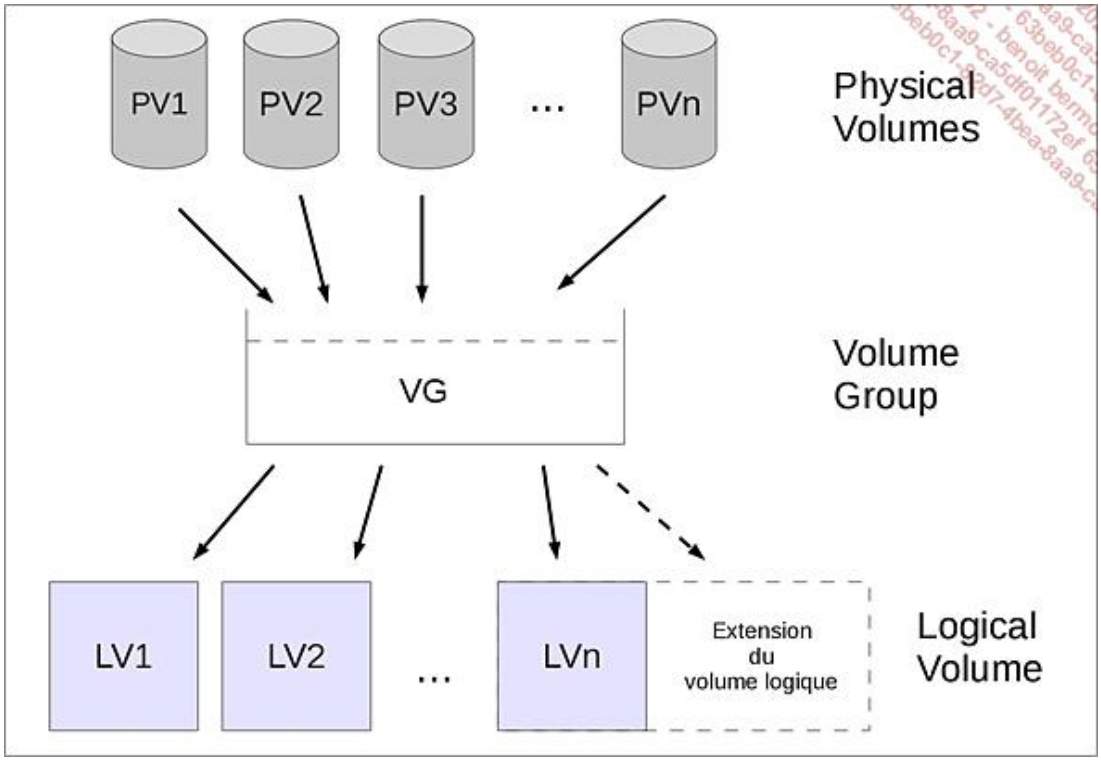
Logical Volume Manager

Le système de partitionnement traditionnel des disques impose certaines limitations comme un nombre de partitions limité à quatre, et le caractère obligatoirement contigu de l'espace partitionné. Si de nombreux utilitaires permettent de redimensionner les partitions à la volée , il reste impossible d'étendre une partition avec de l'espace non contigu, par exemple sur un autre disque dur. Pour pallier ces limitations, la plupart des éditeurs de systèmes d'exploitation ont proposé des gestions d'espaces disque plus ou moins propriétaires, comme les disques dynamiques pour Windows ou les volumes NSS chez Novell. Pour les systèmes Linux, la solution s'appelle Logical Volume Manager (gestionnaire de volumes logiques). Les volumes logiques permettent de créer un nombre illimité de volumes, de les étendre à volonté, y compris à partir d'espace se trouvant sur des disques et des contrôleurs différents.

Il est d'usage de conserver les termes anglais lorsqu'on parle d'éléments LVM, cela aidera notamment à se souvenir facilement des commandes d'exploitation. Certains éléments, comme les Logical Volumes qui supportent une traduction facile et naturelle infirment néanmoins cet usage.

1. Architecture des volumes logiques



Une architecture LVM se compose de **PV** : Physical Volumes, **VG** : Volume Groups et de **LV** : Logical Volumes.

Un volume logique est l'équivalent fonctionnel d'une partition traditionnelle, il est identifié par un fichier spécial en mode bloc, et supportera généralement un filesystem en vue d'un montage.

Les Logical Volumes sont composés de blocs de données, puisés dans une couche d'abstraction appelée Volume Group, elle-même alimentée par des espaces de stockage bruts (disques ou partitions) appelés Physical Volumes.

Dans une architecture LVM basée sur plusieurs volumes physiques, la défaillance du moindre d'entre eux rend tous les volumes logiques qui en dépendent inopérants. Il conviendra donc de ne créer des volumes physiques que depuis des volumes à tolérance de panne comme des éléments soumis à RAID, qu'il soit logiciel ou matériel.

---

2. Commandes LVM

Les commandes de gestion des LVM sont construites selon un préfixe lié à l'objet qu'on veut gérer, et un suffixe selon l'action à entreprendre.

Construction des commandes LVM

préfixe	suffixe	
pv	create	Création d'un élément LVM.
vg	extend	Extension d'un VG ou d'un LV.
lv	reduce	Réduction d'un VG ou d'un LV.
	display	Affichage des informations d'un élément LVM.

a. Création des éléments

On commencera par créer les PV (*physical volumes*) à partir d'espaces de stockage. Il peut s'agir de disques entiers, ou de partitions traditionnelles, dont le type aura été modifié à 8e. Il est à noter que la construction de PV à partir de partitions traditionnelles est généralement réservée à des besoins de test, et qu'un usage en production pour des volumes de données s'appuie presque toujours sur des disques entiers.

Création des volumes physiques

Les volumes physiques sont créés avec la commande **pvcreate**.

Syntaxe de la commande pvcreate

```
pvcreate device
```

Où *device* représente le fichier spécial blocs qui héberge le volume physique, disque ou partition.

Création du groupe de volumes

Les groupes de volumes sont créés avec la commande **vgcreate**.

Syntaxe de la commande pvcreate

```
vgcreate nom_vg pv_device
```

vgcreate : options et paramètres	
nom_vg	Nom du groupe de volume. Valeur au choix.
pv_device	Fichier spécial blocs qui héberge le ou les pv qui alimentent le vg.

Le groupe de volume ainsi créé apparaîtra sous forme de répertoire du nom du groupe de volume créé, directement sous /dev. Attention, ce répertoire n'apparaîtra réellement que lorsqu'un premier volume logique sera créé à partir du groupe de volume.

Création du volume logique

Les volumes logiques sont créés avec la commande **lvcreate**. On peut créer autant de volumes logiques que l'on veut tant qu'il reste de l'espace disponible dans le Volume Group.

Syntaxe de la commande lvcreate

```
lvcreate -L taille -n nom_lv nom_vg
```

lvcreate : options et paramètres	
taille	Taille du volume logique, sous forme de valeur numérique directement suivie de l'unité.
nom_lv	Nom du volume logique. Valeur au choix.

nom_vg	Nom du groupe de volume à partir duquel le volume logique sera créé.

Le volume logique ainsi créé apparaîtra sous forme de fichier spécial en mode blocs dans le répertoire portant le nom de son groupe de volumes sous /dev. C'est ce fichier spécial qui sera employé lors des opérations de montage.

b. Diagnostics LVM

Les architectures LVM sont souvent déroutantes, du fait du grand nombre d'opérations nécessaires pour arriver à la création d'un volume logique. De plus, si on se figure assez bien ce que peut être un volume physique, la nature abstraite du groupe de volume le rend difficile à appréhender. Pour ces raisons, il est essentiel de se faire une idée précise de l'ensemble des éléments utilisés dans une architecture LVM et de les documenter consciencieusement. Par chance, les outils de diagnostics LVM sont précis, et ils permettent à chaque étape de vérifier le bon déroulement des opérations.

Affichage des informations de volume physique

Les informations détaillées de tous les volumes physiques présents sur un système seront affichées par la commande **pvdisplay**. Si vous préférez la concision, vous pouvez essayer **pvs**.

Exemple d'utilisation de la commande pvdisplay

Il est important d'identifier les volumes physiques avec la commande pvdisplay. L'utilitaire fdisk indiquerait un disque sans table des partitions et laisserait à penser qu'on est en présence d'un disque vierge.

```
A:~# pvdisplay
  "/dev/sdb" is a new physical volume of "2,00 GB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               2,00 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               UHSnwO-EKMh-QbDn-1qj0-f7Az-KKkx-3XcyZz
A:~#
```

Exemple d'utilisation de la commande pvs

L'essentiel en deux lignes.

```
A:~# pvs
  PV         VG   Fmt  Attr PSize PFree
  /dev/sdb        lvm2 --   2,00G 2,00G
A:~#
```

Affichage des informations de groupes de volumes

Les informations détaillées de tous les groupes de volumes présents sur un système sont affichées par la commande **vgdisplay**. Si vous préférez la concision, vous pouvez essayer **vgs**.

Exemple d'utilisation de la commande vgdisplay

L'affichage des détails des groupes de volume permet de connaître la taille totale disponible des groupes.

```
A:~# vgdisplay
  --- Volume group ---
  VG Name               vg1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
```

```
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               2,00 GB
  PE Size               4,00 MB
  Total PE              511
  Alloc PE / Size       0 / 0
  Free  PE / Size       511 / 2,00 GB
  VG UUID               D6QwUK-Lltf-uGg5-vH8r-ZmaK-dU0L-Lyyu3T
A:~#
```

Exemple d'utilisation de la commande vgs

```
A:~# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg1    1   0   0 wz--n- 2,00G 2,00G
A:~#
```

Affichage des informations de volumes logiques

Les informations détaillées de tous les volumes logiques présents sur un système seront affichées par la commande **lvdisplay**. Pour la concision, essayez **lvs**.

Exemple d'utilisation de la commande lvdisplay

```
A:~# lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg1/data1
  VG Name                vg1
  LV UUID                Ll7105-aLpz-axKC-Hcuq-pPSq-QZaK-8h5PLC
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                400,00 MB
  Current LE             100
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
A:~#
```

Exemple d'utilisation de la commande lvs

```
A:~# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Log Copy%  Convert
  data1 vg1  -wi-a- 400,00M
A:~#
```

c. Extension de volume logique

Un des principaux avantages des volumes logiques est l'extension facile des volumes logiques. Nous avons vu qu'un volume logique est constitué de Logical Extents fournis par un objet Volume Group. Si des Logical Extents sont encore disponibles dans le Volume Group, il est alors facile d'étendre le Logical Volume à partir de ces Logical Extents. En clair, s'il reste de l'espace non affecté dans le groupe de volume, on peut l'ajouter à un volume logique déjà créé. Dans le cas contraire, il faudra d'abord étendre le Volume Group en y ajoutant un ou plusieurs Physical Volumes.

Extension d'un Volume Group

L'extension d'un Volume Group se fait à partir de Physical Volume(s) avec la commande **pvextend**. Les Physical

Volumes sont alors créés comme précédemment avec la commande **pvcreate**.

Syntaxe de la commande vgextend

```
vgextend nom_vg pv_device
```

vgcreate : options et paramètres	
nom_vg	Nom du groupe de volume à étendre.
pv_device	Fichier spécial blocs qui héberge le ou les PV qui alimentent le VG.

Extension d'un Logical Volume

L'extension d'un Logical Volume se fait avec la commande **lvextend**.

Syntaxe de la commande lvextend

```
lvextend -L taille lv
```

lvcreate : options et paramètres	
taille	Taille du volume logique étendu, sous forme de valeur numérique directement suivie de l'unité. Si la taille est précédée d'un signe +, cette taille s'ajoute à celle du volume existant.
lv	Volume logique à étendre, désigné par son fichier spécial en mode blocs.

Un Logical Volume n'est qu'un espace de stockage, indépendamment du filesystem qui y est apposé. En cas d'extension du Logical Volume, il faudra prévoir d'étendre aussi le filesytem pour pouvoir exploiter l'espace supplémentaire.

d. Réduction de LV

La réduction des éléments LVM est possible, même si ce genre de manœuvre est toujours délicate et doit être bien maîtrisée.

Réduction d'un Logical Volume

La réduction d'un volume logique se fait avec la commande **lvreduce**. Les Logical Extent sont retirés dès l'exécution de la commande et toutes les données s'y trouvant sont perdues. Toutes les précautions devront donc être prises pour éviter des pertes de données.

Réduction d'un LV

```
lvreduce -L taille lv
```

lvreduce : options et paramètres	
taille	Taille à retirer du volume logique étendu, sous forme de valeur numérique directement suivie de l'unité.
lv	Volume logique à réduire, désigné par son fichier spécial en mode blocs.

Réduction d'un Volume Group

Un Volume Group peut être réduit par la commande **vgreduce**.

Réduction d'un VG

```
vgreduce vg pv
```

vgreduce : options et paramètres	
vg	Le groupe de volume à réduire.
pv	Le (ou les) volume(s) physique(s) à retirer du groupe de volumes.

3. Exploitation des volumes logiques

a. Données sur les volumes logiques

Une fois les Logical Volumes créés, il faut pour les exploiter y apposer un filesystem. Il faut bien comprendre que d'un point de vue fonctionnel, les volumes logiques sont le strict équivalent des partitions traditionnelles directement créées avec fdisk, et de type Linux. La démarche sera donc strictement identique à celle employée en partitionnement traditionnel, si ce n'est que le fichier spécial en mode bloc sera celui du Logical Volume.

Exemple de création d'un file system ext3 sur un LV

Les volumes logiques supportent la création de filesystem comme les partitions traditionnelles. Notez le fichier de bloc spécial sous lequel le volume logique est reconnu.

```
A:~# mke2fs -j /dev/vg1/lv99
mke2fs 1.41.3 (12-Oct-2008)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=1024 (log=0)
Taille de fragment=1024 (log=0)
25688 i-noeuds, 102400 blocs
5120 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=1
Nombre maximum de blocs du système de fichiers=67371008
13 groupes de blocs
8192 blocs par groupe, 8192 fragments par groupe
1976 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
  8193, 24577, 40961, 57345, 73729

Écriture des tables d'i-noeuds : complété
Création du journal (4096 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

Le système de fichiers sera automatiquement vérifié tous les 31 montages ou
après 180 jours, selon la première éventualité. Utiliser tune2fs -c ou -i
pour écraser la valeur.
A:~#
```

De même, il sera nécessaire pour exploiter ce filesystem de monter le volume logique, que ce soit de façon manuelle ou par le biais du fichier **/etc/fstab**.

Exemple de montage de volume logique

```
A:/mnt# mount /dev/vg1/lv99 /mnt/data99
A:/mnt#
```

b. Exploitation du snapshot LVM pour les sauvegardes

La nature souple et évolutive des LVM les rend parfaitement aptes à stocker de grands volumes de données. Or, un problème récurent se pose lors de la sauvegarde de ces gros volumes de données. En effet, le temps nécessaire à la sauvegarde interdit souvent de réaliser les opérations hors ligne. La solution est apportée par la fonctionnalité de snapshot (instantané) disponible sur les architectures LVM.

On réalise le snapshot du volume logique à sauvegarder alors qu'il est monté et en exploitation, et on effectue la sauvegarde sur le snapshot qui est une copie conforme du volume logique au moment précis où il a été réalisé. Il faut bien comprendre qu'un snapshot n'est pas un outil de sauvegarde en tant que tel, mais un moyen au service d'une stratégie de sauvegarde.

Réalisation du snapshot

Le snapshot se fait avec la commande **lvcreate**. Un snapshot est donc un volume logique à part entière, et il pourra être monté et exploité en cas de besoin.

Il faudra déterminer la taille du snapshot lors de sa création. Le volume logique de snapshot ne stocke physiquement que les différences entre le volume en production (celui qui a été snapshoté) et le volume de snapshot. S'il n'y a pas d'écritures réalisées sur le volume en production, la consommation en espace de stockage pour le snapshot sera quasi nulle. Si toutes les données sont modifiées sur le volume en production, le snapshot exploitera physiquement un espace disque de l'ordre de celui consommé par le volume de données au moment du snapshot. L'espace exploité par le snapshot pourra être surveillé avec la commande **lvdisplay**.

Syntaxe de la commande lvcreate pour la création de snapshot

```
lvcreate -L taille -s -n nom_snapshot lv_origine
```

lvcreate pour snapshot : options et paramètres	
-L *taille*	Taille du snapshot à créer.
-s	Option qui indique qu'on crée un snapshot de volume logique, et non un volume logique ordinaire.
-n *nom_snapshot*	Le nom du volume de snapshot. Il est recommandé d'avoir une convention de dénomination explicite.
lv_origine	Le nom du volume logique en production à partir duquel le snapshot sera réalisé.

Exemple de création de snapshot

Le snapshot est un volume logique presque comme les autres.

```
A:/mnt# lvcreate -L 1G -s -n clicclac /dev/vg1/data1
  Logical volume "clicclac" created
A:/mnt#
```

Exemple de visualisation de l'espace disque réellement occupé par un snapshot

Dans l'exemple ci-dessous, les données n'ont pas été modifiées sur le volume d'origine entre le lvcreate -s et le lvdisplay. On observe donc la valeur "Allocated to snapshot" à 0%.

```
A:/mnt# lvdisplay /dev/vg1/clicclac
  --- Logical volume ---
  LV Name                /dev/vg1/clicclac
  VG Name                vg1
  LV UUID                xyakf0-2zMf-B3qG-S9gT-KTqw-ZJI3-W06GWi
  LV Write Access        read/write
  LV snapshot status     active destination for /dev/vg1/data1
  LV Status              available
  # open                 0
  LV Size                1,49 GB
  Current LE             381
  COW-table size         1,00 GB
  COW-table LE           256
  Allocated to snapshot  0,00%
  Snapshot chunk size    4,00 KB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
```

```
  Block device           253:1
A:/mnt#
```

Dans ce deuxième exemple, des données ont été ajoutées sur le volume d'origine, obligeant le système à conserver deux versions : les données snapshotées, disponibles pour la sauvegarde, et les données nouvelles écrites sur le disque et affectées au volume en production. La valeur "Allocated to snapshot" est désormais à 1,45 %.

```
A:/mnt/data1# lvdisplay /dev/vg1/clicclac
  --- Logical volume ---
  LV Name                /dev/vg1/clicclac
  VG Name                vg1
  LV UUID                xyakf0-2zMf-B3qG-S9gT-KTqw-ZJI3-W06GWi
  LV Write Access        read/write
  LV snapshot status     active destination for /dev/vg1/data1
  LV Status              available
  # open                 0
  LV Size                1,49 GB
  Current LE             381
  COW-table size         1,00 GB
  COW-table LE           256
  Allocated to snapshot  1,45%
  Snapshot chunk size    4,00 KB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1
A:/mnt/data1#
```

Sauvegarde des données snapshotées

Du point de vue des LVM, il n'y a plus rien à faire. Les données sont disponibles, figées dans le temps au moment où le snapshot a été réalisé, et elles sont sauvegardables par n'importe quel moyen usuel.

Exemple de sauvegarde des données snapshotées

Dans cet exemple, on monte le volume logique de snapshot dans un répertoire /mnt/clicclac, et on réalise une archive tar compressée des données que l'en stocke sur un périphérique USB.

```
A:/mnt# mkdir clicclac
A:/mnt# mount /dev/vg1/clicclac clicclac
A:/mnt# ls clicclac
bigfile.tar  etc  growingfile  lost+found  midfile.tar usr
A:/mnt# tar czf /media/usb/svg_snap.tgz /mnt/clicclac
tar: Suppression de « / » au début des noms des membres
A:/mnt#
```