

Version : **2022.01**

Dernière mise-à-jour : 2022/11/01 07:27

Topic 204: Advanced Storage Device Administration

- **Topic 204: Advanced Storage Device Administration**

- Logical Volume Manager (LVM)
 - LAB #1 - Volumes Logiques Linéaires
 - Physical Volume (PV)
 - Volume Group (VG) et Physical Extent (PE)
 - Logical Volumes (LV)
 - LAB #2 - Étendre un Volume Logique à Chaud
 - LAB #3 - Snapshots
 - LAB #4 - Suppression des Volumes
 - LAB #5 - Volumes Logiques en Miroir
 - LAB #6 - Modifier les Attributs LVM
 - LAB #7 - Volumes Logiques en Bandes
 - LAB #8 - Gérer les Métadonnées
 - RAID Logiciel
 - Concepts RAID
 - Disques en miroir
 - Bandes de données
 - Types de RAID
 - RAID 0 - Concaténation
 - RAID 0 - Striping
 - RAID 1 - Miroir
 - RAID 1+0 - Striping en Miroir
 - RAID 2 - Miroir avec Contrôle d'Erreurs
 - RAID 3 et 4 - Striping avec Parité
 - RAID 5 - Striping avec Parité Distribuée
 - Au delà de RAID 5
 - LAB #9 - Mise en Place du RAID 5 Logiciel
-

- 9.1 - Préparer le disque
 - 9.2 - Créer une Unité RAID
 - 9.3 - Remplacer une Unité Défaillante
- LAB #10 - autofs

Logical Volume Manager (LVM)

LAB #1 - Volumes Logiques Linéaires

Afin de mettre en place le LVM, vous avez besoin du paquet **lvm2** et du paquet **device-mapper**.

Sous Debian 11, installez le paquet lvm2 :

```
root@debian11:~# apt-get -y install lvm2
```

Nous allons travailler avec les partitions suivantes :

/dev/sdc6	1644544	2054143	409600	200M	8e	Linux	LVM
/dev/sdc7	2056192	2670591	614400	300M	8e	Linux	LVM
/dev/sdc9	3698688	4517887	819200	400M	8e	Linux	LVM

Pour initialiser le LVM saisissez la commande suivante :

```
root@debian11:~# vgscan
root@debian11:~#
```

Les options de la commande **vgscan** sont :

```
root@debian11:~# vgscan --help
vgscan - Search for all volume groups
```

```
vgscan
[ --ignorelockingfailure ]
[ --mknodes ]
[ --notifydbus ]
[ --reportformat basic|json ]
[ COMMON_OPTIONS ]
```

Common options for lvm:

```
[ -d|--debug ]
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Physical Volume (PV)

Pour créer le **PV** il convient d'utiliser la commande **pvcreate** :

```
root@debian11:~# pvcreate /dev/sdc6 /dev/sdc7 /dev/sdc9
Physical volume "/dev/sdc6" successfully created.
Physical volume "/dev/sdc7" successfully created.
```

Physical volume "/dev/sdc9" successfully created.

Les options de la commande **pvcreate** sont :

```
root@debian11:~# pvcreate --help
pvcreate - Initialize physical volume(s) for use by LVM
```

```
pvcreate PV ...
  [ -f|--force ]
  [ -M|--metadatatype lvm2 ]
  [ -u|--uuid String ]
  [ -Z|--zero y|n ]
  [ --dataalignment Size[k|UNIT] ]
  [ --dataalignmentoffset Size[k|UNIT] ]
  [ --bootloaderareaseize Size[m|UNIT] ]
  [ --labelsector Number ]
  [ --pvmetadatacopies 0|1|2 ]
  [ --metadatasize Size[m|UNIT] ]
  [ --metadataignore y|n ]
  [ --norestorefile ]
  [ --setphysicalvolumesize Size[m|UNIT] ]
  [ --reportformat basic|json ]
  [ --restorefile String ]
  [ COMMON_OPTIONS ]
```

Common options for lvm:

```
  [ -d|--debug ]
  [ -h|--help ]
  [ -q|--quiet ]
  [ -v|--verbose ]
  [ -y|--yes ]
  [ -t|--test ]
  [ --commandprofile String ]
  [ --config String ]
```

```
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Pour visualiser le PV il convient d'utiliser la commande **pvdisk** :

```
root@debian11:~# pvdisk /dev/sdc6 /dev/sdc7 /dev/sdc9
"/dev/sdc6" is a new physical volume of "200.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sdc6
VG Name
PV Size          200.00 MiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          2NmW8z-bVaE-Y1AJ-NjgJ-iS4U-x3i7-Anjofh
"/dev/sdc7" is a new physical volume of "300.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sdc7
VG Name
PV Size          300.00 MiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          VARemm-RCo9-qP7f-0gGP-I6Ym-b494-RXfNjC
```

```
"/dev/sdc9" is a new physical volume of "400.00 MiB"
--- NEW Physical volume ---
PV Name                /dev/sdc9
VG Name
PV Size                400.00 MiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                Q3Vqjd-rGf1-20vs-S0An-0gY2-eGKr-qmyCY2
```

Les options de la commande **pvdisplay** sont :

```
root@debian11:~# pvdisplay --help
pvdisplay - Display various attributes of physical volume(s)

pvdisplay
  [ -a|--all ]
  [ -c|--colon ]
  [ -C|--columns ]
  [ -m|--maps ]
  [ -o|--options String ]
  [ -S|--select String ]
  [ -s|--short ]
  [ -O|--sort String ]
  [ --aligned ]
  [ --binary ]
  [ --configreport log|vg|lv|pv|pvseg|seg ]
  [ --foreign ]
  [ --ignorelockingfailure ]
  [ --logonly ]
  [ --noheadings ]
  [ --nosuffix ]
```

```
[ --readonly ]
[ --reportformat basic|json ]
[ --separator String ]
[ --shared ]
[ --unbuffered ]
[ --units r|R|h|H|b|B|s|S|k|K|m|M|g|G|t|T|p|P|e|E ]
[ COMMON_OPTIONS ]
[ PV|Tag ... ]
```

Common options for lvm:

```
[ -d|--debug ]
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Volume Group (VG) et Physical Extent (PE)

Pour créer un Volume Group dénommé **vg0**, il convient d'utiliser la commande **vgcreate** :

```
root@debian11:~# vgcreate -s 8M vg0 /dev/sdc6 /dev/sdc7 /dev/sdc9
```

Volume group "vg0" successfully created

Les options de la commande **vgcreate** sont :

```
root@debian11:~# vgcreate --help
vgcreate - Create a volume group

vgcreate VG_new PV ...
  [ -A|--autobackup y|n ]
  [ -c|--clustered y|n ]
  [ -l|--maxlogicalvolumes Number ]
  [ -p|--maxphysicalvolumes Number ]
  [ -M|--metadatatype lvm2 ]
  [ -s|--physicalextentsize Size[m|UNIT] ]
  [ -f|--force ]
  [ -Z|--zero y|n ]
  [   --addtag Tag ]
  [   --alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit ]
  [   --metadataprofile String ]
  [   --labelsector Number ]
  [   --metadatasize Size[m|UNIT] ]
  [   --pvmetadatacopies 0|1|2 ]
  [   --vgmetadatacopies all|unmanaged|Number ]
  [   --reportformat basic|json ]
  [   --dataalignment Size[k|UNIT] ]
  [   --dataalignmentoffset Size[k|UNIT] ]
  [   --shared ]
  [   --systemid String ]
  [   --locktype sanlock|dlm|none ]
  [ COMMON_OPTIONS ]

Common options for lvm:
  [ -d|--debug ]
  [ -h|--help ]
```

```
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Pour afficher les informations concernant **vg0**, il convient d'utiliser la commande **vgdisplay** :

```
root@debian11:~# vgdisplay vg0
--- Volume group ---
VG Name                vg0
System ID
Format                 lvm2
Metadata Areas         3
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                 3
Act PV                  3
VG Size                 880.00 MiB
PE Size                 8.00 MiB
```

```
Total PE          110
Alloc PE / Size   0 / 0
Free PE / Size    110 / 880.00 MiB
VG UUID           d7zxKd-eVpl-Zrp1-sN0e-HbyL-zjIV-2xpDua
```

Les options de la commande **vgdisplay** sont :

```
root@debian11:~# vgdisplay --help
vgdisplay - Display volume group information

vgdisplay
  [ -A|--activevolumegroups ]
  [ -c|--colon ]
  [ -C|--columns ]
  [ -o|--options String ]
  [ -S|--select String ]
  [ -s|--short ]
  [ -O|--sort String ]
  [   --aligned ]
  [   --binary ]
  [   --configreport log|vg|lv|pv|pvseg|seg ]
  [   --foreign ]
  [   --ignorelockingfailure ]
  [   --logonly ]
  [   --noheadings ]
  [   --nosuffix ]
  [   --readonly ]
  [   --reportformat basic|json ]
  [   --shared ]
  [   --separator String ]
  [   --unbuffered ]
  [   --units r|R|h|H|b|B|s|S|k|K|m|M|g|G|t|T|p|P|e|E ]
  [ COMMON_OPTIONS ]
  [ VG|Tag ... ]
```

```
Common options for lvm:
[ -d|--debug ]
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Logical Volumes (LV)

Pour créer un **Logical Volume** dénommé **lv0** dans le **Volume Group** **vg0**, il convient d'utiliser la commande **lvcreate** :

```
root@debian11:~# lvcreate -L 350 -n lv0 vg0
Rounding up size to full physical extent 352.00 MiB
Logical volume "lv0" created.
```



Notez que la taille du LV est un multiple du PE.

Les options de la commande **lvcreate** sont :

```
root@debian11:~# lvcreate --help
lvcreate - Create a logical volume

Create a linear LV.
lvcreate -L|--size Size[m|UNIT] VG
    [ -l|--extents Number[PERCENT] ]
    [ --type linear ]
    [ COMMON_OPTIONS ]
    [ PV ... ]

Create a striped LV (infers --type striped).
lvcreate -i|--stripes Number -L|--size Size[m|UNIT] VG
    [ -l|--extents Number[PERCENT] ]
    [ -I|--stripesize Size[k|UNIT] ]
    [ COMMON_OPTIONS ]
    [ PV ... ]

Create a raid1 or mirror LV (infers --type raid1|mirror).
lvcreate -m|--mirrors Number -L|--size Size[m|UNIT] VG
    [ -l|--extents Number[PERCENT] ]
    [ -R|--regionsize Size[m|UNIT] ]
    [ --mirrorlog core|disk ]
    [ --minrecoveryrate Size[k|UNIT] ]
    [ --maxrecoveryrate Size[k|UNIT] ]
    [ COMMON_OPTIONS ]
    [ PV ... ]

Create a raid LV (a specific raid level must be used, e.g. raid1).
lvcreate --type raid -L|--size Size[m|UNIT] VG
    [ -l|--extents Number[PERCENT] ]
    [ -m|--mirrors Number ]
    [ -i|--stripes Number ]
    [ -I|--stripesize Size[k|UNIT] ]
    [ -R|--regionsize Size[m|UNIT] ]
```

```
[ --minrecoveryrate Size[k|UNIT] ]  
[ --maxrecoveryrate Size[k|UNIT] ]  
[ --raidintegrity y|n ]  
[ --raidintegritymode String ]  
[ --raidintegrityblocksize Number ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Create a raid10 LV.

```
lvcreate -m|--mirrors Number -i|--stripes Number -L|--size Size[m|UNIT] VG  
[ -l|--extents Number[PERCENT] ]  
[ -I|--stripesize Size[k|UNIT] ]  
[ -R|--regionsize Size[m|UNIT] ]  
[ --minrecoveryrate Size[k|UNIT] ]  
[ --maxrecoveryrate Size[k|UNIT] ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Create a COW snapshot LV of an origin LV.

```
lvcreate -s|--snapshot -L|--size Size[m|UNIT] LV  
[ -l|--extents Number[PERCENT] ]  
[ -i|--stripes Number ]  
[ -I|--stripesize Size[k|UNIT] ]  
[ -c|--chunksize Size[k|UNIT] ]  
[ --type snapshot ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Create a thin pool.

```
lvcreate --type thin-pool -L|--size Size[m|UNIT] VG  
[ -l|--extents Number[PERCENT] ]  
[ -c|--chunksize Size[k|UNIT] ]  
[ -i|--stripes Number ]  
[ -I|--stripesize Size[k|UNIT] ]
```

```
[ --thinpool LV_new ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a cache pool.

```
lvcreate --type cache-pool -L|--size Size[m|UNIT] VG
[ -l|--extents Number[PERCENT] ]
[ -H|--cache ]
[ -c|--chunksize Size[k|UNIT] ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
[ --cachesettings String ]
[ --cachemetadadataformat auto|1|2 ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a thin LV in a thin pool (infers --type thin).

```
lvcreate -V|--virtualsize Size[m|UNIT] --thinpool LV_thinpool VG
[ -T|--thin ]
[ --type thin ]
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ COMMON_OPTIONS ]
```

Create a thin LV that is a snapshot of an existing thin LV (infers --type thin).

```
lvcreate -s|--snapshot LV_thin
[ --type thin ]
```

```
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ COMMON_OPTIONS ]
```

Create a thin LV that is a snapshot of an external origin LV.

```
lvcreate --type thin --thinpool LV_thinpool LV
[ -T|--thin ]
[ -c|--chunksize Size[k|UNIT] ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ COMMON_OPTIONS ]
```

Create a LV that returns VDO when used.

```
lvcreate --type vdo -L|--size Size[m|UNIT] VG
[ -l|--extents Number[PERCENT] ]
[ -V|--virtualseize Size[m|UNIT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --vdo ]
[ --vdopool LV_new ]
[ --compression y|n ]
[ --deduplication y|n ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a thin LV, first creating a thin pool for it,
where the new thin pool is named by the --thinpool arg.

```
lvcreate --type thin -V|--virtualseize Size[m|UNIT] -L|--size Size[m|UNIT] --thinpool LV_new
[ -l|--extents Number[PERCENT] ]
[ -T|--thin ]
[ -c|--chunksize Size[k|UNIT] ]
[ -i|--stripes Number ]
```

```
[ -I|--stripesize Size[k|UNIT] ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a new LV, then attach the specified cachepool which converts the new LV to type cache.

```
lvcreate --type cache -L|--size Size[m|UNIT] --cachepool LV_cachepool VG
[ -l|--extents Number[PERCENT] ]
[ -H|--cache ]
[ -c|--chunksize Size[k|UNIT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
[ --cachesettings String ]
[ --cachemetadadataformat auto|1|2 ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a new LV, then attach the specified cachevol which converts the new LV to type cache.

```
lvcreate --type cache -L|--size Size[m|UNIT] --cachevol LV VG
[ -l|--extents Number[PERCENT] ]
[ -c|--chunksize Size[k|UNIT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
```

```
[ --cachesettings String ]
[ --cachemetadadataformat auto|1|2 ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a new LV, then attach a cachevol created from the specified cache device, which converts the new LV to type cache.

```
lvcreate --type cache -L|--size Size[m|UNIT] --cachedevice PV VG
[ -l|--extents Number[PERCENT] ]
[ -c|--chunksize Size[k|UNIT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
[ --cachesettings String ]
[ --cachemetadadataformat auto|1|2 ]
[ --cachesize Size[m|UNIT] ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a new LV, then attach the specified cachevol which converts the new LV to type writecache.

```
lvcreate --type writecache -L|--size Size[m|UNIT] --cachevol LV VG
[ -l|--extents Number[PERCENT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --cachesettings String ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a new LV, then attach a cachevol created from the specified cache device, which converts the new LV to type writecache.

```
lvcreate --type writecache -L|--size Size[m|UNIT] --cachedevice PV VG
[ -l|--extents Number[PERCENT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --cachesize Size[m|UNIT] ]
[ --cachesettings String ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Common options for command:

```
[ -a|--activate y|n|ay ]
[ -A|--autobackup y|n ]
[ -C|--contiguous y|n ]
[ -M|--persistent y|n ]
[ -j|--major Number ]
[ -k|--setactivationskip y|n ]
[ -K|--ignoreactivationskip ]
[ -n|--name String ]
[ -p|--permission rw|r ]
[ -r|--readahead auto|none|Number ]
[ -W|--wipesignatures y|n ]
[ -Z|--zero y|n ]
[ --addtag Tag ]
[ --alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit ]
[ --ignoremonitoring ]
[ --metadataprofile String ]
[ --minor Number ]
[ --monitor y|n ]
[ --nosync ]
[ --noudevsync ]
[ --reportformat basic|json ]
```

Common options for lvm:

```
[ -d|--debug ]
```

```
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Créez maintenant un répertoire dans `/mnt` pour monter `lv0` :

```
root@debian11:~# mkdir /mnt/lvm
```

Créez un système de fichiers en **ext3** sur `/dev/vg0/lv0` :

```
root@debian11:~# mke2fs -j /dev/vg0/lv0
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 360448 1k blocks and 90112 inodes
Filesystem UUID: f6c32097-8d4b-4e65-8880-4b733350193a
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

Montez votre lv0 :

```
root@debian11:~# mount -t ext3 /dev/vg0/lv0 /mnt/lvm
```

Vous allez maintenant copier le contenu du répertoire /home vers /mnt/lvm.

Saisissez donc la commande suivante :

```
root@debian11:~# cp -a /home /mnt/lvm
```

Constatez ensuite le contenu de /mnt/lvm :

```
root@debian11:~# ls -l /mnt/lvm
total 13
drwxr-xr-x 3 root root 1024 Apr 25 07:01 home
drwx----- 2 root root 12288 Apr 26 15:44 lost+found
```

Une particularité du volume logique est la capacité de d'être agrandi ou réduit sans pertes de données. Commencez par constater la taille totale du volume :

```
root@debian11:~# df -h /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 331M  1.2M  313M   1% /mnt/lvm
```

Dans la cas de notre exemple, la taille est de **331 Mo** avec **1,2 Mo** occupé.

LAB #2 - Etendre un Volume Logique à Chaud

Pour agrandir un volume logique, le paquet **lvm2** contient les commandes **lvextend** et **resize2fs** :

```
root@debian11:~# lvextend -L +100M /dev/vg0/lv0
Rounding size to boundary between physical extents: 104.00 MiB.
Size of logical volume vg0/lv0 changed from 352.00 MiB (44 extents) to 456.00 MiB (57 extents).
Logical volume vg0/lv0 successfully resized.
```



Important - Notez que l'agrandissement du volume est un multiple du PE.

Les options de la commande **lvextend** sont :

```
root@debian11:~# lvextend --help
lvextend - Add space to a logical volume

Extend an LV by a specified size.
lvextend -L|--size [+]Size[m|UNIT] LV
  [ -l|--extents [+]Number[PERCENT] ]
  [ -r|--resizefs ]
  [ -i|--stripes Number ]
  [ -I|--stripesize Size[k|UNIT] ]
  [ --poolmetadatasize [+]Size[m|UNIT] ]
  [ COMMON_OPTIONS ]
  [ PV ... ]

Extend an LV by specified PV extents.
lvextend LV PV ...
  [ -r|--resizefs ]
  [ -i|--stripes Number ]
  [ -I|--stripesize Size[k|UNIT] ]
  [ COMMON_OPTIONS ]

Extend a pool metadata SubLV by a specified size.
lvextend --poolmetadatasize [+]Size[m|UNIT] LV_thinpool
```

```
[ -i|--stripes Number ]  
[ -I|--stripesize Size[k|UNIT] ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Extend an LV according to a predefined policy.

```
lvextend --usepolicies LV_snapshot_thinpool  
[ -r|--resizefs ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Common options for command:

```
[ -A|--autobackup y|n ]  
[ -f|--force ]  
[ -m|--mirrors Number ]  
[ -n|--nofsck ]  
[ --alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit ]  
[ --nosync ]  
[ --noudevsync ]  
[ --reportformat basic|json ]  
[ --type linear|striped|snapshot|mirror|raid|thin|cache|vdo|thin-pool|cache-pool|vdo-pool ]
```

Common options for lvm:

```
[ -d|--debug ]  
[ -h|--help ]  
[ -q|--quiet ]  
[ -v|--verbose ]  
[ -y|--yes ]  
[ -t|--test ]  
[ --commandprofile String ]  
[ --config String ]  
[ --driverloaded y|n ]  
[ --nolocking ]  
[ --lockopt String ]
```

```
[ --longhelp ]  
[ --profile String ]  
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Le volume ayant été agrandi, il est nécessaire maintenant d'agrandir le filesystem qui s'y trouve :

```
root@debian11:~# df -h /mnt/lvm  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/vg0-lv0 331M  1.2M  313M   1% /mnt/lvm  
root@debian11:~# resize2fs /dev/vg0/lv0  
resize2fs 1.46.2 (28-Feb-2021)  
Filesystem at /dev/vg0/lv0 is mounted on /mnt/lvm; on-line resizing required  
old_desc_blocks = 2, new_desc_blocks = 2  
The filesystem on /dev/vg0/lv0 is now 466944 (1k) blocks long.
```

Constatez maintenant la modification de la taille du volume :

```
root@debian11:~# df -h /mnt/lvm  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/vg0-lv0 434M  3.5M  409M   1% /mnt/lvm
```

Vous noterez que la taille a augmentée mais que les données sont toujours présentes.

LAB #3 - Snapshots

Un snapshot est un instantané d'un système de fichiers. Dans cet exemple, vous allez créer un snapshot de votre lv0 :

Avant de commencer, créez un fichier de 10Mo dans le volume :

```
root@debian11:~# dd if=/dev/zero of=/mnt/lvm/10M bs=1048576 count=10
```

```
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.023862 s, 439 MB/s
```

Créez maintenant un snapshot :

```
root@debian11:~# lvcreate -s -L 20M -n testsnap /dev/vg0/lv0
Rounding up size to full physical extent 24.00 MiB
Logical volume "testsnap" created.
```

Pour avoir une confirmation de la création du snapshot, utilisez la commande **lvs** :

```
root@debian11:~# lvs
LV      VG  Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv0     vg0 owi-aos--- 456.00m
testsnap vg0 swi-a-s--- 24.00m      lv0    0.05
```



Important - Notez que le snapshot est créé dans le même VG que le LV d'origine.

Les options de la commande **lvs** sont :

```
root@debian11:~# lvs --help
lvs - Display information about logical volumes

lvs
  [ -H|--history ]
  [ -a|--all ]
  [ -o|--options String ]
  [ -S|--select String ]
  [ -O|--sort String ]
  [ --segments ]
```

```
[ --aligned ]
[ --binary ]
[ --configreport log|vg|lv|pv|pvseg|seg ]
[ --foreign ]
[ --ignorelockingfailure ]
[ --logonly ]
[ --nameprefixes ]
[ --noheadings ]
[ --nosuffix ]
[ --readonly ]
[ --reportformat basic|json ]
[ --rows ]
[ --separator String ]
[ --shared ]
[ --unbuffered ]
[ --units r|R|h|H|b|B|s|S|k|K|m|M|g|G|t|T|p|P|e|E ]
[ --unquoted ]
[ COMMON_OPTIONS ]
[ VG|LV|Tag ... ]
```

Common options for lvm:

```
[ -d|--debug ]
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
```

```
[ --version ]
```

Use --longhelp to show all options and advanced commands.

Créez maintenant un répertoire pour monter le snapshot :

```
root@debian11:~# mkdir /mnt/testsnap
```

Montez le snapshot :

```
root@debian11:~# mount /dev/vg0/testsnap /mnt/testsnap
```

Comparez le volume d'origine et le snapshot :

```
root@debian11:~# ls -l /mnt/lvm
total 10294
-rw-r--r-- 1 root root 10485760 Apr 26 15:50 10M
drwxr-xr-x 3 root root    1024 Apr 25 07:01 home
drwx----- 2 root root    12288 Apr 26 15:44 lost+found
root@debian11:~# ls -l /mnt/testsnap
total 10294
-rw-r--r-- 1 root root 10485760 Apr 26 15:50 10M
drwxr-xr-x 3 root root    1024 Apr 25 07:01 home
drwx----- 2 root root    12288 Apr 26 15:44 lost+found
```

Supprimez maintenant le fichier **10M** de votre volume d'origine :

```
root@debian8:~# rm /mnt/lvm/10M
```

Constatez le résultat de cette suppression :

```
root@debian11:~# df -Ph /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
```

```
/dev/mapper/vg0-lv0 434M 3.5M 409M 1% /mnt/lvm
root@debian11:~# df -Ph /mnt/testsnap/
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/vg0-testsnap 435M  15M 399M  4% /mnt/testsnap
```



A Faire - Restaurez le fichier 10M à partir du snapshot.

LAB #4 - Suppression des Volumes

La suppression d'un volume logique se fait grace à la commande **lvremove** :

```
root@debian11:~# umount /mnt/testsnap/

root@debian11:~# lvremove /dev/vg0/testsnap
Do you really want to remove active logical volume vg0/testsnap? [y/n]: y
Logical volume "testsnap" successfully removed

root@debian11:~# umount /mnt/lvm

root@debian11:~# lvremove /dev/vg0/lv0
Do you really want to remove active logical volume vg0/lv0? [y/n]: y
Logical volume "lv0" successfully removed
```



Important - Notez que cette opération nécessite à ce que le volume logique soit démonté.

Les options de la commande **lvremove** sont :

```
root@debian11:~# lvremove --help
lvremove - Remove logical volume(s) from the system
```

```
lvremove VG|LV|Tag|Select ...
  [-A|--autobackup y|n ]
  [-f|--force ]
  [-S|--select String ]
  [ --nohistory ]
  [ --noudevsync ]
  [ --reportformat basic|json ]
  [ COMMON_OPTIONS ]
```

Common options for lvm:

```
[ -d|--debug ]
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use --longhelp to show all options and advanced commands.

Le Volume Group peut aussi être supprimé :

```
root@debian11:~# vgremove vg0
```

Volume group "vg0" successfully removed

Les options de la commande **vgremove** sont :

```
root@debian11:~# vgremove --help
vgremove - Remove volume group(s)

vgremove VG|Tag|Select ...
  [ -f|--force ]
  [ -S|--select String ]
  [ --noudevsync ]
  [ --reportformat basic|json ]
  [ COMMON_OPTIONS ]

Common options for lvm:
  [ -d|--debug ]
  [ -h|--help ]
  [ -q|--quiet ]
  [ -v|--verbose ]
  [ -y|--yes ]
  [ -t|--test ]
  [ --commandprofile String ]
  [ --config String ]
  [ --driverloaded y|n ]
  [ --nolocking ]
  [ --lockopt String ]
  [ --longhelp ]
  [ --profile String ]
  [ --version ]
```

Use `--longhelp` to show all options and advanced commands.

Ainsi que le volume physique :

```
root@debian11:~# pvremove /dev/sdc6 /dev/sdc7 /dev/sdc9
Labels on physical volume "/dev/sdc6" successfully wiped.
Labels on physical volume "/dev/sdc7" successfully wiped.
Labels on physical volume "/dev/sdc9" successfully wiped.
```

Les options de la commande **pvremove** sont :

```
root@debian11:~# pvremove --help
pvremove - Remove LVM label(s) from physical volume(s)

pvremove PV ...
  [ -f|--force ]
  [ --reportformat basic|json ]
  [ COMMON_OPTIONS ]

Common options for lvm:
  [ -d|--debug ]
  [ -h|--help ]
  [ -q|--quiet ]
  [ -v|--verbose ]
  [ -y|--yes ]
  [ -t|--test ]
  [ --commandprofile String ]
  [ --config String ]
  [ --driverloaded y|n ]
  [ --nolocking ]
  [ --lockopt String ]
  [ --longhelp ]
  [ --profile String ]
  [ --version ]

Use --longhelp to show all options and advanced commands.
```

LAB #5 - Volumes Logiques en Miroir

Re-créez maintenant votre Volume Group :

```
root@debian11:~# pvcreate /dev/sdc6 /dev/sdc7 /dev/sdc9
Physical volume "/dev/sdc6" successfully created.
Physical volume "/dev/sdc7" successfully created.
Physical volume "/dev/sdc9" successfully created
```

Créez le VG **vg0** :

```
root@debian11:~# vgcreate -s 8M vg0 /dev/sdc6 /dev/sdc7 /dev/sdc9
Volume group "vg0" successfully created
```

Créez maintenant un Logical Volume en miroir grâce à l'option **-m** de la commande **lvcreate**, suivi du nombre de miroirs :

```
root@debian11:~# lvcreate -m 1 -L 100M -n lv1 vg0
Rounding up size to full physical extent 104.00 MiB
Logical volume "lv1" created.
```

Constatez maintenant la présence du miroir :

```
root@debian11:~# lvdisplay -m /dev/vg0/lv1
--- Logical volume ---
LV Path                /dev/vg0/lv1
LV Name                 lv1
VG Name                 vg0
LV UUID                 2AQE1P-kcp7-5w50-9i3M-Ge6L-0Sa7-HDmKii
LV Write Access         read/write
LV Creation host, time debian11, 2022-04-26 16:21:16 +0200
LV Status                available
# open                  0
LV Size                 104.00 MiB
```

```
Current LE          13
Mirrored volumes   2
Segments           1
Allocation          inherit
Read ahead sectors auto
- currently set to 256
Block device        254:4
--- Segments ---
Logical extents 0 to 12:
  Type              raid1
  Monitoring         monitored
  Raid Data LV 0
    Logical volume   lv1_rimage_0
    Logical extents  0 to 12
  Raid Data LV 1
    Logical volume   lv1_rimage_1
    Logical extents  0 to 12
  Raid Metadata LV 0 lv1_rmeta_0
  Raid Metadata LV 1 lv1_rmeta_1
```

Le miroir s'étend sur plusieurs volumes physiques :

```
root@debian11:~# pvdisplay -m /dev/sdc6 /dev/sdc7 /dev/sdc9
--- Physical volume ---
PV Name           /dev/sdc6
VG Name           vg0
PV Size           200.00 MiB / not usable 8.00 MiB
Allocatable       yes
PE Size           8.00 MiB
Total PE          24
Free PE           10
Allocated PE      14
PV UUID           1J010Q-CM90-tKxI-0sM6-0vbe-3eDG-S10H6d
--- Physical Segments ---
```

```
Physical extent 0 to 0:
  Logical volume      /dev/vg0/lv1_rmeta_0
  Logical extents     0 to 0
Physical extent 1 to 13:
  Logical volume      /dev/vg0/lv1_rimage_0
  Logical extents     0 to 12
Physical extent 14 to 23:
  FREE
--- Physical volume ---
PV Name               /dev/sdc7
VG Name               vg0
PV Size               300.00 MiB / not usable 4.00 MiB
Allocatable           yes
PE Size               8.00 MiB
Total PE              37
Free PE               23
Allocated PE          14
PV UUID               GEk0IP-S7ce-8S1G-K0TX-ocxE-Ud6y-IY3f0Z
--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume      /dev/vg0/lv1_rmeta_1
  Logical extents     0 to 0
Physical extent 1 to 13:
  Logical volume      /dev/vg0/lv1_rimage_1
  Logical extents     0 to 12
Physical extent 14 to 36:
  FREE
--- Physical volume ---
PV Name               /dev/sdc9
VG Name               vg0
PV Size               400.00 MiB / not usable 8.00 MiB
Allocatable           yes
PE Size               8.00 MiB
Total PE              49
```

```
Free PE          49
Allocated PE     0
PV UUID          J7UiEX-m983-j1fp-rU7x-TuCh-MFKh-s105M0
--- Physical Segments ---
Physical extent 0 to 48:
FREE
```

En regardant la sortie de la commande **lsblk**, on observe :

```
root@debian11:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   64G  0 disk
sdb                                  8:16   0   32G  0 disk
├─sdb1                               8:17   0   31G  0 part /
├─sdb2                               8:18   0    1K  0 part
└─sdb5                               8:21   0  975M  0 part [SWAP]
sdc                                  8:32   0    4G  0 disk
├─sdc1                               8:33   0  100M  0 part
├─sdc2                               8:34   0  100M  0 part
├─sdc3                               8:35   0  100M  0 part
├─sdc4                               8:36   0    1K  0 part
├─sdc5                               8:37   0  500M  0 part
├─sdc6                               8:38   0  200M  0 part
│   └─vg0-lv1_rmeta_0                254:0   0    8M  0 lvm
│     └─vg0-lv1                      254:4   0  104M  0 lvm
│       └─vg0-lv1_rimage_0           254:1   0  104M  0 lvm
│         └─vg0-lv1                  254:4   0  104M  0 lvm
├─sdc7                               8:39   0  300M  0 part
│   └─vg0-lv1_rmeta_1                254:2   0    8M  0 lvm
│     └─vg0-lv1                      254:4   0  104M  0 lvm
│       └─vg0-lv1_rimage_1           254:3   0  104M  0 lvm
│         └─vg0-lv1                  254:4   0  104M  0 lvm
├─sdc8                               8:40   0  500M  0 part
└─sdc9                               8:41   0  400M  0 part
```

```
|—sdc10      8:42  0  500M  0 part
|—sdc11      8:43  0  500M  0 part
|—sdc12      8:44  0  200M  0 part
sr0         11:0  1  378M  0 rom
```

La suppression du miroir se fait en utilisant la commande **lvconvert** en indiquant quel volume physique doit être vidé de son contenu :

```
root@debian11:~# lvconvert -m 0 /dev/vg0/lv1 /dev/sdc7
Are you sure you want to convert raid1 LV vg0/lv1 to type linear losing all resilience? [y/n]: y
  Logical volume vg0/lv1 successfully converted.
root@debian11:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda              8:0    0   64G  0 disk
sdb              8:16   0   32G  0 disk
|—sdb1           8:17   0   31G  0 part /
|—sdb2           8:18   0    1K  0 part
|—sdb5           8:21   0  975M  0 part [SWAP]
sdc              8:32   0    4G  0 disk
|—sdc1           8:33   0  100M  0 part
|—sdc2           8:34   0  100M  0 part
|—sdc3           8:35   0  100M  0 part
|—sdc4           8:36   0    1K  0 part
|—sdc5           8:37   0  500M  0 part
|—sdc6           8:38   0  200M  0 part
|  └─vg0-lv1    254:4   0  104M  0 lvm
|—sdc7           8:39   0  300M  0 part
|—sdc8           8:40   0  500M  0 part
|—sdc9           8:41   0  400M  0 part
|—sdc10          8:42   0  500M  0 part
|—sdc11          8:43   0  500M  0 part
|—sdc12          8:44   0  200M  0 part
sr0             11:0   1  378M  0 rom
```

De même, il est possible de créer un miroir pour un volume logique existant :

```
root@debian11:~# lvconvert -m 1 /dev/vg0/lv1
Are you sure you want to convert linear LV vg0/lv1 to raid1 with 2 images enhancing resilience? [y/n]: y
Logical volume vg0/lv1 successfully converted.
root@debian11:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	64G	0	disk	
sdb	8:16	0	32G	0	disk	
├sdb1	8:17	0	31G	0	part	/
├sdb2	8:18	0	1K	0	part	
└sdb5	8:21	0	975M	0	part	[SWAP]
sdc	8:32	0	4G	0	disk	
├sdc1	8:33	0	100M	0	part	
├sdc2	8:34	0	100M	0	part	
├sdc3	8:35	0	100M	0	part	
├sdc4	8:36	0	1K	0	part	
├sdc5	8:37	0	500M	0	part	
├sdc6	8:38	0	200M	0	part	
├├vg0-lv1_rmeta_0	254:0	0	8M	0	lvm	
├├└vg0-lv1	254:4	0	104M	0	lvm	
├├└vg0-lv1_rimage_0	254:1	0	104M	0	lvm	
├├├└vg0-lv1	254:4	0	104M	0	lvm	
├sdc7	8:39	0	300M	0	part	
├├vg0-lv1_rmeta_1	254:2	0	8M	0	lvm	
├├└vg0-lv1	254:4	0	104M	0	lvm	
├├└vg0-lv1_rimage_1	254:3	0	104M	0	lvm	
├├├└vg0-lv1	254:4	0	104M	0	lvm	
├sdc8	8:40	0	500M	0	part	
├sdc9	8:41	0	400M	0	part	
├sdc10	8:42	0	500M	0	part	
├sdc11	8:43	0	500M	0	part	
└sdc12	8:44	0	200M	0	part	
sr0	11:0	1	378M	0	rom	

Supprimez de nouveau votre miroir :

```
root@debian11:~# lvconvert -m 0 /dev/vg0/lv1 /dev/sdc7
Are you sure you want to convert raid1 LV vg0/lv1 to type linear losing all resilience? [y/n]: y
Logical volume vg0/lv1 successfully converted.
```

Les options de la commande **lvconvert** sont :

```
root@debian11:~# lvconvert --help
lvconvert - Change logical volume layout

Convert LV to linear.
lvconvert --type linear LV
    [ COMMON_OPTIONS ]
    [ PV ... ]

Convert LV to striped.
lvconvert --type striped LV
    [ -I|--stripesize Size[k|UNIT] ]
    [ -R|--regionsize Size[m|UNIT] ]
    [ -i|--interval Number ]
    [ --stripes Number ]
    [ COMMON_OPTIONS ]
    [ PV ... ]

Convert LV to type mirror (also see type raid1),
lvconvert --type mirror LV
    [ -m|--mirrors [+|-]Number ]
    [ -I|--stripesize Size[k|UNIT] ]
    [ -R|--regionsize Size[m|UNIT] ]
    [ -i|--interval Number ]
    [ --stripes Number ]
    [ --mirrorlog core|disk ]
    [ COMMON_OPTIONS ]
    [ PV ... ]
```

Convert LV to raid or change raid layout
(a specific raid level must be used, e.g. raid1).

```
lvconvert --type raid LV
  [ -m|--mirrors [+|-]Number ]
  [ -I|--stripesize Size[k|UNIT] ]
  [ -R|--regionsize Size[m|UNIT] ]
  [ -i|--interval Number ]
  [ --stripes Number ]
  [ COMMON_OPTIONS ]
  [ PV ... ]
```

Convert LV to raid1 or mirror, or change number of mirror images.

```
lvconvert -m|--mirrors [+|-]Number LV
  [ -R|--regionsize Size[m|UNIT] ]
  [ -i|--interval Number ]
  [ --mirrorlog core|disk ]
  [ COMMON_OPTIONS ]
  [ PV ... ]
```

Convert raid LV to change number of stripe images.

```
lvconvert --stripes Number LV_raid
  [ -i|--interval Number ]
  [ -R|--regionsize Size[m|UNIT] ]
  [ -I|--stripesize Size[k|UNIT] ]
  [ COMMON_OPTIONS ]
  [ PV ... ]
```

Convert raid LV to change the stripe size.

```
lvconvert -I|--stripesize Size[k|UNIT] LV_raid
  [ -i|--interval Number ]
  [ -R|--regionsize Size[m|UNIT] ]
  [ COMMON_OPTIONS ]
```

Split images from a raid1 or mirror LV and use them to create a new LV.

```
lvconvert --splitmirrors Number -n|--name LV_new LV_cache_mirror_raid1
[ COMMON_OPTIONS ]
[ PV ... ]
```

Split images from a raid1 LV and track changes to origin for later merge.

```
lvconvert --splitmirrors Number --trackchanges LV_cache_raid1
[ COMMON_OPTIONS ]
[ PV ... ]
```

Merge LV images that were split from a raid1 LV.

```
lvconvert --mergemirrors VG|LV_linear_raid|Tag ...
[ COMMON_OPTIONS ]
```

Convert LV to a thin LV, using the original LV as an external origin.

```
lvconvert --type thin --thinpool LV LV_linear_stripped_thin_cache_raid
[ -T|--thin ]
[ -r|--readahead auto|none|Number ]
[ -c|--chunksize Size[k|UNIT] ]
[ -Z|--zero y|n ]
[ --originname LV_new ]
[ --poolmetadata LV ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --metadataprofile String ]
[ COMMON_OPTIONS ]
```

Attach a cache pool to an LV, converts the LV to type cache.

```
lvconvert --type cache --cachepool LV LV_linear_stripped_thinpool_vdo_vdopool_vdopooldata_raid
[ -H|--cache ]
[ -Z|--zero y|n ]
[ -r|--readahead auto|none|Number ]
[ -c|--chunksize Size[k|UNIT] ]
[ --cachemetadatasize Size[m|UNIT] ]
[ --cachemetadataformat auto|1|2 ]
[ --cachemode writethrough|writeback|passthrough ]
```

```
[ --cachepolicy String ]
[ --cachesettings String ]
[ --poolmetadata LV ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ --metadataprofile String ]
[ COMMON_OPTIONS ]
```

Attach a writecache to an LV, converts the LV to type writecache.

```
lvconvert --type writecache --cachevol LV LV_linear_stripped_raid
[ --cachesettings String ]
[ COMMON_OPTIONS ]
```

Attach a cache to an LV, converts the LV to type cache.

```
lvconvert --type cache --cachevol LV LV_linear_stripped_thinpool_raid
[ -H|--cache ]
[ -Z|--zero y|n ]
[ -c|--chunksize Size[k|UNIT] ]
[ --cachemetadataformat auto|1|2 ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
[ --cachesettings String ]
[ --poolmetadatasize Size[m|UNIT] ]
[ COMMON_OPTIONS ]
```

Add a writecache to an LV, using a specified cache device.

```
lvconvert --type writecache --cachedevice PV LV_linear_stripped_raid
[ --cachesize Size[m|UNIT] ]
[ --cachesettings String ]
[ COMMON_OPTIONS ]
```

Add a cache to an LV, using a specified cache device.

```
lvconvert --type cache --cachedevice PV LV_linear_stripped_thinpool_raid
[ -c|--chunksize Size[k|UNIT] ]
```

```
[ --cachesize Size[m|UNIT] ]  
[ --cachesettings String ]  
[ COMMON_OPTIONS ]
```

Convert LV to type thin-pool.

```
lvconvert --type thin-pool LV_linear_striped_cache_raid  
[ -I|--stripesize Size[k|UNIT] ]  
[ -r|--readahead auto|none|Number ]  
[ -c|--chunksize Size[k|UNIT] ]  
[ -Z|--zero y|n ]  
[ --stripes Number ]  
[ --discards passthrough|nopassthrough|ignore ]  
[ --poolmetadata LV ]  
[ --poolmetadatasize Size[m|UNIT] ]  
[ --poolmetadataspare y|n ]  
[ --metadataprofile String ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Convert LV to type cache-pool.

```
lvconvert --type cache-pool LV_linear_striped_raid  
[ -Z|--zero y|n ]  
[ -r|--readahead auto|none|Number ]  
[ -c|--chunksize Size[k|UNIT] ]  
[ --cachemetadataformat auto|1|2 ]  
[ --cachemode writethrough|writeback|passthrough ]  
[ --cachepolicy String ]  
[ --cachesettings String ]  
[ --poolmetadata LV ]  
[ --poolmetadatasize Size[m|UNIT] ]  
[ --poolmetadataspare y|n ]  
[ --metadataprofile String ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Convert LV to type vdopool.

```
lvconvert --type vdo-pool LV_linear_stripped_cache_raid
[ -n|--name LV_new ]
[ -V|--virtualsize Size[m|UNIT] ]
[ --compression y|n ]
[ --deduplication y|n ]
[ COMMON_OPTIONS ]
```

Detach a cache from an LV.

```
lvconvert --splitcache LV_thinpool_cache_cachepool_vdopool_writecache
[ --cachesettings String ]
[ COMMON_OPTIONS ]
```

Merge thin LV into its origin LV.

```
lvconvert --mergethin LV_thin ...
[ COMMON_OPTIONS ]
```

Merge COW snapshot LV into its origin.

```
lvconvert --mergesnapshot LV_snapshot ...
[ -i|--interval Number ]
[ COMMON_OPTIONS ]
```

Combine a former COW snapshot (second arg) with a former origin LV (first arg) to reverse a splitsnapshot command.

```
lvconvert --type snapshot LV LV_linear_stripped
[ -s|--snapshot ]
[ -c|--chunksize Size[k|UNIT] ]
[ -Z|--zero y|n ]
[ COMMON_OPTIONS ]
```

Replace failed PVs in a raid or mirror LV.

Repair a thin pool.

Repair a cache pool.

```
lvconvert --repair LV_thinpool_cache_cachepool_mirror_raid
```

```
[ -i|--interval Number ]
[ --usepolicies ]
[ --poolmetadataspare y|n ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Replace specific PV(s) in a raid LV with another PV.

```
lvconvert --replace PV LV_raid
[ COMMON_OPTIONS ]
[ PV ... ]
```

Poll LV to continue conversion.

```
lvconvert --startpoll LV_mirror_raid
[ COMMON_OPTIONS ]
```

Add or remove data integrity checksums to raid images.

```
lvconvert --raidintegrity y|n LV_raid
[ --raidintegritymode String ]
[ --raidintegrityblocksize Number ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Common options for command:

```
[ -b|--background ]
[ -f|--force ]
[ --alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit ]
[ --noudevsync ]
```

Common options for lvm:

```
[ -d|--debug ]
[ -h|--help ]
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
```

```
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

LAB #6 - Modifier les Attributs LVM

Pour consulter les attributs d'un LV, utilisez la commande **lvs** :

```
root@debian11:~# lvs
LV VG Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv1 vg0 -wi-a----- 104.00m
```



Consultez [cette page](#) pour comprendre les attributs.

La commande équivalente pour les Volume Groups est **vgs** :

```
root@debian11:~# vgs
VG #PV #LV #SN Attr   VSize   VFree
vg0  3  1  0 wz--n- 880.00m 776.00m
```





Consultez [cette page](#) pour comprendre les attributs.

La commande équivalente pour les Physical Volumes est **pvs** :

```
root@debian11:~# pvs
PV          VG  Fmt Attr PSize  PFree
/dev/sdc6   vg0 lvm2 a--  192.00m 88.00m
/dev/sdc7   vg0 lvm2 a--  296.00m 296.00m
/dev/sdc9   vg0 lvm2 a--  392.00m 392.00m
```



Consultez [cette page](#) pour comprendre les attributs.

Les commandes **lvchange**, **vgchange** et **pvchange** permettent de modifier les attributs des Logical Volumes, Volume Groups et Physical Volumes respectivement.

Par exemple, pour rendre inutilisable un Logical Volume, il convient d'enlever l'attribut **a** :

```
root@debian11:~# lvs
LV  VG Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv1  vg0 -wi-a----- 104.00m
root@debian11:~# lvchange -a n /dev/vg0/lv1
root@debian11:~# lvs
LV  VG Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv1  vg0 -wi----- 104.00m
```

L'opération inverse peut être effectuée en utilisant la même commande avec l'argument **y** à la place de **n** :

```
root@debian11:~# lvchange -a y /dev/vg0/lv1
root@debian11:~# lvs
LV  VG Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
```

```
lv1 vg0 -wi-a----- 104.00m
```

Les options de la commande **lvchange** sont :

```
root@debian11:~# lvchange --help
lvchange - Change the attributes of logical volume(s)

Change a general LV attribute.
For options listed in parentheses, any one is
required, after which the others are optional.
lvchange
  ( -C|--contiguous y|n,
    -p|--permission rw|r,
    -r|--readahead auto|none|Number,
    -k|--setactivationskip y|n,
    -Z|--zero y|n,
    -M|--persistent n,
      --addtag Tag,
      --deltag Tag,
      --alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit,
      --compression y|n,
      --deduplication y|n,
      --detachprofile,
      --metadataprofile String,
      --profile String,
      --errorwhenfull y|n,
      --discards passdown|nopassdown|ignore,
      --cachemode writethrough|writeback|passthrough,
      --cachepolicy String,
      --cachesettings String,
      --minrecoveryrate Size[k|UNIT],
      --maxrecoveryrate Size[k|UNIT],
      --writebehind Number,
      --writemostly PV[:t|n|y] )
```

```
VG|LV|Tag|Select ...
[ -a|--activate y|n|ay ]
[   --poll y|n ]
[   --monitor y|n ]
[ COMMON_OPTIONS ]
```

Resynchronize a mirror or raid LV.

Use to reset 'R' attribute on a not initially synchronized LV.

```
lvchange --resync VG|LV_mirror_raid|Tag|Select ...
[ -a|--activate y|n|ay ]
[ COMMON_OPTIONS ]
```

Resynchronize or check a raid LV.

```
lvchange --syncaction check|repair VG|LV_raid|Tag|Select ...
[ COMMON_OPTIONS ]
```

Reconstruct data on specific PVs of a raid LV.

```
lvchange --rebuild PV VG|LV_raid|Tag|Select ...
[ COMMON_OPTIONS ]
```

Activate or deactivate an LV.

```
lvchange -a|--activate y|n|ay VG|LV|Tag|Select ...
[ -P|--partial ]
[ -K|--ignoreactivationskip ]
[   --activationmode partial|degraded|complete ]
[   --poll y|n ]
[   --monitor y|n ]
[   --ignorelockingfailure ]
[   --sysinit ]
[   --readonly ]
[ COMMON_OPTIONS ]
```

Reactivate an LV using the latest metadata.

```
lvchange --refresh VG|LV|Tag|Select ...
```

```
[ -P|--partial ]  
[ --activationmode partial|degraded|complete ]  
[ --poll y|n ]  
[ --monitor y|n ]  
[ COMMON_OPTIONS ]
```

Start or stop monitoring an LV from dmeventd.

```
lvchange --monitor y|n VG|LV|Tag|Select ...  
[ COMMON_OPTIONS ]
```

Start or stop processing an LV conversion.

```
lvchange --poll y|n VG|LV|Tag|Select ...  
[ --monitor y|n ]  
[ COMMON_OPTIONS ]
```

Make the minor device number persistent for an LV.

```
lvchange -M|--persistent y --minor Number LV  
[ -j|--major Number ]  
[ -a|--activate y|n|ay ]  
[ --poll y|n ]  
[ --monitor y|n ]  
[ COMMON_OPTIONS ]
```

Common options for command:

```
[ -A|--autobackup y|n ]  
[ -f|--force ]  
[ -S|--select String ]  
[ --ignoremonitoring ]  
[ --noudevsync ]  
[ --reportformat basic|json ]
```

Common options for lvm:

```
[ -d|--debug ]  
[ -h|--help ]
```

```
[ -q|--quiet ]
[ -v|--verbose ]
[ -y|--yes ]
[ -t|--test ]
[ --commandprofile String ]
[ --config String ]
[ --driverloaded y|n ]
[ --nolocking ]
[ --lockopt String ]
[ --longhelp ]
[ --profile String ]
[ --version ]
```

Use `--longhelp` to show all options and advanced commands.

LAB #7 - Volumes Logiques en Bandes

Un volume logique en bandes est créé pour augmenter, comme dans le cas du RAID, les performances des entrées et sorties. Pour créer ce volume, la commande **lvcreate** prend deux options supplémentaires :

- **-i** - indique le nombre de volumes de bandes,
- **-I** - indique la taille en Ko de chaque bande.

Saisissez donc la commande suivante :

```
root@debian11:~# lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdc7 /dev/sdc9
Rounding up size to full physical extent 104.00 MiB
Rounding size 104.00 MiB (13 extents) up to stripe boundary size 112.00 MiB (14 extents).
Logical volume "lv2" created.
```

Constatez la présence de vos bandes sur `/dev/sdc7` et sur `/dev/sdc9` :

```
root@debian11:~# lvdisplay -m /dev/vg0/lv2
```

```
--- Logical volume ---
LV Path                /dev/vg0/lv2
LV Name                lv2
VG Name                vg0
LV UUID                gtqCux-8FIn-gCLc-35oB-TTsC-k7AZ-3PHIJI
LV Write Access        read/write
LV Creation host, time debian11, 2022-04-26 16:33:17 +0200
LV Status              available
# open                 0equivalente
LV Size                112.00 MiB
Current LE             14
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    512
Block device           254:1
```

```
--- Segments ---
Logical extents 0 to 13:
  Type                 striped
  Stripes               2
  Stripe size          64.00 KiB
  Stripe 0:
    Physical volume    /dev/sdc7
    Physical extents   0 to 6
  Stripe 1:
    Physical volume    /dev/sdc9
    Physical extents   0 to 6
```

```
root@debian11:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   64G  0 disk
sdb          8:16   0   32G  0 disk
├─sdb1       8:17   0   31G  0 part /
└─sdb2       8:18   0    1K  0 part
```

```
└─sdb5      8:21  0  975M  0 part [SWAP]
sdc         8:32  0    4G  0 disk
├─sdc1      8:33  0  100M  0 part
├─sdc2      8:34  0  100M  0 part
├─sdc3      8:35  0  100M  0 part
├─sdc4      8:36  0    1K  0 part
├─sdc5      8:37  0  500M  0 part
├─sdc6      8:38  0  200M  0 part
│   └─vg0-lv1 254:0  0  104M  0 lvm
├─sdc7      8:39  0  300M  0 part
│   └─vg0-lv2 254:1  0  112M  0 lvm
├─sdc8      8:40  0  500M  0 part
├─sdc9      8:41  0  400M  0 part
│   └─vg0-lv2 254:1  0  112M  0 lvm
├─sdc10     8:42  0  500M  0 part
├─sdc11     8:43  0  500M  0 part
└─sdc12     8:44  0  200M  0 part
sr0        11:0  1  378M  0 rom
```

Utilisez maintenant la commande **lvs** pour visualiser les volumes physiques utilisés par le volume logique :

```
root@debian11:~# lvs -o +devices
LV VG Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
lv1 vg0 -wi-a----- 104.00m                               /dev/sdc6(1)
lv2 vg0 -wi-a----- 112.00m                               /dev/sdc7(0),/dev/sdc9(0)
```

LAB #8 - Gérer les Métadonnées

Les métadonnées pour chaque Volume Group sont stockés dans un fichier texte au nom du Volume Group dans le répertoire **/etc/lvm/backup** :

```
root@debian11:~# cat /etc/lvm/backup/vg0
# Generated by LVM2 version 2.03.11(2) (2021-01-08): Tue Apr 26 16:33:17 2022
```

```
contents = "Text Format Volume Group"
version = 1

description = "Created *after* executing 'lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdc7 /dev/sdc9'"

creation_host = "debian11"      # Linux debian11 5.10.0-13-amd64 #1 SMP Debian 5.10.106-1 (2022-03-17) x86_64
creation_time = 1650983597      # Tue Apr 26 16:33:17 2022

vg0 {
    id = "0WzAzT-5kjC-Hsld-MCo1-Z1Qr-zQNZ-XmXsdr"
    seqno = 11
    format = "lvm2"              # informational
    status = ["RESIZEABLE", "READ", "WRITE"]
    flags = []
    extent_size = 16384          # 8 Megabytes
    max_lv = 0
    max_pv = 0
    metadata_copies = 0

    physical_volumes {

        pv0 {
            id = "1J010Q-CM90-tKxI-0sM6-0vbe-3eDG-S10H6d"
            device = "/dev/sdc6"  # Hint only

            status = ["ALLOCATABLE"]
            flags = []
            dev_size = 409600      # 200 Megabytes
            pe_start = 2048
            pe_count = 24         # 192 Megabytes
        }

        pv1 {
            id = "GEk0IP-S7ce-8S1G-K0TX-ocxE-Ud6y-IY3f0Z"
```

```
        device = "/dev/sdc7"    # Hint only

        status = ["ALLOCATABLE"]
        flags = []
        dev_size = 614400      # 300 Megabytes
        pe_start = 2048
        pe_count = 37    # 296 Megabytes
    }

    pv2 {
        id = "J7UiEX-m983-j1fp-rU7x-TuCh-MFKh-s105M0"
        device = "/dev/sdc9"    # Hint only

        status = ["ALLOCATABLE"]
        flags = []
        dev_size = 819200      # 400 Megabytes
        pe_start = 2048
        pe_count = 49    # 392 Megabytes
    }
}

logical_volumes {

    lv1 {
        id = "2AQE1P-kcp7-5w50-9i3M-Ge6L-0Sa7-HDmKii"
        status = ["READ", "WRITE", "VISIBLE"]
        flags = []
        creation_time = 1650982876      # 2022-04-26 16:21:16 +0200
        creation_host = "debian11"
        segment_count = 1

        segment1 {
            start_extent = 0
            extent_count = 13      # 104 Megabytes
        }
    }
}
```

```
        type = "striped"
        stripe_count = 1          # linear

        stripes = [
            "pv0", 1
        ]
    }
}

lv2 {
    id = "gtqCux-8FIn-gCLc-35oB-TTsC-k7AZ-3PHIJI"
    status = ["READ", "WRITE", "VISIBLE"]
    flags = []
    creation_time = 1650983597    # 2022-04-26 16:33:17 +0200
    creation_host = "debian11"
    segment_count = 1

    segment1 {
        start_extent = 0
        extent_count = 14        # 112 Megabytes

        type = "striped"
        stripe_count = 2
        stripe_size = 128        # 64 Kilobytes

        stripes = [
            "pv1", 0,
            "pv2", 0
        ]
    }
}
}
}
```

Des archives sont créées lors de chaque modification d'un groupe de volumes et elles sont placés dans le répertoire **/etc/lvm/archives** :

```
root@debian11:~# ls /etc/lvm/archive/
vg0_00000-267942700.vg  vg0_00004-458787361.vg  vg0_00008-297779072.vg  vg0_00012-1101644815.vg
vg0_00001-854434220.vg  vg0_00005-1786773709.vg  vg0_00009-1557237202.vg
vg0_00002-520659205.vg  vg0_00006-196117920.vg  vg0_00010-550024633.vg
vg0_00003-1606608177.vg  vg0_00007-2024993792.vg  vg0_00011-155655591.vg

root@debian11:~# vgcfgrestore --list vg0
File:          /etc/lvm/archive/vg0_00000-267942700.vg
VG name:       vg0
Description:   Created *before* executing 'vgcreate -s 8M vg0 /dev/sdc6 /dev/sdc7 /dev/sdc9'
Backup Time:  Tue Apr 26 13:54:06 2022

File:          /etc/lvm/archive/vg0_00001-854434220.vg
VG name:       vg0
Description:   Created *before* executing 'lvcreate -L 350 -n lv0 vg0'
Backup Time:  Tue Apr 26 13:55:59 2022

File:          /etc/lvm/archive/vg0_00002-520659205.vg
VG name:       vg0
Description:   Created *before* executing 'lvextend -L +100M /dev/vg0/lv0'
Backup Time:  Tue Apr 26 15:47:38 2022

File:          /etc/lvm/archive/vg0_00003-1606608177.vg
VG name:       vg0
Description:   Created *before* executing 'lvcreate -s -L 20M -n testsnap /dev/vg0/lv0'
Backup Time:  Tue Apr 26 15:53:12 2022

File:          /etc/lvm/archive/vg0_00004-458787361.vg
VG name:       vg0
Description:   Created *before* executing 'lvremove /dev/vg0/testsnap'
Backup Time:  Tue Apr 26 16:15:45 2022
```

File: /etc/lvm/archive/vg0_00005-1786773709.vg
VG name: vg0
Description: Created *before* executing 'lvremove /dev/vg0/lv0'
Backup Time: Tue Apr 26 16:16:19 2022

File: /etc/lvm/archive/vg0_00006-196117920.vg
VG name: vg0
Description: Created *before* executing 'vgremove vg0'
Backup Time: Tue Apr 26 16:17:28 2022

File: /etc/lvm/archive/vg0_00007-2024993792.vg
VG name: vg0
Description: Created *before* executing 'vgcreate -s 8M vg0 /dev/sdc6 /dev/sdc7 /dev/sdc9'
Backup Time: Tue Apr 26 16:20:56 2022

File: /etc/lvm/archive/vg0_00008-297779072.vg
VG name: vg0
Description: Created *before* executing 'lvcreate -m 1 -L 100M -n lv1 vg0'
Backup Time: Tue Apr 26 16:21:16 2022

File: /etc/lvm/archive/vg0_00009-1557237202.vg
VG name: vg0
Description: Created *before* executing 'lvconvert -m 0 /dev/vg0/lv1 /dev/sdc7'
Backup Time: Tue Apr 26 16:24:33 2022

File: /etc/lvm/archive/vg0_00010-550024633.vg
VG name: vg0
Description: Created *before* executing 'lvconvert -m 1 /dev/vg0/lv1'
Backup Time: Tue Apr 26 16:25:20 2022

File: /etc/lvm/archive/vg0_00011-155655591.vg
VG name: vg0
Description: Created *before* executing 'lvconvert -m 0 /dev/vg0/lv1 /dev/sdc7'
Backup Time: Tue Apr 26 16:25:49 2022

```
File:          /etc/lvm/archive/vg0_00012-1101644815.vg
VG name:       vg0
Description:   Created *before* executing 'lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdc7 /dev/sdc9'
Backup Time:  Tue Apr 26 16:33:17 2022

File:          /etc/lvm/backup/vg0
VG name:       vg0
Description:   Created *after* executing 'vgcfgbackup vg0'
Backup Time:  Tue Apr 26 16:37:00 2022
```

La commande **vgcfgbackup** est utilisée pour sauvegarder les métadonnées manuellement dans le fichier **/etc/lvm/backup/nom_du_volume_group** :

La commande **vgcfgrestore** permet de restaurer une sauvegarde. Avec l'option **-list**, cette commande renvoie la liste des sauvegardes disponibles :

```
root@debian11:~# vgcfgbackup vg0
Volume group "vg0" successfully backed up.

root@debian11:~# ls /etc/lvm/backup/
vg0
```

Il est aussi possible de modifier l'emplacement de la sauvegarde avec l'option **-f** de la commande :

```
root@debian11:~# vgcfgbackup -f /tmp/vg0_backup vg0
Volume group "vg0" successfully backed up.

root@debian11:~# ls /tmp
systemd-private-7644749265b24b9a8f6a8695c083cfaa-ModemManager.service-KFBiWe
systemd-private-7644749265b24b9a8f6a8695c083cfaa-systemd-logind.service-3fbzgg
systemd-private-7644749265b24b9a8f6a8695c083cfaa-systemd-timesyncd.service-Gyzzrhf
vg0_backup
```

Concepts RAID

Les solutions RAID ou *Redundant Array of Independent Disks* ou encore *Redundant Array of Inexpensive Disks* permettent la combinaison de plusieurs disques de façon à ce que ceux-ci soient vu comme un seul disque logique.

Les solutions RAID sont issues du travail fourni par l'université de Berkeley en Californie sur un projet de tolérances de pannes. Les systèmes RAID offre maintenant plusieurs avantages :

- Addition des capacités,
- Amélioration des performances,
- Apporter la tolérance de panne.

Deux concepts sont fondamentaux à la compréhension des solutions RAID.

Disques en miroir

La technique des disques en miroir consiste à dupliquer l'écriture des données sur plusieurs disques. Le miroir peut être géré par un logiciel ou par du matériel.

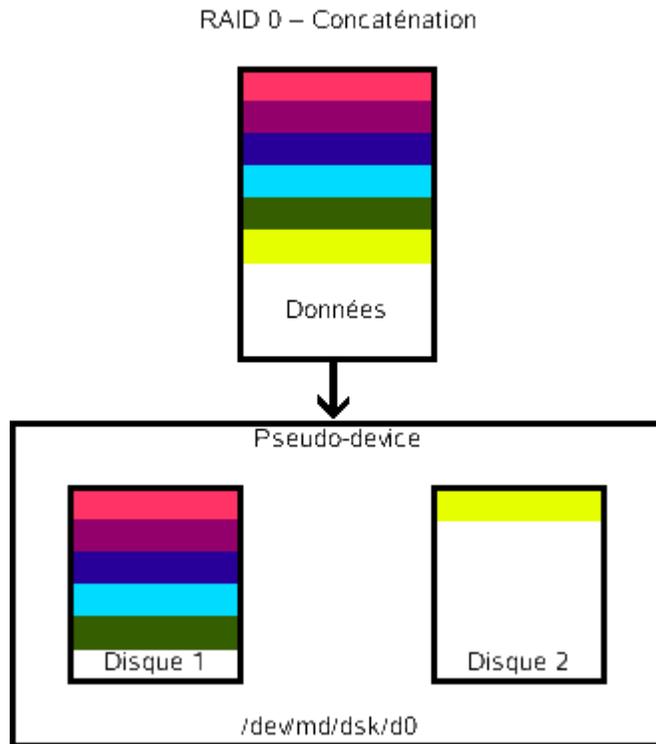
Bandes de données

La technique des bandes de données, autrement appelée *data striping* consiste à couper les données à enregistrer en segments séquentiels et contigus pour les enregistrer sur plusieurs disques physiques. L'ensemble des segments constitue alors un disque logique ou *striped disk*. Cette technique peut être améliorée en déposant une bande de parité, calculée à partir des données des autres bandes, afin de pouvoir reconstituer une bande de données défaillante.

Types de RAID

RAID 0 - Concaténation

Création de volume par récupération de l'espace libre sur un ou plusieurs disques. Le principe de la concaténation est la création d'un volume à bandes où chaque bande est une tranche.



Avantages

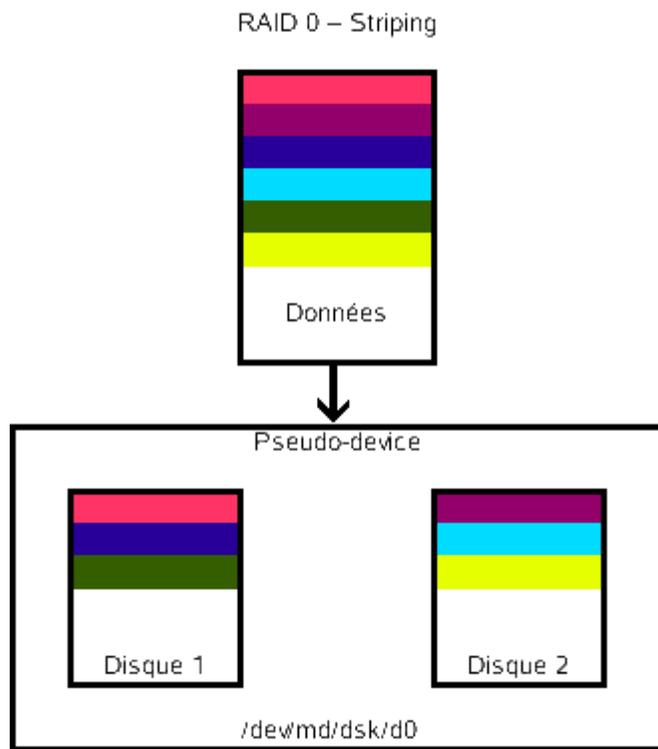
- Récupération de l'espace disque.

Inconvénients

- Pas de protection des données,
- Pas d'augmentation des performances d'E/S.

RAID 0 - Striping

Création de volume sur plusieurs disques afin d'augmenter les performances d'E/S. Le principe du striping est la création d'un volume à bandes réparties sur plusieurs tranches. La taille de la bande doit être fonction des données à écrire sur le volume (16k, 32k, 64k, etc.) Cette taille est choisie à la création du volume.



Avantages

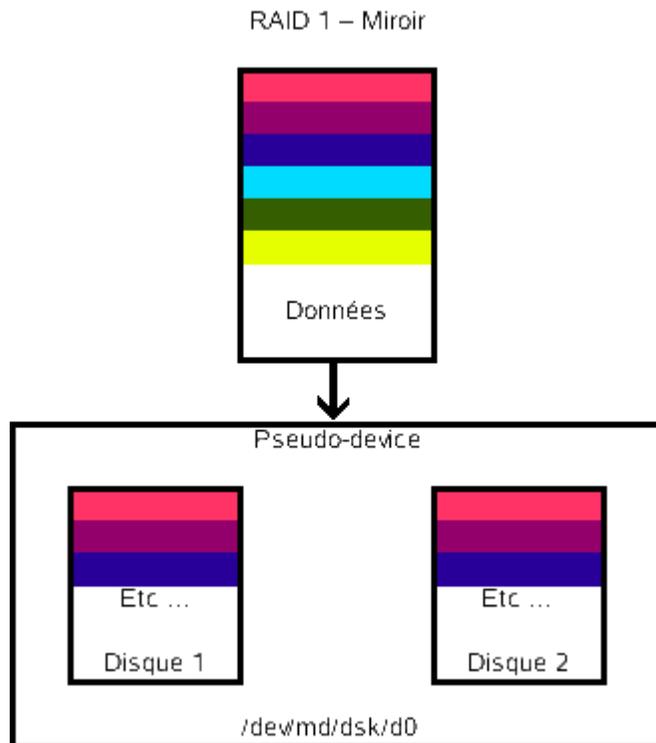
- Augmentation des performances d'E/S par écriture en parallèle sur les disques.

Inconvénients

- Pas de protection des données.

RAID 1 - Miroir

Création d'un volume où les disques sont en miroir. Quand les deux disques sont connectés à des contrôleurs de disques différents, on parle de *duplexing* :



Avantages

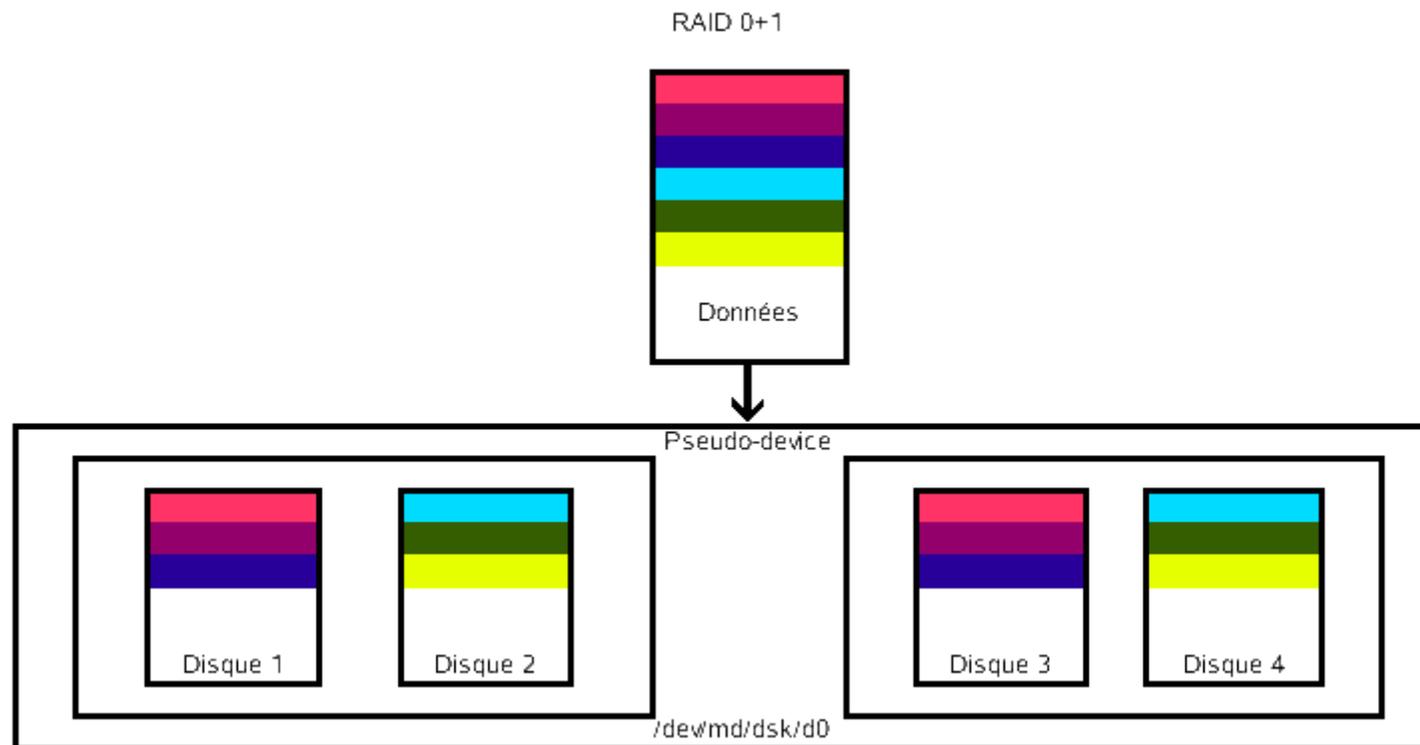
- Protection des données contre une défaillance d'un disque.

Inconvénients

- Coûteux à cause de l'augmentation du nombre de disques.

RAID 1+0 - Striping en Miroir

Le RAID 1+0 ou encore 0+1 est une technique qui réunit le RAID 0 et le RAID 1. On l'appelle aussi un RAID **exotique**:



Avantages

- Protection des données contre une défaillance d'un disque.
- Augmentation des performances d'E/S par écriture en parallèle sur les disques.

Inconvénients

- Coûteux à cause de l'augmentation du nombre de disques.

RAID 2 - Miroir avec Contrôle d'Erreurs

Le RAID 2 est une technique de miroir avec contrôle de correction d'erreurs (EEC). De nos jours cette technique est peu utilisée, ayant été remplacée par les RAID 3, 4 et 5.

RAID 3 et 4 - Striping avec Parité

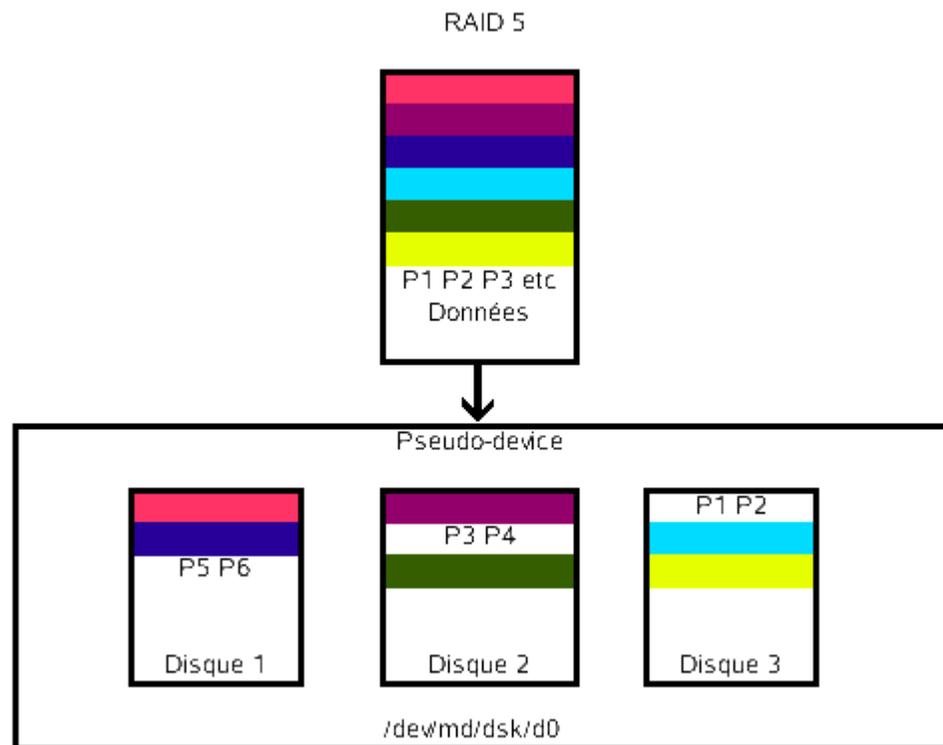
Les RAID 3 et 4 sont des technologies avec bandes de parité distribuées sur un seul disque :

Inconvénients

- Création d'un goulot d'étranglement des données à cause de l'écriture des données de parité sur un seul disque.

RAID 5 - Striping avec Parité Distribuée

Le RAID 5 est une technologie avec bandes de parité distribuées sur plusieurs disques :



Avantages

- Protection des données contre une défaillance d'un disque,
- Evite le goulot d'étranglement d'un seul disque de parité.

Inconvénients

- Lecture moins performante qu'avec RAID 3 et 4.

Au delà de RAID 5

Il existe aussi deux autres technologies RAID, toute deux issues de la technologie RAID 5 :

- RAID 6
 - *Disk Striping with Double Distributed Parity*
- RAID TP
 - *Disk Striping with Triple Distributed Parity*

LAB #9 - Mise en Place du RAID 5 Logiciel

9.1 - Préparer le disque

Rappelez-vous avoir modifié les types de 4 partitions du disque **/dev/sdc** en **fd** :

```
root@debian11:~# fdisk -l
Disk /dev/sda: 32 GiB, 34359738368 bytes, 67108864 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: dos
Disk identifier: 0xf2e3a71a
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	65107967	65105920	31G	83	Linux
/dev/sda2		65110014	67106815	1996802	975M	5	Extended
/dev/sda5		65110016	67106815	1996800	975M	82	Linux swap / Solaris

```
Disk /dev/sdb: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: QEMU HARDDISK
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/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	fd	Linux raid autodetect
/dev/sdc6		1644544	2054143	409600	200M	8e	Linux LVM
/dev/sdc7		2056192	2670591	614400	300M	8e	Linux LVM
/dev/sdc8		2672640	3696639	1024000	500M	fd	Linux raid autodetect
/dev/sdc9		3698688	4517887	819200	400M	8e	Linux LVM

```
/dev/sdc10      4519936 5543935 1024000 500M fd Linux raid autodetect
/dev/sdc11      5545984 6569983 1024000 500M fd Linux raid autodetect
/dev/sdc12      6572032 6981631  409600  200M 83 Linux
```

```
Disk /dev/mapper/vg0-lv1: 104 MiB, 109051904 bytes, 212992 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/vg0-lv2: 112 MiB, 117440512 bytes, 229376 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 65536 bytes / 131072 bytes
```

```
Disk /dev/mapper/sdc11: 484 MiB, 507510784 bytes, 991232 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
```

Dans le cas de cet exemple les quatre partitions concernées par la mise en place d'un RAID 5 sont :

```
/dev/sdc5       618496 1642495 1024000 500M fd Linux raid autodetect
/dev/sdc8       2672640 3696639 1024000 500M fd Linux raid autodetect
/dev/sdc10      4519936 5543935 1024000 500M fd Linux raid autodetect
/dev/sdc11      5545984 6569983 1024000 500M fd Linux raid autodetect
```

9.2 - Créer une Unité RAID

La création d'une unité RAID avec la commande **mdadm** se fait grâce aux options passées en arguments à la commande :

```
mdadm --create <unité RAID> [options] <unités physiques>
```

Sous Debian 11, mdadm n'est pas installé par défaut :

```
root@debian11:~# apt-get -y install mdadm
```

Saisissez maintenant la commande suivante :

```
root@debian11:~# mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sdc5 /dev/sdc8 /dev/sdc10
mdadm: /dev/sdc8 appears to contain a reiserfs file system
      size = 512000K
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
```

Les options dans la ligne de commande sont :

Option Courte	Option Longue	Description
-l	--level	Le niveau RAID - linear, 0,1,2,4 ou 5
-n	--raid-devices=<nombre>	Le nombre de périphériques actifs dans le RAID

Les options de la commande **mdadm** sont :

```
root@debian11:~# mdadm --help-options
Any parameter that does not start with '-' is treated as a device name
or, for --examine-bitmap, a file name.
The first such name is often the name of an md device. Subsequent
names are often names of component devices.
```

Some common options are:

```
--help      -h      : General help message or, after above option,
                    mode specific help message
--help-options : This help message
```

```
--version      -V  : Print version information for mdadm
--verbose      -v  : Be more verbose about what is happening
--quiet        -q  : Don't print un-necessary messages
--brief        -b  : Be less verbose, more brief
--export       -Y  : With --detail, --detail-platform or --examine use
                   key=value format for easy import into environment
--force        -f  : Override normal checks and be more forceful

--assemble     -A  : Assemble an array
--build        -B  : Build an array without metadata
--create       -C  : Create a new array
--detail       -D  : Display details of an array
--examine      -E  : Examine superblock on an array component
--examine-bitmap -X: Display the detail of a bitmap file
--examine-badblocks: Display list of known bad blocks on device
--monitor      -F  : monitor (follow) some arrays
--grow         -G  : resize/ reshape and array
--incremental  -I  : add/remove a single device to/from an array as appropriate
--query        -Q  : Display general information about how a
                   device relates to the md driver
--auto-detect  : Start arrays auto-detected by the kernel
```

La commande **mdadm** utilise des sous-commandes ou *mode majeurs* :

```
root@debian11:~# mdadm --help
mdadm is used for building, managing, and monitoring
Linux md devices (aka RAID arrays)
Usage: mdadm --create device options...
        Create a new array from unused devices.
mdadm --assemble device options...
        Assemble a previously created array.
mdadm --build device options...
        Create or assemble an array without metadata.
mdadm --manage device options...
```

```
    make changes to an existing array.
mdadm --misc options... devices
    report on or modify various md related devices.
mdadm --grow options device
    resize/reshape an active array
mdadm --incremental device
    add/remove a device to/from an array as appropriate
mdadm --monitor options...
    Monitor one or more array for significant changes.
mdadm device options...
    Shorthand for --manage.
```

Any parameter that does not start with '-' is treated as a device name or, for --examine-bitmap, a file name. The first such name is often the name of an md device. Subsequent names are often names of component devices.

For detailed help on the above major modes use --help after the mode e.g.

```
mdadm --assemble --help
```

For general help on options use

```
mdadm --help-options
```

Chaque sous-commande bénéficie d'un aide spécifique, par exemple :

```
root@debian11:~# mdadm --create --help
Usage: mdadm --create device --chunk=X --level=Y --raid-devices=Z devices
```

This usage will initialise a new md array, associate some devices with it, and activate the array. In order to create an array with some devices missing, use the special word 'missing' in place of the relevant device name.

Before devices are added, they are checked to see if they already contain raid superblocks or filesystems. They are also checked to see if

the variance in device size exceeds 1%.

If any discrepancy is found, the user will be prompted for confirmation before the array is created. The presence of a '--run' can override this caution.

If the --size option is given then only that many kilobytes of each device is used, no matter how big each device is.

If no --size is given, the apparent size of the smallest drive given is used for raid level 1 and greater, and the full device is used for other levels.

Options that are valid with --create (-C) are:

```
--bitmap=          -b : Create a bitmap for the array with the given filename
                   : or an internal bitmap if 'internal' is given
--chunk=           -c : chunk size in kibibytes
--rounding=        : rounding factor for linear array (==chunk size)
--level=           -l : raid level: 0,1,4,5,6,10,linear,multipath and synonyms
--parity=          -p : raid5/6 parity algorithm: {left,right}-{,a}symmetric
--layout=          : same as --parity, for RAID10: [fno]NN
--raid-devices=    -n : number of active devices in array
--spare-devices=   -x : number of spare (eXtra) devices in initial array
--size=            -z : Size (in K) of each drive in RAID1/4/5/6/10 - optional
--data-offset=     : Space to leave between start of device and start
                   : of array data.
--force            -f : Honour devices as listed on command line. Don't
                   : insert a missing drive for RAID5.
--run              -R : insist of running the array even if not all
                   : devices are present or some look odd.
--readonly         -o : start the array readonly - not supported yet.
--name=           -N : Textual name for array - max 32 characters
--bitmap-chunk=    : bitmap chunksize in Kilobytes.
--delay=          -d : bitmap update delay in seconds.
--write-journal=   : Specify journal device for RAID-4/5/6 array
--consistency-policy= : Specify the policy that determines how the array
```

-k : maintains consistency in case of unexpected shutdown.

Les *modes majeurs* de la commande **mdadm** peuvent être visualisés grâce à la commande suivante :

```
root@debian8:~# mdadm --help
mdadm is used for building, managing, and monitoring
Linux md devices (aka RAID arrays)
Usage: mdadm --create device options...
        Create a new array from unused devices.
mdadm --assemble device options...
        Assemble a previously created array.
mdadm --build device options...
        Create or assemble an array without metadata.
mdadm --manage device options...
        make changes to an existing array.
mdadm --misc options... devices
        report on or modify various md related devices.
mdadm --grow options device
        resize/reshape an active array
mdadm --incremental device
        add/remove a device to/from an array as appropriate
mdadm --monitor options...
        Monitor one or more array for significant changes.
mdadm device options...
        Shorthand for --manage.
Any parameter that does not start with '-' is treated as a device name
or, for --examine-bitmap, a file name.
The first such name is often the name of an md device. Subsequent
names are often names of component devices.

For detailed help on the above major modes use --help after the mode
e.g.
    mdadm --assemble --help
For general help on options use
```

mdadm --help-options

Constatez maintenant les informations concernant le RAID 5 créé :

```
root@debian11:~# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md1 : active raid5 sdc10[3] sdc8[1] sdc5[0]
      1019904 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
```

Grâce à la commande mdadm, il est possible d'obtenir d'avantage d'informations :

```
root@debian11:~# mdadm --query /dev/md1
/dev/md1: 996.00MiB raid5 3 devices, 0 spares. Use mdadm --detail for more detail.
```

L'option - **detail** produit le résultat suivant :

```
root@debian11:~# mdadm --detail /dev/md1
/dev/md1:
      Version : 1.2
  Creation Time : Sun May  1 13:27:48 2022
    Raid Level : raid5
    Array Size : 1019904 (996.00 MiB 1044.38 MB)
  Used Dev Size : 509952 (498.00 MiB 522.19 MB)
    Raid Devices : 3
   Total Devices : 3
 Persistence : Superblock is persistent

   Update Time : Sun May  1 13:27:53 2022
     State : clean
 Active Devices : 3
 Working Devices : 3
  Failed Devices : 0
   Spare Devices : 0
```

```
Layout : left-symmetric
Chunk Size : 512K
```

```
Consistency Policy : resync
```

```
Name : debian11:1 (local to host debian11)
UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2
Events : 18
```

Number	Major	Minor	RaidDevice	State	
0	8	37	0	active sync	/dev/sdc5
1	8	40	1	active sync	/dev/sdc8
3	8	42	2	active sync	/dev/sdc10

Notez la ligne **Persistence : Superblock is persistent**. En effet, cette implémentation de RAID inscrit les caractéristiques du volume dans un *super bloc* persistant en début de chaque unité de type bloc dans le volume.

Cependant, il est nécessaire de renseigner le fichier **/etc/mdadm/mdadm.conf** afin que le RAID soit construit à chaque démarrage :

```
root@debian11:~# cat /etc/mdadm/mdadm.conf
# mdadm.conf
#
# !NB! Run update-initramfs -u after updating this file.
# !NB! This will ensure that initramfs has an uptodate copy.
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
#DEVICE partitions containers

# automatically tag new arrays as belonging to the local system
```

```
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays

# This configuration was auto-generated on Sun, 01 May 2022 13:26:29 +0200 by mkconf
```

Ecrasez le contenu de ce fichier avec les informations suivantes :

```
root@debian11:~# echo 'DEVICE /dev/sdc5 /dev/sdc8 /dev/sdc10' > /etc/mdadm/mdadm.conf
root@debian11:~# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
root@debian11:~# cat /etc/mdadm/mdadm.conf
DEVICE /dev/sdc5 /dev/sdc8 /dev/sdc10
ARRAY /dev/md1 metadata=1.2 name=debian11:1 UUID=c0f945a0:f65b2136:b7913f8a:3707ffa2
```

Mettez à jour l'initramfs :

```
root@debian11:~# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-5.10.0-13-amd64
```

Chaque unité peut être examinée individuellement :

```
root@debian11:~# mdadm --examine /dev/sdc5
/dev/sdc5:
    Magic : a92b4efc
    Version : 1.2
    Feature Map : 0x0
    Array UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2
    Name : debian11:1 (local to host debian11)
    Creation Time : Sun May 1 13:27:48 2022
    Raid Level : raid5
    Raid Devices : 3
```

```
Avail Dev Size : 1019904 (498.00 MiB 522.19 MB)
  Array Size : 1019904 (996.00 MiB 1044.38 MB)
  Data Offset : 4096 sectors
  Super Offset : 8 sectors
  Unused Space : before=4016 sectors, after=0 sectors
  State : clean
  Device UUID : 1d34dda2:28775dbb:53d242e9:9acba5dd

  Update Time : Sun May  1 13:27:53 2022
  Bad Block Log : 512 entries available at offset 16 sectors
  Checksum : 31909df9 - correct
  Events : 18

  Layout : left-symmetric
  Chunk Size : 512K

  Device Role : Active device 0
  Array State : AAA ('A' == active, '.' == missing, 'R' == replacing)
```

9.3 - Remplacer une Unité Défaillante

A ce stade il est intéressant de noter comment réagir lors d'une défaillance d'un disque. Dans notre cas nous allons indiquer au système que la partition `/dev/sdc5` est devenue défaillante :

```
root@debian11:~# mdadm --manage --set-faulty /dev/md1 /dev/sdc5
mdadm: set /dev/sdc5 faulty in /dev/md1
```

L'utilisation de la ligne de commande suivante nous confirme le statut de `/dev/sdc5` :

```
root@debian11:~# mdadm --detail /dev/md1
/dev/md1:
  Version : 1.2
```

```
Creation Time : Sun May 1 13:27:48 2022
Raid Level : raid5
Array Size : 1019904 (996.00 MiB 1044.38 MB)
Used Dev Size : 509952 (498.00 MiB 522.19 MB)
Raid Devices : 3
Total Devices : 3
Persistence : Superblock is persistent
```

```
Update Time : Sun May 1 13:43:24 2022
State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0
```

```
Layout : left-symmetric
Chunk Size : 512K
```

```
Consistency Policy : resync
```

```
Name : debian11:1 (local to host debian11)
UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2
Events : 20
```

Number	Major	Minor	RaidDevice	State	
-	0	0	0	removed	
1	8	40	1	active sync	/dev/sdc8
3	8	42	2	active sync	/dev/sdc10
0	8	37	-	faulty	/dev/sdc5

Il est maintenant nécessaire de supprimer /dev/sdc5 de notre RAID 5 :

```
root@debian11:~# mdadm --manage --remove /dev/md1 /dev/sdc5
```

```
mdadm: hot removed /dev/sdc5 from /dev/md1
```

A l'examen de notre RAID, on constate que /dev/sdc5 a été supprimé :

```
root@debian11:~# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
    Creation Time : Sun May  1 13:27:48 2022
    Raid Level : raid5
    Array Size : 1019904 (996.00 MiB 1044.38 MB)
    Used Dev Size : 509952 (498.00 MiB 522.19 MB)
    Raid Devices : 3
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Sun May  1 13:44:41 2022
    State : clean, degraded
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Layout : left-symmetric
    Chunk Size : 512K

Consistency Policy : resync

    Name : debian11:1 (local to host debian11)
    UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2
    Events : 21

    Number Major Minor RaidDevice State
     -     0      0        0     removed
     1     8     40        1     active sync  /dev/sdc8
```

```
3      8      42      2      active sync  /dev/sdc10
```

Constatez maintenant l'existence de votre RAID :

```
root@debian11:~# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md1 : active raid5 sdc10[3] sdc8[1]
      1019904 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
unused devices: <none>
```



IMPORTANT - Notez que le RAID a 2 unités au lieu de trois.

Nous avons déjà utilisé **/dev/sdc11** pour héberger LUKs. Constatez le statut de celui-ci :

```
root@debian11:~# umount /dev/sdc11

root@debian11:~# cryptsetup status sdc11
/dev/mapper/sdc11 is active.
 type:    LUKS2
 cipher:  aes-xts-plain64
 keysize: 512 bits
 key location: keyring
 device:  /dev/sdc11
 sector size: 512
 offset:  32768 sectors
 size:    991232 sectors
 mode:    read/write
```

Avant de supprimer LUKs, il convient de supprimer la dernière passphrase :

```
root@debian11:~# cryptsetup luksRemoveKey /dev/sdc11
```

```
Enter passphrase to be deleted: fenestros123456789
```

```
WARNING!
```

```
=====
```

```
This is the last keyslot. Device will become unusable after purging this key.
```

```
Are you sure? (Type 'yes' in capital letters): YES
```

Supprimez maintenant LUKs :

```
root@debian11:~# cryptsetup remove /dev/mapper/sdc11
```

Vérifiez de nouveau le statut :

```
root@debian11:~# cryptsetup status sdc11
/dev/mapper/sdc11 is inactive.
```

```
root@debian11:~# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	32G	0	disk	
├─sda1	8:1	0	31G	0	part	/
├─sda2	8:2	0	1K	0	part	
└─sda5	8:5	0	975M	0	part	[SWAP]
sdb	8:16	0	64G	0	disk	
sdc	8:32	0	4G	0	disk	
├─sdc1	8:33	0	100M	0	part	
├─sdc2	8:34	0	100M	0	part	
├─sdc3	8:35	0	100M	0	part	
├─sdc4	8:36	0	1K	0	part	
├─sdc5	8:37	0	500M	0	part	
├─sdc6	8:38	0	200M	0	part	
└─┬vg0-lv1	254:0	0	104M	0	lvm	
├─sdc7	8:39	0	300M	0	part	
└─┬vg0-lv2	254:1	0	112M	0	lvm	

```
|—sdc8      8:40  0  500M  0 part
|  └md1     9:1   0  996M  0 raid5
|—sdc9      8:41  0  400M  0 part
|  └vg0-lv2 254:1  0  112M  0 lvm
|—sdc10     8:42  0  500M  0 part
|  └md1     9:1   0  996M  0 raid5
|—sdc11     8:43  0  500M  0 part
└sdc12     8:44  0  200M  0 part
sr0        11:0  1  378M  0 rom
```

Pour ajouter un autre disque à notre RAID afin de remplacer /dev/sdc5 il convient d'utiliser l'option **-add** :

```
root@debian11:~# mdadm --manage --add /dev/md1 /dev/sdc11
mdadm: added /dev/sdc11
```

L'examen du RAID indique que /dev/sdc11 a été ajouté en tant que *spare* et à l'issue de quelques secondes le RAID 5 a été reconstruite :

```
root@debian11:~# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Sun May  1 13:27:48 2022
    Raid Level : raid5
    Array Size : 1019904 (996.00 MiB 1044.38 MB)
  Used Dev Size : 509952 (498.00 MiB 522.19 MB)
    Raid Devices : 3
    Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Sun May  1 14:03:05 2022
      State : clean, degraded, recovering
  Active Devices : 2
 Working Devices : 3
  Failed Devices : 0
   Spare Devices : 1
```

```
Layout : left-symmetric
Chunk Size : 512K
```

```
Consistency Policy : resync
```

```
Rebuild Status : 56% complete
```

```
Name : debian11:1 (local to host debian11)
UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2
Events : 32
```

Number	Major	Minor	RaidDevice	State	
4	8	43	0	spare rebuilding	/dev/sdc11
1	8	40	1	active sync	/dev/sdc8
3	8	42	2	active sync	/dev/sdc10

```
root@debian11:~# mdadm --detail /dev/md1
/dev/md1:
```

```
Version : 1.2
Creation Time : Sun May 1 13:27:48 2022
Raid Level : raid5
Array Size : 1019904 (996.00 MiB 1044.38 MB)
Used Dev Size : 509952 (498.00 MiB 522.19 MB)
Raid Devices : 3
Total Devices : 3
Persistence : Superblock is persistent

Update Time : Sun May 1 14:03:07 2022
State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0
```

```
Layout : left-symmetric
Chunk Size : 512K
```

```
Consistency Policy : resync
```

```
Name : debian11:1 (local to host debian11)
UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2
Events : 40
```

Number	Major	Minor	RaidDevice	State	
4	8	43	0	active sync	/dev/sdc11
1	8	40	1	active sync	/dev/sdc8
3	8	42	2	active sync	/dev/sdc10

Vérifiez la prise en compte de la configuration :

```
root@debian11:~# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda              8:0    0   32G  0 disk
├─sda1           8:1    0   31G  0 part /
├─sda2           8:2    0    1K  0 part
└─sda5           8:5    0   975M  0 part [SWAP]
sdb              8:16   0   64G  0 disk
sdc              8:32   0    4G  0 disk
├─sdc1           8:33   0   100M  0 part
├─sdc2           8:34   0   100M  0 part
├─sdc3           8:35   0   100M  0 part
├─sdc4           8:36   0    1K  0 part
├─sdc5           8:37   0   500M  0 part
├─sdc6           8:38   0   200M  0 part
│   └─vg0-lv1    254:0   0   104M  0 lvm
├─sdc7           8:39   0   300M  0 part
│   └─vg0-lv2    254:1   0   112M  0 lvm
└─sdc8           8:40   0   500M  0 part
```

```
|  └md1      9:1    0  996M  0 raid5
| └sdc9      8:41   0  400M  0 part
|   └vg0-lv2 254:1   0  112M  0 lvm
| └sdc10     8:42   0  500M  0 part
|   └md1     9:1    0  996M  0 raid5
| └sdc11     8:43   0  500M  0 part
|   └md1     9:1    0  996M  0 raid5
└sdc12     8:44   0  200M  0 part
sr0        11:0    1  378M  0 rom
```

```
root@debian11:~# cat /proc/mdstatPersonalities : [raid6] [raid5] [raid4]
md1 : active raid5 sdc11[4] sdc10[3] sdc8[1]
      1019904 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
```

Dernièrement, il est nécessaire de renseigner le fichier **/etc/mdadm/mdadm.conf** du changement afin que le RAID soit construit à chaque démarrage :

```
root@debian11:~# echo 'DEVICE /dev/sdc11 /dev/sdc8 /dev/sdc10' > /etc/mdadm/mdadm.conf
root@debian11:~# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
root@debian11:~# cat /etc/mdadm/mdadm.conf
DEVICE /dev/sdc11 /dev/sdc8 /dev/sdc10
ARRAY /dev/md1 metadata=1.2 name=debian11:1 UUID=c0f945a0:f65b2136:b7913f8a:3707ffa2
```

Mettez à jour l'initramfs :

```
root@debian11:~# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-5.10.0-13-amd64
```

LAB #10 - autofs

```
[root@centos8 ~]# apt install autofs
```

```
[root@centos8 ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service →
/usr/lib/systemd/system/autofs.service.
```

```
[root@centos8 ~]# systemctl status autofs
● autofs.service - Automounts filesystems on demand
  Loaded: loaded (/usr/lib/systemd/system/autofs.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2022-10-11 09:26:39 EDT; 2min 36s ago
  Main PID: 67631 (automount)
    Tasks: 5 (limit: 100949)
   Memory: 1.8M
   CGroup: /system.slice/autofs.service
           └─67631 /usr/sbin/automount --systemd-service --dont-check-daemon
```

```
Oct 11 09:26:39 centos8.ittraining.loc systemd[1]: Starting Automounts filesystems on demand...
Oct 11 09:26:39 centos8.ittraining.loc systemd[1]: Started Automounts filesystems on demand.
```

```
[root@centos8 ~]# cat /etc/sysconfig/autofs
#
# Init system options
#
# If the kernel supports using the autofs miscellaneous device
# and you wish to use it you must set this configuration option
# to "yes" otherwise it will not be used.
#
USE_MISC_DEVICE="yes"
#
# Use OPTIONS to add automount(8) command line options that
# will be used when the daemon is started.
#
#OPTIONS=""
#
```

```
[root@centos8 ~]# vi /etc/sysconfig/autofs
```

```
[root@centos8 ~]# cat /etc/sysconfig/autofs
#
# Init system options
#
# If the kernel supports using the autofs miscellaneous device
# and you wish to use it you must set this configuration option
# to "yes" otherwise it will not be used.
#
USE_MISC_DEVICE="yes"
#
# Use OPTIONS to add automount(8) command line options that
# will be used when the daemon is started.
#
OPTIONS="--timeout=600"
#
```

```
[root@centos8 ~]# systemctl restart autofs
[root@centos8 ~]# systemctl status autofs
● autofs.service - Automounts filesystems on demand
   Loaded: loaded (/usr/lib/systemd/system/autofs.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2022-10-11 09:32:06 EDT; 3s ago
     Main PID: 67756 (automount)
        Tasks: 5 (limit: 100949)
       Memory: 1.7M
      CGroup: /system.slice/autofs.service
              └─67756 /usr/sbin/automount --timeout=600 --systemd-service --dont-check-daemon

Oct 11 09:32:05 centos8.ittraining.loc systemd[1]: Starting Automounts filesystems on demand...
Oct 11 09:32:06 centos8.ittraining.loc systemd[1]: Started Automounts filesystems on demand.
```

```
[root@centos8 ~]# cat /etc/auto.master
#
# Sample auto.master file
# This is a 'master' automounter map and it has the following format:
```

```
# mount-point [map-type[,format]:]map [options]
# For details of the format look at auto.master(5).
#
/misc    /etc/auto.misc
#
# NOTE: mounts done from a hosts map will be mounted with the
#       "nosuid" and "nodev" options unless the "suid" and "dev"
#       options are explicitly given.
#
/net     -hosts
#
# Include /etc/auto.master.d/*.autofs
# The included files must conform to the format of this file.
#
+dir:/etc/auto.master.d
#
# If you have fedfs set up and the related binaries, either
# built as part of autofs or installed from another package,
# uncomment this line to use the fedfs program map to access
# your fedfs mounts.
#/nfs4   /usr/sbin/fedfs-map-nfs4 nobind
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master
```

```
[root@centos8 ~]# cat /etc/auto.misc
#
```

```
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

cd                -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

# the following entries are samples to pique your imagination
#linux            -ro,soft                ftp.example.org:/pub/linux
#boot             -fstype=ext2              :/dev/hda1
#floppy           -fstype=auto             :/dev/fd0
#floppy           -fstype=ext2             :/dev/fd0
#e2floppy         -fstype=ext2             :/dev/fd0
#jaz              -fstype=ext2             :/dev/sdc1
#removable        -fstype=ext2             :/dev/hdd
```