sdb1 -m
--- Physical volume ---
PV Name           /dev/sdb1
VG Name           vgDemo1
PV Size           100,00 GiB / not usable 4,00 MiB
Allocatable       yes
PE Size           4,00 MiB
Total PE          25599
Free PE           25599
Allocated PE      0
PV UUID           uykjZq-bAce-yVrA-Tawj-v7do-9dZH-xFvpaz

--- Physical Segments ---
Physical extent 0 to 25598:
FREE
```

9. Retirer le volume physique `sdb1` du groupe de volumes :

```
root@lucario:~# vgreduce vgDemo1 /dev/sdb1
Removed "/dev/sdb1" from volume group "vgDemo1"
```

10. Supprimer le volume physique `sdb1` de LVM :

```
root@lucario:~# pvremove /dev/sdb1
Labels on physical volume "/dev/sdb1" successfully wiped.
```

11. Vérifier que le volume physique `sdb1` ne fait plus partie de la liste :

```
root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2   centos  lvm2 a--  <49,51g 44,00m
```

```
/dev/sdb2  vgDemo2  lvm2  a--  <50,00g  0
/dev/sdc   vgDemo1  lvm2  a--  <150,00g  4,00m
```

12. Créer un nouveau volume physique *sdd* :

```
root@lucario:~# pvcreate /dev/sdd
Physical volume "/dev/sdd" successfully created.
```

13. Ajouter le volume physique *sdd* au groupe de volumes *vgDemo2* :

```
root@lucario:~# vgextend vgDemo2 /dev/sdd
Volume group "vgDemo2" successfully extended
```

14. Déplacer les extents du volume physique *sdb2* vers le volume physique *sdd* :

```
root@lucario:~# pvmove /dev/sdb2 /dev/sdd
/dev/sdb2: Moved: 0,01%
/dev/sdb2: Moved: 5,46%
/dev/sdb2: Moved: 11,08%
[...]
/dev/sdb2: Moved: 98,27%
/dev/sdb2: Moved: 100,00%
```

15. Vérifier que le volume physique *sdb2* ne contient plus aucune donnée, c'est-à-dire plus aucun extent alloué :

```
root@lucario:~# pvdisplay /dev/sdb2 -m
--- Physical volume ---
PV Name           /dev/sdb2
VG Name           vgDemo2
PV Size           <50,00 GiB / not usable 3,00 MiB
Allocatable       yes
PE Size           4,00 MiB
Total PE          12799
Free PE           12799
Allocated PE      0
PV UUID           MUVQF0-oCov-38nL-xZqi-ARJO-AI2M-TisvZH

--- Physical Segments ---
Physical extent 0 to 12798:
FREE
```

16. Retirer le volume physique *sdb2* du groupe de volumes :

```
root@lucario:~# vgreduce vgDemo2 /dev/sdb2
Removed "/dev/sdb2" from volume group "vgDemo2"
```

17. Supprimer le volume physique *sdb2* de LVM :

```
root@lucario:~# pvremove /dev/sdb2
Labels on physical volume "/dev/sdb2" successfully wiped.
```

18. Vérifier que le volume physique *sdb2* ne fait plus partie de la liste :

```

root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2   centos  lvm2 a--  <49,51g 44,00m
/dev/sdc    vgDemo1 lvm2 a--  <150,00g 4,00m
/dev/sdd    vgDemo2 lvm2 a--  <50,00g  0

```

19. Supprimer le disque *sdb* de la machine, il n'est plus utilisé.
20. Redémarrer la machine pour que les modifications soient correctement prises en compte : les disques *sdc* et *sdd* sont automatiquement renommés par le système en *sdb* et *sdc*. Cela ne perturbe en rien le fonctionnement de LVM.
21. Vérifier avec `lsblk` que l'ensemble des disques sont bien reconnus et points de montages opérationnels :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0   500M  0 part /boot
└─sda2                8:2    0   49,5G  0 part
   ├─centos-swap      253:0    0    2G  0 lvm  [SWAP]
   └─centos-root      253:1    0   47,5G  0 lvm  /
sdb                  8:16    0   150G  0 disk
└─vgDemo1-lvDemo1    253:2    0   150G  0 lvm  /data
sdc                  8:32    0    50G  0 disk
└─vgDemo2-lvDemo2    253:3    0    50G  0 lvm  /storage
sr0                  11:0    1  1024M  0 rom

```

La configuration des disques de la machine est désormais bien plus cohérente et simple à comprendre. Grâce à LVM, les opérations ont pu se faire à chaud, en dehors du redémarrage final permettant de réétiqueter les disques.

6.5.3 Regroupement en un seul groupe de volumes

Le principe est le suivant :

- le disque *sdc* sera agrandi pour y transférer les données de *sdb1* et *sdb2* ;
- le disque *sdb* sera ensuite retiré du système.

Procédure :

1. Analyser la configuration de la machine avec les commandes `lsblk`, `pvs`, `vgs` et `lvs`.
 - a. La commande `lsblk` donne une vision d'ensemble de la situation. Les disques *sdb* et *sdc* sont utilisés par LVM et contiennent les volumes logiques *lvDemo1* et *lvDemo2* qui sont montés, respectivement, sur */data* et */storage* :

```

root@lucario:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0   500M  0 part /boot
└─sda2                8:2    0   49,5G  0 part
   ├─centos-swap      253:0    0    2G  0 lvm  [SWAP]
   └─centos-root      253:1    0   47,5G  0 lvm  /
sdb                  8:16    0   150G  0 disk
├─sdb1                8:17    0   100G  0 part
└─vgDemo1-lvDemo1    253:2    0   150G  0 lvm  /data
sdc                  8:18    0    50G  0 part

```

```

└─vgDemo1-lvDemo2 253:3    0   50G  0 lvm  /storage
sdc                8:32    0   50G  0 disk
└─vgDemo1-lvDemo1 253:2    0  150G  0 lvm  /data
sr0                11:0     1 1024M  0 rom

```

- b. La commande `pvs` liste les volumes physiques et le groupe de volumes associé. Les volumes physiques `sdb1`, `sdb2` et `sdc` sont affectés au groupe de volumes `vgDemo1` :

```

root@lucario:~# pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sda2  centos  lvm2 a--   <49,51g 44,00m
/dev/sdb1  vgDemo1 lvm2 a--   <100,00g  0
/dev/sdb2  vgDemo1 lvm2 a--   <50,00g  0
/dev/sdc   vgDemo1 lvm2 a--   <50,00g  0

```

- c. La commande `vgs` liste les groupes de volumes présents. Le groupe de volumes `vgDemo1` est présent :

```

root@lucario:~# vgs
VG      #PV #LV #SN Attr   VSize   VFree
centos   1  2  0 wz--n- <49,51g 44,00m
vgDemo1  2  1  0 wz--n- <200,00g  0

```

- d. La commande `lvs` liste les volumes logiques et le groupe de volumes associé. Les volumes `lvDemo1` et `lvDemo2` sont affectés au groupe de volumes `vgDemo1` :

```

root@lucario:~# lvs
LV          VG      Attr              LSize   Pool Origin Data%  Meta%  Move  L
og Cpy%Sync Convert
root       centos  -wi-ao-----   47,46g
swap       centos  -wi-ao-----    2,00g
lvDemo1    vgDemo1 -wi-ao-----  149,99g
lvDemo2    vgDemo1 -wi-ao-----   <50,00g

```

- Agrandir le disque `sdc` de 150 Go afin qu'il puisse contenir l'ensemble des données de `lvDemo1` et de `lvDemo2`.
- Rescanner le disque `sdc` pour prendre en compte sa nouvelle taille :

```

root@lucario:~# echo 1 >/sys/class/block/sdc/device/rescan

```

- Étendre le volume physique `sdc` :

```

root@lucario:~# pvresize /dev/sdc
Physical volume "/dev/sdc" changed
1 physical volume(s) resized / 0 physical volume(s) not resized

```

- Vérifier que la modification a bien été prise en compte avec `pvs`. Le volume `sdc` a une taille de 200 Go, dont 150 Go de libres :

```

root@lucario:~# pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sda2  centos  lvm2 a--   <49,51g 44,00m
/dev/sdb1  vgDemo1 lvm2 a--   <100,00g  0

```

```
/dev/sdb2  vgDemo1 lvm2 a-- <50,00g 0
/dev/sdc   vgDemo1 lvm2 a-- <200,00g 150,00g
```

6. Déplacer les extents des volumes physiques *sdb1* et *sdb2* vers le volume physique *sdc* :

```
root@lucario:~# pvmove /dev/sdb1 /dev/sdc
/dev/sdb1: Moved: 0,01%
/dev/sdb1: Moved: 2,51%
/dev/sdb1: Moved: 5,01%
[...]
/dev/sdb1: Moved: 99,30%
/dev/sdb1: Moved: 100,00%

root@lucario:~# pvmove /dev/sdb2 /dev/sdc
/dev/sdb2: Moved: 0,01%
/dev/sdb2: Moved: 5,46%
/dev/sdb2: Moved: 11,08%
[...]
/dev/sdb2: Moved: 98,27%
/dev/sdb2: Moved: 100,00%
```

7. Vérifier que les volumes physiques *sdb1* et *sdb2* ne contiennent plus aucune donnée, c'est-à-dire plus aucun extant alloué :

```
root@lucario:~# pvdisplay /dev/sdb1 -m
--- Physical volume ---
PV Name                /dev/sdb1
VG Name                vgDemo1
PV Size                100,00 GiB / not usable 4,00 MiB
Allocatable           yes
PE Size               4,00 MiB
Total PE              25599
Free PE               25599
Allocated PE          0
PV UUID               uykjZq-bAce-yVrA-Tawj-v7do-9dZH-xFvpaz

--- Physical Segments ---
Physical extent 0 to 25598:
FREE

root@lucario:~# pvdisplay /dev/sdb2 -m
--- Physical volume ---
PV Name                /dev/sdb2
VG Name                vgDemo2
PV Size                <50,00 GiB / not usable 3,00 MiB
Allocatable           yes
PE Size               4,00 MiB
Total PE              12799
Free PE               12799
Allocated PE          0
PV UUID               MUVQF0-oCov-38nL-xZqi-ARJO-AI2M-TisvZH

--- Physical Segments ---
Physical extent 0 to 12798:
FREE
```

8. Retirer les volumes physiques *sdb1* et *sdb2* du groupe de volumes :

```
root@lucario:~# vgreduce vgDemo1 /dev/sdb1
Removed "/dev/sdb1" from volume group "vgDemo1"

root@lucario:~# vgreduce vgDemo1 /dev/sdb2
Removed "/dev/sdb2" from volume group "vgDemo1"
```

9. Supprimer les volumes physiques *sdb1* et *sdb2* de LVM

```
root@lucario:~# pvremove /dev/sdb1
Labels on physical volume "/dev/sdb1" successfully wiped.

root@lucario:~# pvremove /dev/sdb2
Labels on physical volume "/dev/sdb2" successfully wiped.
```

10. Vérifier que les volumes physiques *sdb1* et *sdb2* ne font plus partie de la liste :

```
root@lucario:~# pvs
PV          VG      Fmt Attr PSize   PFree
/dev/sda2   centos  lvm2 a--  <49,51g 44,00m
/dev/sdc    vgDemo1 lvm2 a--  <200,00g 8,00m
```

11. Supprimer le disque *sdb* de la machine, il n'est plus utilisé.
12. Redémarrer la machine pour que les modifications soient correctement prises en compte : le disque *sdc* est automatiquement renommé par le système en *sdb*. Cela ne perturbe en rien le fonctionnement de LVM.
13. Vérifier avec `lsblk` que l'ensemble des disques sont bien reconnus et les points de montages opérationnels :

```
root@lucario:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0   50G  0 disk
├─sda1           8:1    0  500M  0 part /boot
└─sda2           8:2    0  49,5G  0 part
   ├─centos-swap 253:0    0    2G   0 lvm  [SWAP]
   └─centos-root 253:1    0  47,5G  0 lvm  /
sdb              8:16    0  200G  0 disk
├─vgDemo-lvDemo1 253:2    0  150G  0 lvm  /data
└─vgDemo-lvDemo2 253:3    0   50G  0 lvm  /storage
sr0             11:0    1 1024M  0 rom
```

La configuration des disques de la machine est désormais bien plus cohérente et simple à comprendre. Grâce à LVM, les opérations ont pu se faire à chaud, en dehors du redémarrage final permettant de réétiqueter les disques.

7 Mémento des commandes utiles pour LVM

Commande	Description
<code>pvs</code>	Affiche la liste des PV dans un format condensé.
<code>pvdisplay [pvName] [-m]</code>	Affiche la liste des PV, ou un seul, avec leurs informations détaillées. Option « -m » : affiche la répartition des extents du PV entre les différents LV.
<code>pvcreate <devicePath></code>	Crée un PV sur le périphérique indiqué (un disque ou une partition).
<code>pvresize <devicePath></code>	Redimensionne un PV pour occuper tout l'espace disponible sur le périphérique.
<code>pvmove <devicePathSrc> <devicePathDst></code>	Déplace les données (extents) du PV source vers le PV cible. Les deux PV doivent appartenir au même VG et la cible doit avoir suffisamment de place pour recevoir les données. Après cette opération, le PV source est vide et peut être retiré du VG.
<code>pvremove <devicePath></code>	Supprime un PV. Prérequis : il ne doit plus contenir de données et ne plus appartenir à un VG.
<code>vgs</code>	Affiche la liste des VG dans un format condensé.
<code>vgdisplay [vgName]</code>	Affiche la liste des VG, ou un seul, avec leurs informations détaillées.
<code>vgcreate <vgName> <pvName></code>	Crée un VG contenant le PV indiqué.
<code>vgextend <vgName> <pvName></code>	Ajoute le PV indiqué dans un le VG.
<code>vgreduce <vgName> <pvName></code>	Retire le PV indiqué du VG. Le PV ne doit plus contenir de données (utiliser <code>pvmove</code> pour cela).
<code>vgchange [-a y n] <vgName></code>	Permet de modifier des attributs d'un VG. En particulier, l'option « -a » permet d'activer ou de désactiver le VG et tous les LV qu'il contient. Prérequis : pour désactiver un VG, les systèmes de fichiers associés aux LV doivent être démontés. Utilité : nécessaire pour déplacer des LV vers un autre VG (fusion ou séparation de VG).

<code>vgmerge <vgNameDst> <vgNameSrc> [-vt]</code>	<p>Permet de fusionner deux VG en intégrant au VG destination les PV et LV du VG source.</p> <p>Prérequis : le VG source doit être inactif (utiliser <code>vgchange</code> pour cela).</p> <p>Option « -vt » : effectue un test de fusion.</p>
<code>vgsplit -n <lvName> <vgNameSrc> <vgNameDst> [-vt]</code>	<p>Permet de déplacer le LV du VG source vers le VG destination. Tous les PV associés au LV seront migrés. Si le VG destination n'existe pas, il est créé. Si d'autres LV partagent les mêmes PV, ils seront déplacés aussi.</p> <p>Prérequis : le LV doit être inactif (utiliser <code>lvchange</code> pour cela).</p> <p>Option « -vt » : effectue un test de division/déplacement.</p>
<code>vgremove <vgName></code>	Détruit un VG (doit être vide).
<code>lvs</code>	Affiche la liste des LV dans un format condensé.
<code>lvdisplay [lvName]</code>	Affiche la liste des LV, ou un seul, avec leurs informations détaillées.
<code>lvcreate <lvName> <vgName></code>	Crée un LV attaché au VG indiqué.
<code>lvresize <lvName> <pvName></code>	Redimensionne un LV pour occuper tout l'espace du PV indiqué. Le PV doit appartenir au même VG que le LV.
<code>lvchange [-a y n] <lvName></code>	<p>Permet de modifier des attributs d'un LV. En particulier, l'option « -a » permet d'activer ou de désactiver le LV.</p> <p>Prérequis : pour désactiver un LV, le système de fichiers associé doit être démonté.</p> <p>Utilité : nécessaire pour déplacer des LV vers un autre VG (fusion ou séparation de VG).</p>
<code>lvremove <lvName></code>	Détruit un LV.

8 Sources

- <https://www.admin-linux.fr/memo-logical-volume-manager-lvm2/>
- <https://www.linuxtricks.fr/wiki/lvm-sous-linux-volumes-logiques>
- <https://doc.ubuntu-fr.org/lvm>
- <https://www.rootusers.com/how-to-increase-the-size-of-a-linux-lvm-by-expanding-the-virtual-machine-disk/>
- <https://kb.vmware.com/s/article/1006371>
- <http://xmodulo.com/manage-lvm-volumes-centos-rhel-7-system-storage-manager.html>
- <https://blog.tinned-software.net/lvm-remove-physical-volume-from-volume-group/>
- <https://sites.google.com/site/rhelworldexperience/home/rhel-disk-management---tips-and-tricks/mergingvolumeingroupsinrhel-vgmerge>