



Logical Volume Management (LVM)

Serveur Linux CentOS

Table des matières

1	Introduction	3
2	Implémentation.....	5
3	Ajouter un disque	6
4	Créer des partitions.....	8
5	Créer un volume physique (PV).....	11
6	Créer un groupe de volumes(VG)	14
7	Créer un volume logique (LV)	16
8	Créer un système de fichiers (FileSystem).....	19
9	Monter un système de fichiers (FileSystem).....	20
10	Redimensionner un VG	21
11	Redimensionner un LV	24
12	Redimensionner un système de fichiers (FileSystem)	25
13	Outil graphique.....	26

1 Introduction

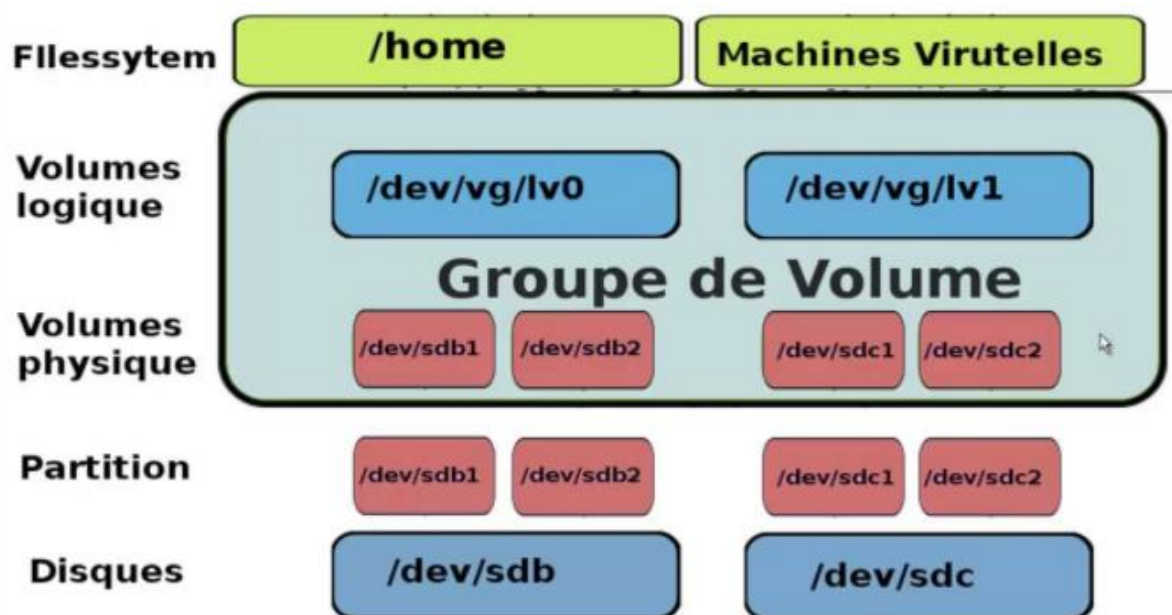
Les schémas de partitionnement courants divisent les disques en sections de taille fixe, cela présente deux inconvénients :

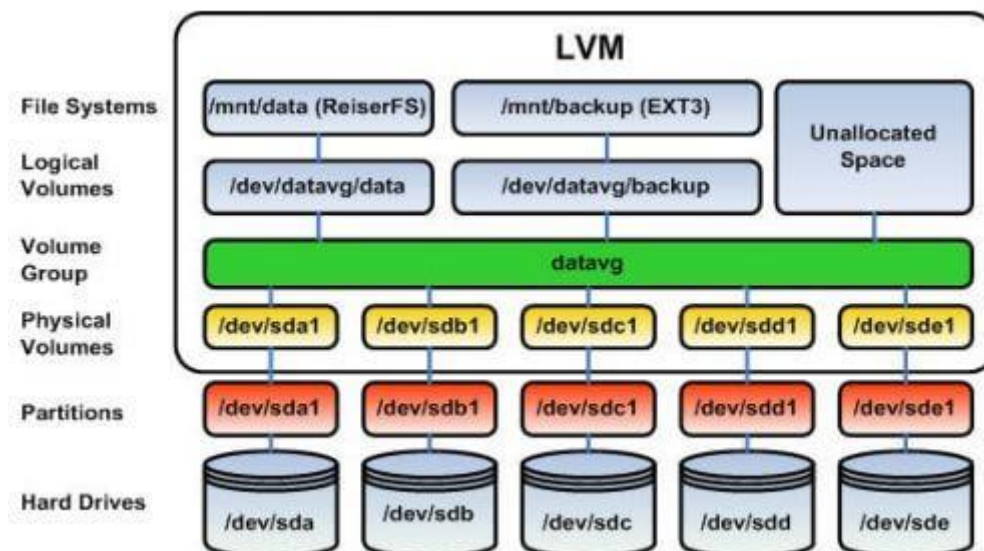
- La taille des partitions étant déterminée à la création, il est quasiment impossible de déplacer ou de redimensionner une partition.
- La taille d'une partition est limitée par la capacité du disque sur lequel elle se trouve.
- Le nombre maximum de partition est de 4.

Pour se soustraire à ses limitations, un niveau supplémentaire d'abstraction entre les périphériques de stockage et les unités de bloc a été introduit :

Le LVM (Logical Volume Manager).

Contrairement au partitionnement classique, le LVM considère les disques ou partitions comme un ensemble commun d'unité de stockage. Ces unités constituent des volumes logiques qui ne sont plus limités par la capacité d'un seul disque et qui peuvent être redimensionnés dynamiquement par l'ajout ou la suppression d'une unité.





Volume Physique ou PV

Désigne les partitions et autres disques physiquement présents sur le système.

Groupe de Volumes ou VG

Ensemble de volumes physiques qui peut être considéré comme un grand disque virtuel. Sa taille est égale à la somme des capacités des volumes physiques qui le composent.

Domaine physique ou PE (Physical Extent)

Unité de stockage atomique au niveau de LVM ayant pour taille 4 Mo par défaut.

Volume Logique ou LV

Partition virtuelle obtenue en regroupant plusieurs domaines physiques d'un même groupe de volumes. Les volumes logiques peuvent être utilisés comme toute autre unité de type bloc.

Un volume logique peut donc être repartitionné physiquement sur plusieurs disques et il est possible de redimensionner celui-ci par l'ajout ou la suppression de domaines physiques.

De plus, il est également possible de redimensionner un groupe de volumes par l'ajout ou la suppression d'un volume physique.

Cette architecture permet donc de créer des partitions logiques utilisables comme toute autre unité de type bloc.

2 Implémentation

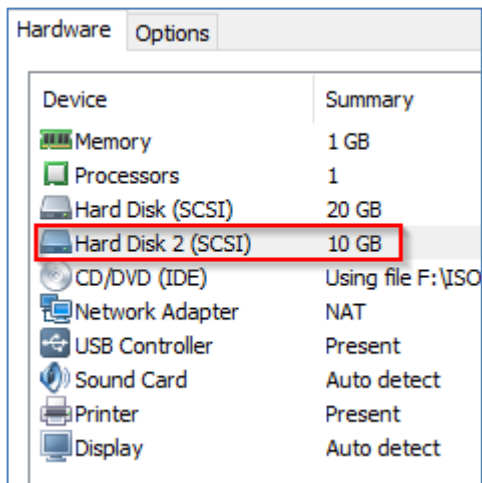
Pour mettre en place LVM sous Linux, il faudra :

- Ajouter les disques;
- Partitionner les disques;
- Identifier et créer les volumes physiques à utiliser;
- Définir le groupe de volume;
- Partitionner le groupe de volumes en volumes logiques;
- Monter les volumes logiques;

L'outil de partitionnement automatique du programme d'installation de CentOS crée le groupe de volumes **centos** à partir de l'espace disque disponible. Les volumes logiques **/dev/centos/root** et **/dev/centos/swap** sont définis dans ce groupe de volumes pour le système de fichiers racine et l'espace de pagination système.

3 Ajouter un disque

Avant de démarrer le serveur, ajouter un disque de 10 GB :



Les disques SATA ou SCSI sont `/dev/sdx` où `x` est une lettre représentant le support physique. Par exemple `/dev/sda` représente le premier disque et `/dev/sdb` le deuxième disque.

Les partitions d'un disque SATA ou SCSI sont nommées `sdx` où `x` est une lettre représentant le support physique et `y` est le numéro de la partition.

Par exemple `/dev/sda2` représente la deuxième partition du premier disque.

Les partitions d'un disque IDE sont nommées `hdx` où `x` est une lettre représentant le support physique et `y` est le numéro de la partition.

`/dev/hda1` représente la première partition du premier disque.

Pour afficher la liste des disques et des partitions, il faut lancer la commande **fdisk -l**

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000b7864

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1    *        2048     2099199     1048576   83   Linux
/dev/sda2           2099200     41943039     19921920   8e   Linux LVM

Disk /dev/sdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x27108166

Device Boot          Start          End      Blocks   Id  System

Disk /dev/mapper/centos-root: 18.2 GB, 18249416704 bytes,
35643392 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-swap: 2147 MB, 2147483648 bytes,
4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

4 Créer des partitions

On va créer deux partitions de 5 GB :

```
[root@localhost ~]#
(echo n;echo p;echo 1;echo "";echo +5G;echo w;)|fdisk /dev/sdb

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write
them.
Be careful before using the write command.
Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x27108166.

Command (m for help): Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): Partition number (1-4, default 1): First
sector (2048-20971519, default 2048): Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default
20971519):
Partition 1 of type Linux and of size 5 GiB is set

Command (m for help): The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@localhost ~]#
(echo n;echo p;echo 2;echo "";echo "";echo w;)|fdisk /dev/sdb

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write
them.
Be careful before using the write command.
Command (m for help): Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): Partition number (2-4, default 2): First
sector (10487808-20971519, default 10487808): Using default
value 10487808
Last sector, +sectors or +size{K,M,G} (10487808-20971519,
default 20971519): Using default value 20971519
Partition 2 of type Linux and of size 5 GiB is set

Command (m for help): The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```


Changer le type de partition du deuxième disque pour *Linux LVM* :

```
[root@localhost ~]#  
(echo t;echo 1;echo "8e";echo w;)|fdisk /dev/sdb  
  
Welcome to fdisk (util-linux 2.23.2).  
  
Changes will remain in memory only, until you decide to write  
them.  
Be careful before using the write command.  
  
Command (m for help): Partition number (1,2, default 2): Hex  
code (type L to list all codes):  
Changed type of partition 'Linux' to 'Linux LVM'  
  
Command (m for help): The partition table has been altered!  
  
Calling ioctl() to re-read partition table.  
Syncing disks.
```

```
[root@localhost ~]#  
(echo t;echo 2;echo "8e";echo w;)|fdisk /dev/sdb  
  
Welcome to fdisk (util-linux 2.23.2).  
  
Changes will remain in memory only, until you decide to write  
them.  
Be careful before using the write command.  
  
Command (m for help): Partition number (1,2, default 2): Hex  
code (type L to list all codes):  
Changed type of partition 'Linux' to 'Linux LVM'  
  
Command (m for help): The partition table has been altered!  
  
Calling ioctl() to re-read partition table.  
Syncing disks.
```

Afficher les partitions du deuxième disque :

```
root@localhost ~]# fdisk -l /dev/sdb
```

Disk /dev/sdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x27108166

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	10487807	5242880	8e	Linux LVM
/dev/sdb2		10487808	20971519	5241856	8e	Linux LVM

```
[root@localhost ~]#
```

5 Créer un volume physique (PV)

Pour créer le premier volume physique sur la partition sur `/dev/sdb1` :

```
[root@localhost ~]# pvcreate /dev/sdb1  
  
Physical volume "/dev/sdb1" successfully created.
```

Pour créer le premier volume physique sur la partition sur `/dev/sdb2` :

```
[root@localhost ~]# pvcreate /dev/sdb2  
  
Physical volume "/dev/sdb2" successfully created.
```

Afficher la liste sommaire des PV :

```
[root@localhost ~]# pvscan  
  
PV /dev/sda2    VG centos          lvm2 [<19.00 GiB / 0    free]  
PV /dev/sdb2    lvm2 [<5.00 GiB]  
PV /dev/sdb1    lvm2 [5.00 GiB]  
Total:3[<29.00 GiB]/in use:1[<19.00 GiB]/ n no VG: 2 [<10.00 GiB]
```

Afficher la liste détaillée des PV :

```
[root@localhost ~]# pvdisplay

--- Physical volume ---
PV Name                /dev/sda2
VG Name                centos
PV Size                <19.00 GiB / not usable 3.00 MiB
Allocatable            yes (but full)
PE Size                4.00 MiB
Total PE               4863
Free PE                0
Allocated PE           4863
PV UUID                fpqepY-j8Lo-Lj12-Lo9V-cG11-iu5Z-HZP7NQ

"/dev/sdb2" is a new physical volume of "<5.00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdb2
VG Name
PV Size                <5.00 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                pB9tSM-dYVJ-OWGO-lpvD-GsKL-s3Lg-zfXHZL

"/dev/sdb1" is a new physical volume of "5.00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdb1
VG Name
PV Size                5.00 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                cKZAkB-KkFs-HSKv-3M6i-99IR-0JAv-MKptji
```

Différentes commandes permettent de manipuler les volumes physiques :

La commande **pvcreate** permet de formater un ou plusieurs volumes physiques avant de les intégrer à un groupe de volumes;

La commande **pvmove** permet de déplacer les domaines physiques d'un volume physique vers un autre, en vue de le supprimer du groupe de volumes par exemple.

La commande **pvdisplay** permet d'obtenir des informations sur les volumes physiques.

La commande **pvchange** permet de modifier les attributs d'un volume physique.

Les noms des volumes physiques peuvent être soit des disques (comme **/dev/sda**) soit des partitions (**/dev/sda1**). Ces dernières doivent être de type 8e ('Linux LVM' sous *fdisk*).

6 Créer un groupe de volumes(VG)

La commande qui permet de créer un groupe de volumes est la suivante :

vgcreate[-s <taille des PE>] <VG> <PV> [<PV>....]

L'option **-v** permet d'augmenter la verbosité de la commande *vgcreate*.

Ainsi pour créer un groupe de volumes nommé **VG_DATA** avec les deux volumes physiques précédents :

```
[root@localhost ~]# vgcreate VG_DATA /dev/sdb1 /dev/sdb2  
  
Volume group "VG_DATA" successfully created
```

Afficher la liste sommaire des VG :

```
[root@localhost ~]# vgscan  
Reading volume groups from cache.  
Found volume group "centos" using metadata type lvm2  
Found volume group "VG_DATA" using metadata type lvm2
```

Afficher la liste détaillée des VG :

```
[root@localhost ~]# vgdisplay  
  
--- Volume group ---  
VG Name                centos  
System ID  
Format                 lvm2  
Metadata Areas         1  
Metadata Sequence No   3  
VG Access              read/write  
VG Status              resizable  
MAX LV                 0  
Cur LV                2  
Open LV               2  
Max PV                 0  
Cur PV                1  
Act PV                1  
VG Size                <19.00 GiB  
PE Size                4.00 MiB  
Total PE              4863  
Alloc PE / Size        4863 / <19.00 GiB  
Free PE / Size         0 / 0  
VG UUID                BIIJDD-Kee4-lihY-91Ra-1pjA-qAQy-qd6HDZ  
  
--- Volume group ---  
VG Name                VG_DATA  
System ID  
Format                 lvm2  
Metadata Areas         2
```

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	2
Act PV	2
VG Size	9.99 GiB
PE Size	4.00 MiB
Total PE	2558
Alloc PE / Size	0 / 0
Free PE / Size	2558 / 9.99 GiB
VG UUID	IuwOtt-Qnto-unpe-iZLY-QX0A-z2qT-7UCdNj

Pour manipuler les groupes de volumes et changer leurs attributs :

La commande **vgchange** permet de rendre disponible le groupe de volumes en vue de l'activer ou indisponible en vue de le désactiver; cette opération fait partie des scripts de démarrage du système.

La commande **vgexport** permet de désactiver un groupe de volumes dans le but de déplacer ses disques vers une autre machine.

La commande **vgimport** permet d'activer un groupe de volumes contenu sur des disques provenant d'une autre machine.

La commande **vgremove** permet de supprimer un groupe de volumes.

Il est possible aussi de redimensionner les groupes de volumes avec :

La commande **vgextend** permet d'ajouter un volume physique au groupe de volumes.

La commande **vgreduce** permet de supprimer un volume physique au groupe de volumes.

7 Créer un volume logique (LV)

La dernière étape pour la mise en place de LVM sous Linux est la création des volumes logiques avec la commande :

lvcreate -L <taille> [-n <LV>] <VG> [<PV>....]

Cette commande crée une unité de bloc accessible via `/dev/<VG>/<LG>` dont la taille sera arrondie au nombre de domaines physiques le plus proche.

L'activation du support du pilote '*Multiple devices*' dans le noyau 2.6 permet aussi d'accéder à l'unité de bloc créée avec **lvcreate** sous le nom de fichier `/dev/mapper/<VG>-<LV>`

On va créer deux volumes logiques **lv_info** et **lv_finance** de 2 GB chaque pour les départements d'informatique de finance respectivement :

```
[root@localhost ~]# lvcreate -L 2000 VG_DATA -n lv_info
Logical volume "lv_info" created.
```

```
[root@localhost ~]# lvcreate -L 2000 VG_DATA -n lv_finance
Logical volume "lv_finance" created.
```

Afficher la liste sommaire des LV :

```
[root@localhost ~]# lvscan
ACTIVE          '/dev/centos/swap' [2.00 GiB] inherit
ACTIVE          '/dev/centos/root' [<17.00 GiB] inherit
ACTIVE          '/dev/VG_DATA/lv_info' [1.95 GiB] inherit
ACTIVE          '/dev/VG_DATA/lv_finance' [1.95 GiB] inherit
```


Afficher la liste détaillée des LV :

```
[root@localhost ~]# lvdisplay

--- Logical volume ---
LV Path                /dev/centos/swap
LV Name                swap
VG Name                centos
LV UUID                Hpdrob-zHgC-jGOs-EUaS-uNME-xb4T-xYt0hm
LV Write Access        read/write
LV Creation host, time localhost, 2018-10-25 06:40:22 -0400
LV Status              available
# open                 2
LV Size                2.00 GiB
Current LE             512
Segments              1
Allocation             inherit
Read ahead sectors     auto
- currently set to    8192
Block device          253:1

--- Logical volume ---
LV Path                /dev/centos/root
LV Name                root
VG Name                centos
LV UUID                KhlqKc-b1KR-FNMT-gaRc-9TQW-tyWD-SFFgIl
LV Write Access        read/write
LV Creation host, time localhost, 2018-10-25 06:40:23 -0400
LV Status              available
# open                 1
LV Size                <17.00 GiB
Current LE             4351
Segments              1
Allocation             inherit
Read ahead sectors     auto
- currently set to    8192
Block device          253:0

--- Logical volume ---
LV Path                /dev/VG_DATA/lv_info
LV Name                lv_info
VG Name                VG_DATA
LV UUID                aTtChO-zK28-FpMR-URX1-k9zI-GAvL-2X410w
LV Write Access        read/write
LV Creation host, time localhost.localdomain, 2019-03-11 17:51:05 -0400
LV Status              available
# open                 0
LV Size                1.95 GiB
Current LE             500
Segments              1
Allocation             inherit
Read ahead sectors     auto
- currently set to    8192
Block device          253:2

--- Logical volume ---
LV Path                /dev/VG_DATA/lv_finance
LV Name                lv_finance
VG Name                VG_DATA
LV UUID                aTtld3-tbzb-K62C-TOFk-df3Q-DFu8-peAxG1
```

LV Write Access	read/write
LV Creation host, time	localhost.localdomain, 2019-03-11 17:52:34 -0400
LV Status	available
# open	0
LV Size	1.95 GiB
Current LE	500
Segments	1
Allocation	inherit
Read ahead sectors	auto
- currently set to	8192
Block device	253:3

Le terme '**LE**' dans l'affichage des informations sur le volume logique signifie '**Logical Extents**' ou domaines logiques. Ceux-ci représentent les domaines physiques du groupe de volumes associés à ce volume logique.

Il existe toute une panoplie de commandes pour manipuler un volume logique :

La commande **lvchange** permet de changer les attributs d'un ou de plusieurs volumes logiques.

La commande **lvextend** permet d'augmenter (+) la taille d'un volume logique.

La commande **lvreduce** permet de diminuer (-) la taille d'un volume logique.

La commande **lvrename** permet de renommer un volume logique.

La commande **lvremove** permet de supprimer un ou plusieurs volumes logiques.

8 Créer un système de fichiers (FileSystem)

Il reste à créer un système de fichiers dans chaque volume logique :

```
[root@localhost ~]# mkfs -t xfs /dev/VG_DATA/lv_finance
meta-data=/dev/VG_DATA/lv_finance isize=512  agcount=4, agsize=128000 blks
          =                               sectsz=512   attr=2, projid32bit=1
          =                               crc=1        finobt=0, sparse=0
data      =                               bsize=4096   blocks=512000, 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
```

```
[root@localhost ~]# mkfs -t xfs /dev/VG_DATA/lv_info
meta-data=/dev/VG_DATA/lv_info isize=512  agcount=4, agsize=128000 blks
          =                               sectsz=512   attr=2, projid32bit=1
          =                               crc=1        finobt=0, sparse=0
data      =                               bsize=4096   blocks=512000, 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
```

9 Monter un système de fichiers (FileSystem)

Pour pouvoir utiliser les deux systèmes de fichiers, il faut les monter.

On va d'abord créer les points de montage :

```
[root@localhost ~]# mkdir /finance
```

```
[root@localhost ~]# mkdir /info
```

Ensuite monter chaque système de fichiers dans son point de montage :

```
[root@localhost ~]# mount /dev/VG_DATA/lv_info /info
```

```
[root@localhost ~]# mount /dev/VG_DATA/lv_finance /finance
```

Vérifier les montages :

```
[root@localhost ~]# mount|grep VG
/dev/mapper/VG_DATA-lv_info on /info type xfs
(rw,relatime,attr2,inode64,noquota)
/dev/mapper/VG_DATA-lv_finance on /finance type xfs
(rw,relatime,attr2,inode64,noquota)
```

Pour que les montages se fassent automatiquement au démarrage du serveur, il faut ajouter les deux lignes suivantes dans le fichier **/etc/fstab** :

/dev/VG_DATA/lv_info	/info	xfs	defaults	0	0
/dev/VG_DATA/lv_finance	/finance	xfs	defaults	0	0

10 Redimensionner un VG

- 1) Ajouter un nouveau disque de 2GB et démarrer le serveur.

▼ Devices	
Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
Hard Disk 2 (SCSI)	10 GB
Hard Disk 3 (SCSI)	2 GB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
Display	Auto detect

- 2) Lancer la commande fdisk

```
root@localhost:~  
[root@localhost ~]# fdisk /dev/sdc  
Welcome to fdisk (util-linux 2.23.2).  
  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
Device does not contain a recognized partition table  
Building a new DOS disklabel with disk identifier 0x6c846482.  
  
Command (m for help):
```

- 3) Créer une nouvelle partition

```
root@localhost:~  
Command (m for help): n  
Partition type:  
  p   primary (0 primary, 0 extended, 4 free)  
  e   extended  
Select (default p):  
Using default response p  
Partition number (1-4, default 1):  
First sector (2048-4194303, default 2048):  
Using default value 2048  
Last sector, +sectors or +size{K,M,G} (2048-4194303, default 4194303):  
Using default value 4194303  
Partition 1 of type Linux and of size 2 GiB is set  
  
Command (m for help):
```

- 4) Le type de la nouvelle partition est **8e** (LVM)

```
root@localhost:~  
Command (m for help): t  
Selected partition 1  
Hex code (type L to list all codes): 8e  
Changed type of partition 'Linux' to 'Linux LVM'  
Command (m for help):
```

- 5) Afficher la table des partitions

```
root@localhost:~  
Command (m for help): p  
  
Disk /dev/sdc: 2147 MB, 2147483648 bytes, 4194304 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk label type: dos  
Disk identifier: 0x6c846482  


| Device    | Boot | Start | End     | Blocks  | Id | System    |
|-----------|------|-------|---------|---------|----|-----------|
| /dev/sdc1 |      | 2048  | 4194303 | 2096128 | 8e | Linux LVM |

  
Command (m for help):
```

- 6) Sauvegarder

```
root@localhost:~  
Command (m for help): w  
The partition table has been altered!  
  
Calling ioctl() to re-read partition table.  
Syncing disks.  
[root@localhost ~]#
```

7) Créer un volume physique

```
[root@localhost ~]# pvcreate /dev/sdc1

Physical volume "/dev/sdc1" successfully created.
```

8) Afficher les propriétés du nouveau volume physique

```
[root@localhost ~]# pvdisplay /dev/sdc1

"/dev/sdc1" is a new physical volume of "<2.00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdc1
VG Name
PV Size                <2.00 GiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               tspAF4-kejY-E134-ZQZK-ND4n-MmnA-WJhFxL
```

9) Ajouter le volume physique au VG VG_DATA

```
[root@localhost ~]# vgextend VG_DATA /dev/sdc1

Volume group "VG_DATA" successfully extended
```

10) Afficher les propriétés du VG VG_DATA

```
[root@localhost ~]# vgdisplay VG_DATA

--- Volume group ---
VG Name                VG_DATA
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   4
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                2
Open LV                2
Max PV                 0
Cur PV                3
Act PV                 3
VG Size                <11.99 GiB
PE Size                4.00 MiB
Total PE               3069
Alloc PE / Size        1000 / <3.91 GiB
Free PE / Size         2069 / 8.08 GiB
VG UUID                IuwOtt-Qnto-unpe-iZLY-QX0A-z2qT-7UCdNj
```

11 Redimensionner un LV

Pour redimensionner un volume logique, on utilise la commande *lvextend*.

On va redimensionner la taille du volume logique à *lv_info* à 5Gb.

```
[root@localhost ~]# lvextend -L5G /dev/VG_DATA/lv_info

Size of logical volume VG_DATA/lv_info changed from 1.95
GiB (500 extents) to 5.00 GiB (1280 extents).
Logical volume VG_DATA/lv_info successfully resized.
```

Si on veut redimensionner en utilisant 100% de l'espace libre du VG :

```
[root@localhost ~]# lvextend -l +100%FREE /dev/VG_DATA/lv_info
```

Pour vérifier la nouvelle taille :

```
[root@localhost ~]# lvdisplay /dev/VG_DATA/lv_info

--- Logical volume ---
LV Path                /dev/VG_DATA/lv_info
LV Name                lv_info
VG Name                VG_DATA
LV UUID                aTtChO-zK28-FpMR-URX1-k9zI-GAvL-
2X410w
LV Write Access        read/write
LV Creation host, time localhost.localdomain, 2019-03-11
17:51:05 -0400
LV Status              available
# open                 1
LV Size                5.00 GiB
Current LE             1280
Segments               3
Allocation             inherit
Read ahead sectors     auto
- currently set to     8192
Block device           253:2
```


12 Redimensionner un système de fichiers (FileSystem)

Afficher la taille de l'espace avant de redimensionner :

```
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/centos-root	17G	1.2G	16G	7%	/
devtmpfs	900M	0	900M	0%	/dev
tmpfs	911M	0	911M	0%	/dev/shm
tmpfs	911M	9.5M	902M	2%	/run
tmpfs	911M	0	911M	0%	/sys/fs/cgroup
/dev/sda1	1014M	184M	831M	19%	/boot
tmpfs	183M	0	183M	0%	/run/user/0
/dev/mapper/VG_DATA-lv_finance	2.0G	33M	2.0G	2%	/finance
/dev/mapper/VG_DATA-lv_info	2.0G	33M	2.0G	2%	/info

On va redimensionner la taille du système de fichiers afin qu'il occupe la totalité de l'espace du volume logique à *lv_info* (5Gb).

```
[root@localhost ~]# xfs_growfs /info
```

meta-data=/dev/mapper/VG_DATA-lv_info isize=512 agcount=4,
agsize=128000 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=1 finobt=0 spinodes=0
data = bsize=4096 blocks=512000, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal bsize=4096 blocks=2560, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
data blocks changed from 512000 to 1310720

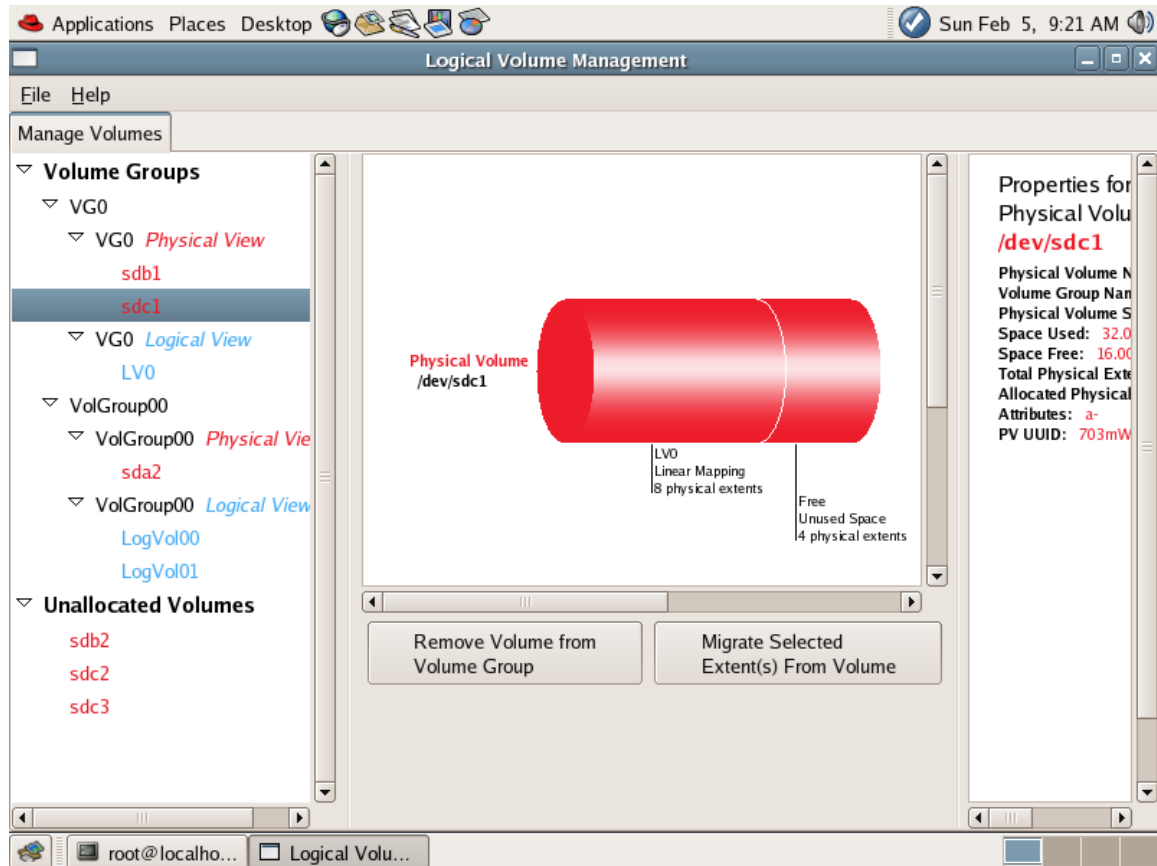
Vérifier la nouvelle taille :

```
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/centos-root	17G	1.2G	16G	7%	/
devtmpfs	900M	0	900M	0%	/dev
tmpfs	911M	0	911M	0%	/dev/shm
tmpfs	911M	9.5M	902M	2%	/run
tmpfs	911M	0	911M	0%	/sys/fs/cgroup
/dev/sda1	1014M	184M	831M	19%	/boot
tmpfs	183M	0	183M	0%	/run/user/0
/dev/mapper/VG_DATA-lv_finance	2.0G	33M	2.0G	2%	/finance
/dev/mapper/VG_DATA-lv_info	5.0G	33M	5.0G	1%	/info

13 Outil graphique

CentOS est accompagné de l'outil graphique **system-config-lvm** qui permet de retrouver toutes les informations précédentes de manière plus conviviale :



Cet outil est accessible via le menu GNOME **Environnement de bureau – Paramètres de système – Logical Volume Management**