

Version : **2024.01**

Dernière mise-à-jour : 2024/10/24 12:57

RH13405 - Gestion des Disques, des Systèmes de Fichiers et du Swap

Contenu du Module

- **RH13405 - Gestion des Disques, des Systèmes de Fichiers et du Swap**

- Contenu du Module
- Périphériques de stockage
- Partitions
- Partitionnement
 - LAB #1 - Partitionnement de votre Disque avec fdisk
 - LAB #2 - Modifier les Drapeaux des Partitions avec fdisk
- Logical Volume Manager (LVM)
 - LAB #3 - Volumes Logiques Linéaires
 - Physical Volume (PV)
 - Volume Group (VG) et Physical Extent (PE)
 - Logical Volumes (LV)
 - LAB #4 - Étendre un Volume Logique à Chaud
 - LAB #5 - Snapshots
 - LAB #6 - Suppression des Volumes
 - LAB #7 - Volumes Logiques en Miroir
 - LAB #8 - Modifier les Attributs LVM
 - LAB #9 - Volumes Logiques en Bandes
 - LAB #10 - 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 #11 - Mise en Place du RAID 5 Logiciel
 - 11.1 - Préparer le disque
 - 11.2 - Créer une Unité RAID
 - 11.3 - Remplacer une Unité Défaillante
- LAB #12 - Travailler avec Stratis
 - 12.1 - Présentation
 - Concepts clés de Stratis
 - Gestion simplifiée
 - Avantages de Stratis
 - Stratis vs ZFS/Btrfs
 - 12.2 - Travailler avec Stratis
 - Installation
 - Création d'un Pool
 - Création d'un Système de Fichiers
 - Montage du System de Fichiers
 - Croissance Dynamique du Système de Fichiers
 - Création d'un Snapshot
 - Suppression des Systèmes de Fichiers
- Systèmes de Fichiers Journalisés
 - Présentation
 - LAB #13 - Ext3
 - 13.1 - Gestion d'Ext3
 - 13.2 - Convertir un Système de Fichiers Ext3 en Ext2

- 13.3 - Convertir un Système de Fichiers Ext2 en Ext3
 - 13.4 - Placer le Journal sur un autre Partition
 - 13.5 - Modifier la Fréquence de Vérification du Système de Fichiers Ext3
- LAB #14 - Ext4
 - 14.1 - Créer un Système de Fichiers Ext4
 - 14.2 - Ajouter une Étiquette au Système de Fichiers Ext4
 - 14.3 - Convertir un Système de Fichiers Ext3 en Ext4
- LAB #15 - XFS
 - 15.1 - Créer un Système de Fichiers XFS
 - 15.2 - Ajouter une Étiquette au Système de Fichiers XFS
- Autres Systèmes de Fichiers
 - ReiserFS
 - JFS
 - Btrfs
- Comparaison des Commandes par Système de Fichiers
- LAB #16 - Système de Fichiers ISO
 - 16.1 - La Commande mkisofs
- LAB #17 - Systèmes de Fichiers Chiffrés
 - 17.1 - Créer un Système de Fichiers Chiffré avec LUKS
 - Présentation
 - Mise en Place
 - Ajouter une deuxième Passphrase
 - Supprimer une Passphrase
 - Supprimer LUKS
- LAB #18 - Le Swap
 - 18.1 - Taille du swap
 - 18.2 - Partitions de swap
 - 18.3 - La Commande swapon
 - 18.4 - La Commande swapoff
 - 18.5 - Créer un Fichier de Swap

Périphériques de stockage

Les unités de stockage sous Linux sont référencées par un des fichiers se trouvant dans le répertoire **/dev** :

- **hd[a-d]**
 - Les disques IDE et les lecteurs ATAPI
- **sd[a-z]**
 - Les disques SCSI et SATA
- **mmcblk[0-7]**
 - Les cartes SD/MMC
- **scd[0-7]**
 - Les CDRoms SCSI
- **xd[a-d]**
 - Les premiers disques sur IBM XT
- **fd[0-7]**
 - Les lecteurs de disquettes
- **st[0-7]**
 - Les lecteurs de bandes SCSI/SATA qui **supportent** le rembobinage
- **nst[0-7]**
 - Les lecteurs de bandes SCSI/SATA qui ne supportent **pas** le rembobinage
- **ht[0-7]**
 - Les lecteurs de bandes PATA qui **supportent** le rembobinage
- **nht[0-7]**
 - Les lecteurs de bandes PATA qui ne supportent **pas** le rembobinage
- **rmt8, rmt16, tape-d, tape-reset**
 - Les lecteurs QIC-80
- **ram[0-15]**
 - Les disques virtuels. Ils sont supprimés à l'extinction de la machine. Un de ces disques est utilisé par le système pour monter l'image d'un disque racine défini par le fichier **initrd** au démarrage de la machine
- Périphériques **loop**
 - Il existe 16 unités loop qui sont utilisés pour accéder en mode bloc à un système de fichiers contenu dans un fichier, par exemple, une image **iso**
- **md[x]**

- Un volume **RAID** logiciel
- vg[x]
 - Un groupe de volumes
- lv[x]
 - Un volume logique

Partitions

Un PC comportent en règle générale 2 **contrôleurs** de disque, chacun capable de gérer 2 disques, un **maître** et un **esclave**. Les disques attachés à ces contrôleurs comportent des noms différents pour pouvoir les distinguer :

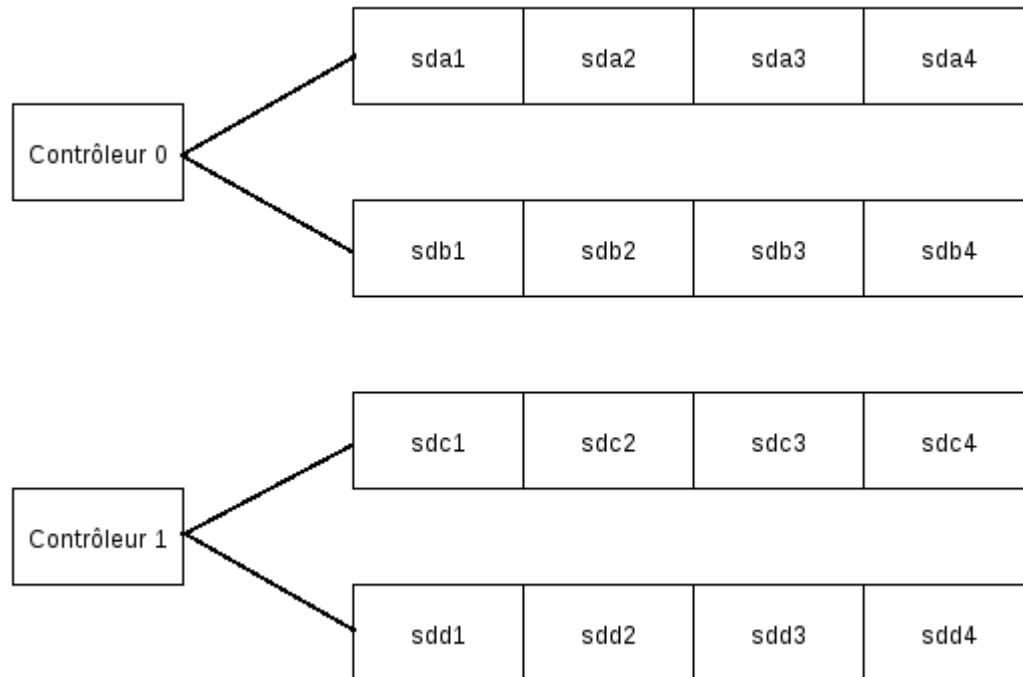
- Contrôleur 0
 - Maître
 - **hda** - disque IDE
 - **sda** - disque SATA ou SCSI
 - Esclave
 - **hdb** - disque IDE
 - **sdb** - disque SATA ou SCSI
- Contrôleur 1
 - Maître
 - **hdc** - disque IDE
 - **sd**c - disque SATA ou SCSI
 - Esclave
 - **hdd** - disque IDE
 - **sdd** - disque SATA ou SCSI

Un disque peut comporter trois types de partitions :

- **Partitions primaires**,
 - Maximum de **4**. En effet, la Table des Partitions est grande de 64 octets. Il faut 16 octets pour coder une partition.
- **Partitions Étendues**,
 - Généralement une seule partition étendue par disque. Elle contient des **Lecteurs Logiques** aussi appelés des partitions,
- **Lecteurs Logiques**.

Les 4 partitions primaires sont numérotées de 1 à 4. Par exemple :

- **hda1**, **hda2**, **hda3** et **hda4** pour le premier disque **IDE** sur le premier contrôleur de disque,
- **sda1**, **sda2**, **sda3** et **sda4** pour le premier disque **SCSI** ou **SATA** sur le premier contrôleur de disque.



Une partition étendue prend la place d'une partition primaire et les lecteurs logiques qui s'y trouvent commencent à partir de **hda5** ou de **sda5**.

Pour clarifier ceci, considérons un disque **SATA** contenant deux partitions primaires, une seule partition étendue et 3 lecteurs logiques. Dans ce cas, les deux premières partitions sont **sda1** et **sda2**, la partition étendue prend la place de la troisième partition primaire, la **sda3** et s'appelle ainsi tandis que la quatrième partition primaire est inexistante.

Les lecteurs logiques commençant à **sda5**, nous obtenons la liste de partitions suivante : sda1, sda2, sda5, sda6, sda7. Notez que la sda3 ne peut pas être utilisée en tant que partition car elle est cachée par les lecteurs sda5, sda6 et sda7.



Le nombre de partitions sur un disque est limité :

- **IDE,**
 - Jusqu'à **63**,
- **SCSI,**
 - Jusqu'à **15**,
- **Disques utilisant l'API libata,**
 - Jusqu'à **15**.



Important : Ces limites peuvent être dépassées en utilisant la gestion **LVM** (*Logical Volume Management*).

Partitionnement

Le schéma de partitionnement **Master Boot Record** (MBR) est la norme sur les systèmes qui utilisent le BIOS. Ce schéma prend en charge un maximum de quatre partitions primaires. Sur les systèmes Linux, avec des partitions étendues et logiques, vous pouvez créer jusqu'à 15 partitions. Avec des systèmes 32 bits, les disques partitionnés avec MBR peuvent avoir une taille allant jusqu'à 2 TiB.

La limite de 2 TiB pour la taille des disques et des partitions est désormais une limite courante et restrictive. Par conséquent, le schéma MBR est remplacé par le système de partitionnement **GUID Partition Table** (GPT).

Pour les systèmes qui utilisent le micrologiciel UEFI (Unified Extensible Firmware Interface), GPT est la norme pour le partitionnement des disques et répond aux limites du schéma MBR. GPT offre un maximum de 128 partitions au maximum. Le schéma GPT alloue 64 bits pour les adresses de blocs logiques, afin de prendre en charge des partitions et des disques allant jusqu'à huit zébibytes (ZiB) ou huit milliards de tébibytes (TiB).

Le partitionnement GPT offre des fonctionnalités et des avantages supplémentaires par rapport au MBR. GPT utilise un identifiant unique (GUID) pour identifier chaque disque et chaque partition. GPT rend la table de partition redondante avec le GPT primaire en tête de disque et un GPT secondaire de sauvegarde à la fin du disque. GPT utilise une somme de contrôle pour détecter les erreurs dans l'en-tête GPT et la table de partition.

LAB #1 - Partitionnement de votre Disque sous RedHat 9 avec fdisk

Pour procéder au partitionnement de votre disque ou de vos disques, RedHat 9 possède l'outil dénommé **fdisk**.

Lancez fdisk en fournissant en argument le fichier de référence de votre premier disque dur, par exemple :

```
[root@redhat9 ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
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.
Created a new DOS disklabel with disk identifier 0x23a56c2e.

Command (m for help):
```

Tapez ensuite la lettre **m** puis pour obtenir le menu :

```
Command (m for help): m

Help:

DOS (MBR)
  a  toggle a bootable flag
  b  edit nested BSD disklabel
  c  toggle the dos compatibility flag

Generic
```

```
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
t  change a partition type
v  verify the partition table
i  print information about a partition
```

Misc

```
m  print this menu
u  change display/entry units
x  extra functionality (experts only)
```

Script

```
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file
```

Save & Exit

```
w  write table to disk and exit
q  quit without saving changes
```

Create a new label

```
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty DOS partition table
s  create a new empty Sun partition table
```

Command (m for help):

Pour créer une nouvelle partition, vous devez utiliser la commande **n**.

Créez donc les partitions suivantes sur votre disque :

Partition	Type	Taille de la Partition
/dev/sdb1	Primaire	100 Mo
/dev/sdb2	Primaire	100 Mo
/dev/sdb3	Primaire	100 Mo
/dev/sdb4	Extended	Du premier secteur disponible au dernier secteur du disque
/dev/sdb5	Logique	500 Mo
/dev/sdb6	Logique	200 Mo
/dev/sdb7	Logique	300 Mo
/dev/sda8	Logique	500 Mo
/dev/sdb9	Logique	400 Mo
/dev/sdb10	Logique	500 Mo
/dev/sdb11	Logique	500 Mo
/dev/sdb12	Logique	200 Mo

Créez d'abord les partitions primaires :

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-67108863, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-67108863, default 67108863): +100M

Created a new partition 1 of type 'Linux' and of size 100 MiB.

Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
```

```
Select (default p):
```

```
Using default response p.
```

```
Partition number (2-4, default 2):
```

```
First sector (206848-67108863, default 206848):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (206848-67108863, default 67108863): +100M
```

```
Created a new partition 2 of type 'Linux' and of size 100 MiB.
```

```
Command (m for help): n
```

```
Partition type
```

```
    p   primary (2 primary, 0 extended, 2 free)
```

```
    e   extended (container for logical partitions)
```

```
Select (default p):
```

```
Using default response p.
```

```
Partition number (3,4, default 3):
```

```
First sector (411648-67108863, default 411648):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (411648-67108863, default 67108863): +100M
```

```
Created a new partition 3 of type 'Linux' and of size 100 MiB.
```

```
Command (m for help):
```

Créez ensuite la partition étendue :

```
Command (m for help): n
```

```
Partition type
```

```
    p   primary (3 primary, 0 extended, 1 free)
```

```
    e   extended (container for logical partitions)
```

```
Select (default e):
```

```
Using default response e.
```

```
Selected partition 4
```

```
First sector (616448-67108863, default 616448):  
Last sector, +/-sectors or +/-size{K,M,G,T,P} (616448-67108863, default 67108863):  
  
Created a new partition 4 of type 'Extended' and of size 31.7 GiB.  
  
Command (m for help):
```

Créez maintenant les autres partitions l'une après l'autre :

```
Command (m for help): n  
All primary partitions are in use.  
Adding logical partition 5  
First sector (618496-67108863, default 618496):  
Last sector, +/-sectors or +/-size{K,M,G,T,P} (618496-67108863, default 67108863): +500M  
  
Created a new partition 5 of type 'Linux' and of size 500 MiB.  
  
Command (m for help): n  
All primary partitions are in use.  
Adding logical partition 6  
First sector (1644544-67108863, default 1644544):  
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1644544-67108863, default 67108863): +200M  
  
Created a new partition 6 of type 'Linux' and of size 200 MiB.  
  
Command (m for help): n  
All primary partitions are in use.  
Adding logical partition 7  
First sector (2056192-67108863, default 2056192):  
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2056192-67108863, default 67108863): +300M  
  
Created a new partition 7 of type 'Linux' and of size 300 MiB.  
  
Command (m for help): n
```

All primary partitions are in use.

Adding logical partition 8

First sector (2672640-67108863, default 2672640):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2672640-67108863, default 67108863): +500M

Created a new partition 8 of type 'Linux' and of size 500 MiB.

Command (m for help): n

All primary partitions are in use.

Adding logical partition 9

First sector (3698688-67108863, default 3698688):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (3698688-67108863, default 67108863): +400M

Created a new partition 9 of type 'Linux' and of size 400 MiB.

Command (m for help): n

All primary partitions are in use.

Adding logical partition 10

First sector (4519936-67108863, default 4519936):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (4519936-67108863, default 67108863): +500M

Created a new partition 10 of type 'Linux' and of size 500 MiB.

Command (m for help): n

All primary partitions are in use.

Adding logical partition 11

First sector (5545984-67108863, default 5545984):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (5545984-67108863, default 67108863): +500M

Created a new partition 11 of type 'Linux' and of size 500 MiB.

Command (m for help): n

All primary partitions are in use.

Adding logical partition 12

```
First sector (6572032-67108863, default 6572032):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (6572032-67108863, default 67108863): +200M

Created a new partition 12 of type 'Linux' and of size 200 MiB.

Command (m for help):
```

Tapez ensuite la lettre **p** puis pour visualiser la nouvelle table des partitions. Vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): p
Disk /dev/sdb: 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: 0x23a56c2e
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended
/dev/sdb5		618496	1642495	1024000	500M	83	Linux
/dev/sdb6		1644544	2054143	409600	200M	83	Linux
/dev/sdb7		2056192	2670591	614400	300M	83	Linux
/dev/sdb8		2672640	3696639	1024000	500M	83	Linux
/dev/sdb9		3698688	4517887	819200	400M	83	Linux
/dev/sdb10		4519936	5543935	1024000	500M	83	Linux
/dev/sdb11		5545984	6569983	1024000	500M	83	Linux
/dev/sdb12		6572032	6981631	409600	200M	83	Linux

```
Command (m for help):
```

Ecrivez la table des partitions sur disque et exécutez la commande **partprobe** :

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
[root@redhat9 ~]# partprobe
[root@redhat9 ~]#
```

Lancez fdisk puis tapez ensuite la lettre **p** puis pour visualiser la table des partitions actuelle :

```
[root@redhat9 ~]# fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help): p
Disk /dev/sdb: 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: 0x23a56c2e
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended
/dev/sdb5		618496	1642495	1024000	500M	83	Linux

/dev/sdb6	1644544	2054143	409600	200M	83	Linux
/dev/sdb7	2056192	2670591	614400	300M	83	Linux
/dev/sdb8	2672640	3696639	1024000	500M	83	Linux
/dev/sdb9	3698688	4517887	819200	400M	83	Linux
/dev/sdb10	4519936	5543935	1024000	500M	83	Linux
/dev/sdb11	5545984	6569983	1024000	500M	83	Linux
/dev/sdb12	6572032	6981631	409600	200M	83	Linux

Command (m for help):

Pour supprimer une partition, utilisez la commande **d** puis ↵ Entrée. fdisk vous demandera le numéro de la partition à supprimer, par exemple :

Command (m for help): d

Partition number (1-12, default 12): 12

Partition 12 has been deleted.

Command (m for help): p

Disk /dev/sdb: 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: 0x23a56c2e

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended
/dev/sdb5		618496	1642495	1024000	500M	83	Linux
/dev/sdb6		1644544	2054143	409600	200M	83	Linux
/dev/sdb7		2056192	2670591	614400	300M	83	Linux

```
/dev/sdb8      2672640  3696639  1024000  500M 83 Linux
/dev/sdb9      3698688  4517887   819200  400M 83 Linux
/dev/sdb10     4519936  5543935  1024000  500M 83 Linux
/dev/sdb11     5545984  6569983  1024000  500M 83 Linux
```

Command (m for help):

A ce stade, la partition n'a **pas** été réellement supprimée. En effet, vous avez la possibilité de sortir de fdisk en utilisant la commande **q**.

Tapez donc q pour sortir de fdisk puis relancez fdisk. Vous obtiendrez un résultat similaire à celui-ci :

Command (m for help): q

```
[root@redhat9 ~]# fdisk /dev/sdb
```

Welcome to fdisk (util-linux 2.37.4).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): p

Disk /dev/sdb: 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: 0x23a56c2e

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended

```
/dev/sdb5      618496 1642495 1024000 500M 83 Linux
/dev/sdb6      1644544 2054143 409600 200M 83 Linux
/dev/sdb7      2056192 2670591 614400 300M 83 Linux
/dev/sdb8      2672640 3696639 1024000 500M 83 Linux
/dev/sdb9      3698688 4517887 819200 400M 83 Linux
/dev/sdb10     4519936 5543935 1024000 500M 83 Linux
/dev/sdb11     5545984 6569983 1024000 500M 83 Linux
/dev/sdb12     6572032 6981631 409600 200M 83 Linux
```

Command (m for help):

LAB #2 - Modifier les Drapeaux des Partitions avec fdisk

Afin de mettre en place un RAID logiciel ou un volume logique, il est nécessaire de modifier les types de systèmes de fichiers sur les partitions créées.

Modifiez donc les nouvelles partitions à l'aide de la commande **t** de **fdisk** selon le tableau ci-dessous :

Taille de la Partition	Type de Système de Fichiers
500 Mo	RAID (fd)
200 Mo	Linux LVM (8e)
300 Mo	Linux LVM (8e)
500 Mo	RAID (fd)
400 Mo	Linux LVM (8e)
500 Mo	RAID (fd)
500 Mo	RAID (fd)
200 Mo	Inchangé

Vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): t
Partition number (1-12, default 12): 5
Hex code or alias (type L to list all): fd
```

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): t

Partition number (1-12, default 12): 6

Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t

Partition number (1-12, default 12): 7

Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t

Partition number (1-12, default 12): 8

Hex code or alias (type L to list all): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): t

Partition number (1-12, default 12): 9

Hex code or alias (type L to list all): 8e

Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): t

Partition number (1-12, default 12): 10

Hex code or alias (type L to list all): fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): t

Partition number (1-12, default 12): 11

```
Hex code or alias (type L to list all): fd
```

```
Changed type of partition 'Linux' to 'Linux raid autodetect'.
```

```
Command (m for help):
```

A l'issu des modifications, vous obtiendrez un résultat similaire à celui-ci :

```
Command (m for help): p
```

```
Disk /dev/sdb: 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: 0x23a56c2e
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended
/dev/sdb5		618496	1642495	1024000	500M	fd	Linux raid autodetect
/dev/sdb6		1644544	2054143	409600	200M	8e	Linux LVM
/dev/sdb7		2056192	2670591	614400	300M	8e	Linux LVM
/dev/sdb8		2672640	3696639	1024000	500M	fd	Linux raid autodetect
/dev/sdb9		3698688	4517887	819200	400M	8e	Linux LVM
/dev/sdb10		4519936	5543935	1024000	500M	fd	Linux raid autodetect
/dev/sdb11		5545984	6569983	1024000	500M	fd	Linux raid autodetect
/dev/sdb12		6572032	6981631	409600	200M	83	Linux

```
Command (m for help):
```

Pour écrire la nouvelle table des partitions sur disque, vous devez utiliser la commande **w** puis la commande **partprobe** :


```
-l, --list                display partitions and exit
-x, --list-details        like --list but with more details
-n, --noauto-pt           don't create default partition table on empty devices
-o, --output <list>       output columns
-t, --type <type>         recognize specified partition table type only
-u, --units[=<unit>]      display units: 'cylinders' or 'sectors' (default)
-s, --getsz               display device size in 512-byte sectors [DEPRECATED]
    --bytes               print SIZE in bytes rather than in human readable format
    --lock[=<mode>]       use exclusive device lock (yes, no or nonblock)
-w, --wipe <mode>         wipe signatures (auto, always or never)
-W, --wipe-partitions <mode> wipe signatures from new partitions (auto, always or never)

-C, --cylinders <number>  specify the number of cylinders
-H, --heads <number>      specify the number of heads
-S, --sectors <number>    specify the number of sectors per track

-h, --help                display this help
-V, --version              display version
```

Available output columns:

```
gpt: Device Start End Sectors Size Type Type-UUID Attrs Name UUID
dos: Device Start End Sectors Cylinders Size Type Id Attrs Boot End-C/H/S Start-C/H/S
bsd: Slice Start End Sectors Cylinders Size Type Bsize Cpg Fsize
sgi: Device Start End Sectors Cylinders Size Type Id Attrs
sun: Device Start End Sectors Cylinders Size Type Id Flags
```

For more details see `fdisk(8)`.

Logical Volume Manager (LVM)

LAB #3 - Volumes Logiques Linéaires

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

Nous allons travailler sous RedHat 9 avec les partitions suivantes :

/dev/sdb6	1644544	2054143	409600	200M	8e	Linux	LVM
/dev/sdb7	2056192	2670591	614400	300M	8e	Linux	LVM
/dev/sdb9	3698688	4517887	819200	400M	8e	Linux	LVM

Pour initialiser le LVM saisissez la commande suivante :

```
[root@redhat9 ~]# vgscan
Found volume group "rhel" using metadata type lvm2
```

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

```
[root@redhat9 ~]# vgscan --longhelp
vgscan - Search for all volume groups

vgscan
[ --ignorelockingfailure ]
[ --mknodes ]
[ --notifydbus ]
[ --reportformat basic|json|json_std ]
[ 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 ]
[   --devicesfile String ]
[   --devices PV ]
[   --nohints ]
[   --journal String ]
```

Common variables for lvm:

Variables in option or position args are capitalized,
e.g. PV, VG, LV, Size, Number, String, Tag.

PV

Physical Volume name, a device path under /dev.

For commands managing physical extents, a PV positional arg generally accepts a suffix indicating a range (or multiple ranges) of PEs. When the first PE is omitted, it defaults to the start of the device, and when the last PE is omitted it defaults to the end.

PV[:PE-PE]... is start and end range (inclusive),

PV[:PE+PE]... is start and length range (counting from 0).

LV

Logical Volume name. See lvm(8) for valid names. An LV positional arg generally includes the VG name and LV name, e.g. VG/LV.

LV followed by _<type> indicates that an LV of the given type is required. (raid represents raid<N> type).

The _new suffix indicates that the LV name is new.

Tag

Tag name. See `lvm(8)` for information about tag names and using tags in place of a VG, LV or PV.

Select

Select indicates that a required positional arg can be omitted if the `--select` option is used. No arg appears in this position.

Size[UNIT]

Size is an input number that accepts an optional unit.

Input units are always treated as base two values, regardless of capitalization, e.g. 'k' and 'K' both refer to 1024.

The default input unit is specified by letter, followed by |UNIT.

UNIT represents other possible input units: BbBsSkKmMgGtTpPeE.

(This should not be confused with the output control `--units`, where capital letters mean multiple of 1000.)

Physical Volume (PV)

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

```
[root@redhat9 ~]# pvcreate /dev/sdb6 /dev/sdb7 /dev/sdb9
Physical volume "/dev/sdb6" successfully created.
Physical volume "/dev/sdb7" successfully created.
Physical volume "/dev/sdb9" successfully created.
```

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

```
[root@redhat9 ~]# pvcreate --longhelp
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 ]
[   --pvmetadataacopies 0|1|2 ]
[   --metadatasize Size[m|UNIT] ]
[   --metadataignore y|n ]
[   --norestorefile ]
[   --setphysicalvolumesize Size[m|UNIT] ]
[   --reportformat basic|json|json_std ]
[   --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 ]
[   --devicesfile String ]
[   --devices PV ]
```

```
[ --nohints ]  
[ --journal String ]
```

Common variables for lvm:

Variables in option or position args are capitalized,
e.g. PV, VG, LV, Size, Number, String, Tag.

PV

Physical Volume name, a device path under /dev.

For commands managing physical extents, a PV positional arg generally accepts a suffix indicating a range (or multiple ranges) of PEs. When the first PE is omitted, it defaults to the start of the device, and when the last PE is omitted it defaults to the end. PV[:PE-PE]... is start and end range (inclusive), PV[:PE+PE]... is start and length range (counting from 0).

LV

Logical Volume name. See lvm(8) for valid names. An LV positional arg generally includes the VG name and LV name, e.g. VG/LV.

LV followed by _<type> indicates that an LV of the given type is required. (raid represents raid<N> type).

The _new suffix indicates that the LV name is new.

Tag

Tag name. See lvm(8) for information about tag names and using tags in place of a VG, LV or PV.

Select

Select indicates that a required positional arg can be omitted if the --select option is used. No arg appears in this position.

Size[UNIT]

Size is an input number that accepts an optional unit.

Input units are always treated as base two values, regardless of

capitalization, e.g. 'k' and 'K' both refer to 1024.
The default input unit is specified by letter, followed by |UNIT.
UNIT represents other possible input units: BbBsSkKmMgGtTpPeE.
(This should not be confused with the output control --units, where
capital letters mean multiple of 1000.)

Pour visualiser les PVs il convient d'utiliser la commande **pvdisplay** :

```
[root@redhat9 ~]# pvdisplay /dev/sdb6 /dev/sdb7 /dev/sdb9
"/dev/sdb6" is a new physical volume of "200.00 MiB"
--- NEW Physical volume ---
PV Name                /dev/sdb6
VG Name
PV Size                200.00 MiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                qFjsnE-XjSN-HBqU-z4ML-NnJm-E6EQ-6hNFUo
"/dev/sdb7" is a new physical volume of "300.00 MiB"
--- NEW Physical volume ---
PV Name                /dev/sdb7
VG Name
PV Size                300.00 MiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                CKljsJ-dZ1n-WwQZ-G0w3-3E7k-juL1-rd80hF
"/dev/sdb9" is a new physical volume of "400.00 MiB"
--- NEW Physical volume ---
PV Name                /dev/sdb9
```

VG Name	
PV Size	400.00 MiB
Allocatable	NO
PE Size	0
Total PE	0
Free PE	0
Allocated PE	0
PV UUID	80N1XI-aoN5-Vcnp-A0aP-eSWF-cWIH-QMbe0A

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

```
[root@redhat9 ~]# pvdisk --longhelp
pvdisk - Display various attributes of physical volume(s)
```

```
pvdisk
  [ -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|json_std ]
  [ --separator String ]
```

```
[ --shared ]  
[ --unbuffered ]  
[ --units [Number]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 ]  
[ --devicesfile String ]  
[ --devices PV ]  
[ --nohints ]  
[ --journal String ]
```

Common variables for lvm:

Variables in option or position args are capitalized,
e.g. PV, VG, LV, Size, Number, String, Tag.

PV

Physical Volume name, a device path under /dev.

For commands managing physical extents, a PV positional arg
generally accepts a suffix indicating a range (or multiple ranges)

of PEs. When the first PE is omitted, it defaults to the start of the device, and when the last PE is omitted it defaults to the end. PV[:PE-PE]... is start and end range (inclusive), PV[:PE+PE]... is start and length range (counting from 0).

LV

Logical Volume name. See lvm(8) for valid names. An LV positional arg generally includes the VG name and LV name, e.g. VG/LV.

LV followed by _<type> indicates that an LV of the given type is required. (raid represents raid<N> type).

The _new suffix indicates that the LV name is new.

Tag

Tag name. See lvm(8) for information about tag names and using tags in place of a VG, LV or PV.

Select

Select indicates that a required positional arg can be omitted if the --select option is used. No arg appears in this position.

Size[UNIT]

Size is an input number that accepts an optional unit.

Input units are always treated as base two values, regardless of capitalization, e.g. 'k' and 'K' both refer to 1024.

The default input unit is specified by letter, followed by |UNIT.

UNIT represents other possible input units: BbBsSkKmMgGtTpPeE.

(This should not be confused with the output control --units, where capital letters mean multiple of 1000.)

Volume Group (VG) et Physical Extent (PE)

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

```
[root@redhat9 ~]# vgcreate -s 8M vg0 /dev/sdb6 /dev/sdb7 /dev/sdb9
Volume group "vg0" successfully created
```

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

```
[root@redhat9 ~]# 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] ]
  [ --pvmetadataacopies 0|1|2 ]
  [ --vgmetadataacopies all|unmanaged|Number ]
  [ --reportformat basic|json|json_std ]
  [ --dataalignment Size[k|UNIT] ]
  [ --dataalignmentoffset Size[k|UNIT] ]
  [ --shared ]
  [ --systemid String ]
  [ --locktype sanlock|dlm|none ]
  [ --setautoactivation y|n ]
  [ 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 ]
[ --devicesfile String ]
[ --devices PV ]
[ --nohints ]
[ --journal String ]
```

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

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

```
[root@redhat9 ~]# 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	7tQDuL-D900-dWk5-AEo1-J0dP-9Tdq-BKfiBF

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

```
[root@redhat9 ~]# 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 ]
  [ --shared ]
```

```
[ --separator String ]  
[ --unbuffered ]  
[ --units [Number]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 ]  
[ --devicesfile String ]  
[ --devices PV ]  
[ --nohints ]  
[ --journal String ]
```

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@redhat9 ~]# lvcreate -L 350 -n lv0 vg0
```

Rounding up size to full physical extent 352.00 MiB
Logical volume "lv0" created.



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

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

```
[root@redhat9 ~]# lvcreate --help
lvcreate - Create a logical volume
```

Create a linear LV.

```
lvcreate -L|--size Size[m|UNIT] VG
[ --type linear ] (implied)
[ -l|--extents Number[PERCENT] ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a striped LV.

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

Create a raid1 or mirror LV.

```
lvcreate -m|--mirrors Number -L|--size Size[m|UNIT] VG
[ --type raid1|mirror ] (implied)
[ -l|--extents Number[PERCENT] ]
[ -I|--stripesize Size[k|UNIT] ]
[ -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] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -m|--mirrors Number ]
[ -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
[ --type raid10 ] (implied)
[ -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
```

```
[ --type snapshot ] (implied)
[ -l|--extents Number[PERCENT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -c|--chunksize Size[k|UNIT] ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a thin pool.

```
lvcreate --type thin-pool -L|--size Size[m|UNIT] VG
[ -l|--extents Number[PERCENT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -T|--thin ]
[ -c|--chunksize Size[k|UNIT] ]
[ --thinpool LV_new ]
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a cache pool.

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

```
[ --poolmetadatasize Size[m|UNIT] ]  
[ --poolmetadataspare y|n ]  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

Create a thin LV in a thin pool.

```
lvcreate -V|--virtualsize Size[m|UNIT] --thinpool LV VG  
[ --type thin ] (implied)  
[ -T|--thin ]  
[ COMMON_OPTIONS ]
```

Create a thin LV that is a snapshot of an existing thin LV.

```
lvcreate -s|--snapshot LV  
[ --type thin ] (implied)  
[ -T|--thin ]  
[ COMMON_OPTIONS ]
```

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

```
lvcreate --type thin --thinpool LV LV  
[ -T|--thin ]  
[ COMMON_OPTIONS ]
```

Create a LV that returns VDO when used.

```
lvcreate --type vdo -L|--size Size[m|UNIT] VG  
[ -l|--extents Number[PERCENT] ]  
[ -i|--stripes Number ]  
[ -I|--stripesize Size[k|UNIT] ]  
[ -V|--virtualsize Size[m|UNIT] ]  
[ --vdo ]  
[ --vdopool LV_new ]  
[ --compression y|n ]  
[ --deduplication y|n ]  
[ --vdosettings String ]  
[ 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 VG
[ -l|--extents Number[PERCENT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -H|--cache ]
[ -c|--chunksize Size[k|UNIT] ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
[ --cachesettings String ]
[ --cachemetadadataformat auto|1|2 ]
[ --poolmetadatasize Size[m|UNIT] ]
[ --poolmetadataspare y|n ]
[ 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] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -c|--chunksize 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] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -c|--chunksize Size[k|UNIT] ]
[ --cachesize Size[m|UNIT] ]
[ --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 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 ]  
[ --setautoactivation y|n ]  
[ --ignoremonitoring ]  
[ --metadataprofile String ]  
[ --minor Number ]  
[ --monitor y|n ]  
[ --nosync ]  
[ --noudevsync ]  
[ --reportformat basic|json|json_std ]
```

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 ]  
[ --devicesfile String ]  
[ --devices PV ]  
[ --nohints ]  
[ --journal String ]
```

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

Créez maintenant un répertoire dans /mnt pour monter lv0 :

```
[root@redhat9 ~]# mkdir /mnt/lvm
```

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

```
[root@redhat9 ~]# mke2fs -j /dev/vg0/lv0  
mke2fs 1.46.5 (30-Dec-2021)  
Discarding device blocks: done  
Creating filesystem with 360448 1k blocks and 90112 inodes  
Filesystem UUID: 34d451f7-96c2-43b1-8cc0-9fe8bafcf498  
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@redhat9 ~]# mount -t ext3 /dev/vg0/lv0 /mnt/lvm
```

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

Saisissez donc la commande pour copier le contenu de /home :

```
[root@redhat9 ~]# cp -a /home /mnt/lvm
```

Constatez ensuite le contenu de /mnt/lvm :

```
[root@redhat9 ~]# ls -l /mnt/lvm
total 13
drwxr-xr-x. 3 root root 1024 Oct 19 2023 home
drwx----- 2 root root 12288 Oct 23 15:21 lost+found
[root@redhat9 ~]#
```

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@redhat9 ~]# df -h /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 320M  4.4M  298M   2% /mnt/lvm
```

Dans la cas de notre exemple, la taille est de **320 Mo** avec **4.4 Mo** occupé.

LAB #4 - Etendre un Volume Logique à Chaud

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

```
[root@redhat9 ~]# 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@redhat9 ~]# 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] ]
    [ --fs String ]
    [ --fsmode String ]
    [ COMMON_OPTIONS ]
    [ PV ... ]

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

Extend a pool metadata SubLV by a specified size.

```
lvextend --poolmetadatasize [+]Size[m|UNIT] LV
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Extend an LV according to a predefined policy.

```
lvextend --usepolicies LV
[ -r|--resizefs ]
[ --fs String ]
[ --fsmode String ]
[ 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|json_std ]
[ --type linear|striped|snapshot|raid|mirror|thin|thin-pool|vdo|vdo-pool|cache|cache-pool|writecache ]
```

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 ]
[ --devicesfile String ]
[ --devices PV ]
[ --nohints ]
[ --journal String ]
```

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@redhat9 ~]# resize2fs /dev/vg0/lv0
resize2fs 1.46.5 (30-Dec-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@redhat9 ~]# df -h /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 418M  4.4M  391M   2% /mnt/lvm
```

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

LAB #5 - 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@redhat9 ~]# 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, 0.0239133 s, 438 MB/s
```

Créez maintenant un snapshot :

```
[root@redhat9 ~]# lvcreate -s -L 5M -n testsnap /dev/vg0/lv0
Rounding up size to full physical extent 8.00 MiB
Logical volume "testsnap" created.
```

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

```
[root@redhat9 ~]# lvs
LV      VG   Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root    rhel -wi-ao---- <44.00g
swap    rhel -wi-ao---- 5.00g
lv0     vg0  owi-aos--- 456.00m
testsnap vg0  swi-a-s--- 8.00m    lv0    0.15
```



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

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

```
[root@redhat9 ~]# 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 ]
[   --headings none|abbrev|full|0|1|2 ]
[   --ignorelockingfailure ]
[   --logonly ]
[   --nameprefixes ]
[   --noheadings ]
[   --nosuffix ]
[   --readonly ]
[   --reportformat basic|json|json_std ]
[   --rows ]
[   --separator String ]
[   --shared ]
[   --unbuffered ]
[   --units [Number]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 ]
[ --devicesfile String ]
[ --devices PV ]
[ --nohints ]
[ --journal String ]
```

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

Créez maintenant un répertoire pour monter le snapshot :

```
[root@redhat9 ~]# mkdir /mnt/testsnap
```

Montez le snapshot :

```
[root@redhat9 ~]# mount /dev/vg0/testsnap /mnt/testsnap
```

Comparez le volume d'origine et le snapshot :

```
[root@redhat9 ~]# ls -l /mnt/lvm
total 10294
-rw-r--r--. 1 root root 10485760 Oct 23 15:27 10M
drwxr-xr-x. 3 root root    1024 Oct 19  2023 home
drwx-----. 2 root root    12288 Oct 23 15:21 lost+found
```

```
[root@redhat9 ~]# ls -l /mnt/testsnap/
total 10294
```

```
-rw-r--r--. 1 root root 10485760 Oct 23 15:27 10M
drwxr-xr-x. 3 root root    1024 Oct 19  2023 home
drwx-----. 2 root root    12288 Oct 23 15:21 lost+found
```

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

```
[root@redhat9 ~]# rm /mnt/lvm/10M
rm: remove regular file '/mnt/lvm/10M'? y
```

Constatez le résultat de cette suppression :

```
[root@redhat9 ~]# df -Ph /mnt/lvm
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-lv0 418M  4.4M  391M   2% /mnt/lvm

[root@redhat9 ~]# df -Ph /mnt/testsnap/
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-testsnap 418M   15M  381M   4% /mnt/testsnap

[root@redhat9 ~]# lvs
LV      VG   Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root    rhel -wi-ao---- <44.00g
swap    rhel -wi-ao----  5.00g
lv0     vg0  owi-aos--- 456.00m
testsnap vg0  swi-aos---  8.00m          lv0    0.68
```



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

LAB #6 - Suppression des Volumes

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

```
[root@redhat9 ~]# umount /mnt/testsnap/

[root@redhat9 ~]# lvremove /dev/vg0/testsnap
Do you really want to remove active logical volume vg0/testsnap? [y/n]: y
Logical volume "testsnap" successfully removed.

[root@redhat9 ~]# umount /mnt/lvm

[root@redhat9 ~]# 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@redhat9 ~]# 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|json_std ]  
[ 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 ]  
[ --devicesfile String ]  
[ --devices PV ]  
[ --nohints ]  
[ --journal String ]
```

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

Le Volume Group peut aussi être supprimé :

```
[root@redhat9 ~]# vgremove vg0  
Volume group "vg0" successfully removed
```

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

```
[root@redhat9 ~]# vgremove --help  
vgremove - Remove volume group(s)
```

```
vgremove VG|Tag|Select ...
[ -f|--force ]
[ -S|--select String ]
[ --noudevsync ]
[ --reportformat basic|json|json_std ]
[ 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 ]
[ --devicesfile String ]
[ --devices PV ]
[ --nohints ]
[ --journal String ]
```

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

Ainsi que le volume physique :

```
[root@redhat9 ~]# pvremove /dev/sdb6 /dev/sdb7 /dev/sdb9
Labels on physical volume "/dev/sdb6" successfully wiped.
Labels on physical volume "/dev/sdb7" successfully wiped.
```

Labels on physical volume "/dev/sdb9" successfully wiped.

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

```
[root@redhat9 ~]# pvremove --help
pvremove - Remove LVM label(s) from physical volume(s)
```

```
pvremove PV ...
  [ -f|--force ]
  [ --reportformat basic|json|json_std ]
  [ 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 ]
  [ --devicesfile String ]
  [ --devices PV ]
  [ --nohints ]
  [ --journal String ]
```

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

LAB #7 - Volumes Logiques en Miroir

Re-créez maintenant votre Volume Group :

```
[root@redhat9 ~]# pvcreate /dev/sdb6 /dev/sdb7 /dev/sdb9
Physical volume "/dev/sdb6" successfully created.
Physical volume "/dev/sdb7" successfully created.
Physical volume "/dev/sdb9" successfully created.

[root@redhat9 ~]# vgcreate -s 8M vg0 /dev/sdb6 /dev/sdb7 /dev/sdb9
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@redhat9 ~]# 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@redhat9 ~]# lvdisplay -m /dev/vg0/lv1
--- Logical volume ---
LV Path                /dev/vg0/lv1
LV Name                 lv1
VG Name                 vg0
LV UUID                 L7F0YB-c0tG-dE1u-0yod-LltI-PEcv-rebeIa
LV Write Access         read/write
LV Creation host, time  redhat9.ittraining.loc, 2024-10-23 15:42:34 +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            253:6
--- 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@redhat9 ~]# pvdisplay -m /dev/sdb6 /dev/sdb7 /dev/sdb9
--- Physical volume ---
PV Name                /dev/sdb6
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                TnqoyD-LQvR-pHro-wlud-vVUq-Kime-xuVTJD
--- 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/sdb7
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            eff83E-hlnA-FWdf-VQzX-c3dP-dsDf-3dvhik
--- 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/sdb9
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                Uhd7RK-T8QS-nUAV-Sh0X-4JXp-Cbju-f8o2yC
--- Physical Segments ---
Physical extent 0 to 48:
    FREE
```

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

```
[root@redhat9 ~]# lvconvert -m 0 /dev/vg0/lv1 /dev/sdb7
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.
```

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

```
[root@redhat9 ~]# 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.
```

Supprimez de nouveau votre miroir :

```
[root@redhat9 ~]# lvconvert -m 0 /dev/vg0/lv1 /dev/sdb7
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@redhat9 ~]# 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  
[ -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  
[ -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  
[ COMMON_OPTIONS ]  
[ PV ... ]
```

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

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

Merge LV images that were split from a raid1 LV.

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

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

```
lvconvert --type thin --thinpool LV LV
```

```
[ -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 ]
[ PV ... ]
```

Convert LV to a thin LV, using LV as thin-pool data volume.

lvconvert --type thin LV

```
[ -T|--thin ]
[ -r|--readahead auto|none|Number ]
[ -c|--chunksize Size[k|UNIT] ]
[ -Z|--zero y|n ]
[     --poolmetadata LV ]
[     --poolmetadatasize Size[m|UNIT] ]
[     --poolmetadataspare y|n ]
[     --metadataprofile String ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

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

lvconvert --type cache --cachepool LV LV

```
[ -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 ]
[ PV ... ]
```

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

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

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

```
lvconvert --type cache --cachevol LV LV
[ -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
[ --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
[ -c|--chunksize Size[k|UNIT] ]
```

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

Convert LV to type thin-pool.

```
lvconvert --type thin-pool LV
[ -I|--stripesize Size[k|UNIT] ]
[ -r|--readahead auto|none|Number ]
[ -c|--chunksize Size[k|UNIT] ]
[ -Z|--zero y|n ]
[ --stripes Number ]
[ --discards passdown|nopassdown|ignore ]
[ --errorwhenfull y|n ]
[ --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
[ -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
[ -n|--name LV_new ]
[ -V|--virtualsize Size[m|UNIT] ]
[ -r|--readahead auto|none|Number ]
[ -Z|--zero y|n ]
[ --metadataprofile String ]
[ --compression y|n ]
[ --deduplication y|n ]
[ --vdosettings String ]
[ COMMON_OPTIONS ]
```

Detach a cache from an LV.

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

Merge thin LV into its origin LV.

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

Merge COW snapshot LV into its origin.

```
lvconvert --mergesnapshot LV ...
[ -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
[ -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
```

```
[ -i|--interval Number ]
```

```
[ -k|--setactivationskip y|n ]
```

```
[ --usepolicies ]
```

```
[ --poolmetadataspare y|n ]
```

```
[ COMMON_OPTIONS ]
```

```
[ PV ... ]
```

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

```
lvconvert --replace PV LV
```

```
[ COMMON_OPTIONS ]
```

```
[ PV ... ]
```

Poll LV to continue conversion.

```
lvconvert --startpoll LV
```

```
[ COMMON_OPTIONS ]
```

Add or remove data integrity checksums to raid images.

```
lvconvert --raidintegrity y|n LV
```

```
[ --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 ]
[ --devicesfile String ]
[ --devices PV ]
[ --nohints ]
[ --journal String ]
```

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

LAB #8 - Modifier les Attributs LVM

En cas de présence d'un miroir, la commande **lvs** indique la présence du miroir dans la colonne **Attr** avec la lettre **m** :

```
[root@redhat9 ~]# lvs
LV   VG   Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root rhel -wi-ao---- <44.00g
swap rhel -wi-ao----  5.00g
lv1  vg0   -wi-a----- 104.00m
```



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

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

```
[root@redhat9 ~]# vgs
VG   #PV #LV #SN Attr   VSize   VFree
rhel   1   2   0 wz--n- <49.00g    0
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@redhat9 ~]# pvs
PV          VG   Fmt  Attr PSize   PFree
/dev/sda2   rhel lvm2 a--  <49.00g    0
/dev/sdb6   vg0  lvm2 a--  192.00m  88.00m
/dev/sdb7   vg0  lvm2 a--  296.00m 296.00m
/dev/sdb9   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@redhat9 ~]# lvchange -a n /dev/vg0/lv1
```

```
[root@redhat9 ~]# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
root	rhel	-wi-ao----	<44.00g									
swap	rhel	-wi-ao----	5.00g									
lv1	vg0	-wi-----	104.00m									

Pour faire l'inverse il convient de saisir la commande suivante :

```
[root@redhat9 ~]# lvchange -a y /dev/vg0/lv1
```

```
[root@redhat9 ~]# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
root	rhel	-wi-ao----	<44.00g									
swap	rhel	-wi-ao----	5.00g									
lv1	vg0	-wi-a-----	104.00m									

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

```
[root@redhat9 ~]# 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,  
--setautoactivation y|n,  
--errorwhenfull y|n,  
--discards passdown|nopassdown|ignore,  
--cachemode writethrough|writeback|passthrough,  
--cachepolicy String,  
--cachesettings String,  
--minrecoveryrate Size[k|UNIT],  
--maxrecoveryrate Size[k|UNIT],  
--vdosettings String,  
--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|Tag|Select ...
```

```
  [ -a|--activate y|n|ay ]  
  [ COMMON_OPTIONS ]
```

Resynchronize or check a raid LV.

```
lvchange --syncaction check|repair VG|LV|Tag|Select ...
```

```
  [ COMMON_OPTIONS ]
```

Reconstruct data on specific PVs of a raid LV.

```
lvchange --rebuild PV VG|LV|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|json_std ]
```

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 ]
[ --devicesfile String ]
[ --devices PV ]
[ --nohints ]
[ --journal String ]
```

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

LAB #9 - 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 option supplémentaires :

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

Saisissez donc la commande suivante :

```
[root@redhat9 ~]# lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdb7 /dev/sdb9
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/sda7` et sur `/dev/sda9` :

```
[root@redhat9 ~]# lvdisplay -m /dev/vg0/lv2
--- Logical volume ---
LV Path                /dev/vg0/lv2
LV Name                 lv2
VG Name                 vg0
LV UUID                 rgVJEa-mbyk-nQhm-rLt0-EqPB-QyI3-zyG2qX
LV Write Access         read/write
LV Creation host, time  redhat9.ittraining.loc, 2024-10-23 15:49:12 +0200
LV Status                available
# open                  0
LV Size                 112.00 MiB
Current LE              14
Segments                1
Allocation               inherit
```

```

Read ahead sectors      auto
- currently set to      512
Block device            253:3
--- Segments ---
Logical extents 0 to 13:
  Type                  striped
  Stripes               2
  Stripe size           64.00 KiB
  Stripe 0:
    Physical volume     /dev/sdb7
    Physical extents    0 to 6
  Stripe 1:
    Physical volume     /dev/sdb9
    Physical extents    0 to 6

```

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

```

[root@redhat9 ~]# lvs -o +devices
LV VG Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
root rhel -wi-ao---- <44.00g                               /dev/sda2(1280)
swap rhel -wi-ao---- 5.00g                               /dev/sda2(0)
lv1  vg0 -wi-a----- 104.00m                              /dev/sdb6(1)
lv2  vg0 -wi-a----- 112.00m                              /dev/sdb7(0),/dev/sdb9(0)

```

LAB #10 - 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@redhat9 ~]# cat /etc/lvm/backup/vg0
# Generated by LVM2 version 2.03.23(2) (2023-11-21): Wed Oct 23 15:49:12 2024

contents = "Text Format Volume Group"
version = 1

```

```
description = "Created *after* executing 'lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdb7 /dev/sdb9'"

creation_host = "redhat9.ittraining.loc"          # Linux redhat9.ittraining.loc 5.14.0-427.37.1.el9_4.x86_64 #1
SMP PREEMPT_DYNAMIC Fri Sep 13 12:41:50 EDT 2024 x86_64
creation_time = 1729691352          # Wed Oct 23 15:49:12 2024

vg0 {
    id = "akzcF6-eQ28-j9Xa-Cuiq-Kio0-dAKJ-GEK034"
    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 = "TnqoyD-LQvR-pHro-wlud-vVUq-Kime-xuVTJD"
            device = "/dev/sdb6"      # Hint only

            device_id_type = "sys_wwid"
            device_id = "t10.ATA_QEMU_HARDDISK_QM00005"
            status = ["ALLOCATABLE"]
            flags = []
            dev_size = 409600          # 200 Megabytes
            pe_start = 2048
            pe_count = 24             # 192 Megabytes
        }

        pv1 {
            id = "eff83E-hlnA-FWdf-VQzX-c3dP-dsDf-3dvhik"
```

```
        device = "/dev/sdb7"    # Hint only

        device_id_type = "sys_wwid"
        device_id = "t10.ATA_QEMU_HARDDISK_QM00005"
        status = ["ALLOCATABLE"]
        flags = []
        dev_size = 614400        # 300 Megabytes
        pe_start = 2048
        pe_count = 37    # 296 Megabytes
    }

    pv2 {
        id = "Uhd7RK-T8QS-nUAV-Sh0X-4JXp-Cbju-f8o2yC"
        device = "/dev/sdb9"    # Hint only

        device_id_type = "sys_wwid"
        device_id = "t10.ATA_QEMU_HARDDISK_QM00005"
        status = ["ALLOCATABLE"]
        flags = []
        dev_size = 819200        # 400 Megabytes
        pe_start = 2048
        pe_count = 49    # 392 Megabytes
    }
}

logical_volumes {

    lv1 {
        id = "L7F0YB-c0tG-dE1u-0yod-LltI-PEcv-rebeIa"
        status = ["READ", "WRITE", "VISIBLE"]
        flags = []
        creation_time = 1729690954    # 2024-10-23 15:42:34 +0200
        creation_host = "redhat9.ittraining.loc"
        segment_count = 1
    }
}
```

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

            type = "striped"
            stripe_count = 1           # linear

            stripes = [
                "pv0", 1
            ]
        }
    }

    lv2 {
        id = "rgVJEa-mbyk-nQhm-rLt0-EqPB-QyI3-zyG2qX"
        status = ["READ", "WRITE", "VISIBLE"]
        flags = []
        creation_time = 1729691352      # 2024-10-23 15:49:12 +0200
        creation_host = "redhat9.ittraining.loc"
        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 placées dans le répertoire **/etc/lvm/archives** :

```
[root@redhat9 ~]# ls /etc/lvm/archive/  
rhel_00000-628100095.vg  vg0_00001-944764243.vg  vg0_00003-1616314336.vg  vg0_00005-1097824442.vg  
vg0_00007-341212008.vg  vg0_00009-689917891.vg  vg0_00011-1035270871.vg  
vg0_00000-565701167.vg  vg0_00002-556906173.vg  vg0_00004-1230709712.vg  vg0_00006-1896107975.vg  
vg0_00008-439634752.vg  vg0_00010-1717657247.vg  vg0_00012-219256591.vg
```

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. Sans l'option **-f** pour spécifier la sauvegarde à restaurer, cette commande renvoie la liste des sauvegardes disponibles :

```
[root@redhat9 ~]# vgcfgbackup vg0  
Volume group "vg0" successfully backed up.
```

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

```
[root@redhat9 ~]# vgcfgbackup -f /tmp/vg0_backup vg0  
Volume group "vg0" successfully backed up.
```

Par contre, toute sauvegarde en dehors des répertoires par défaut n'est pas visible dans la sortie de la commande **vgcfgrestore -list** :

```
[root@redhat9 ~]# vgcfgbackup vg0  
Volume group "vg0" successfully backed up.  
[root@redhat9 ~]# vgcfgrestore --list vg0  
File:      /etc/lvm/archive/vg0_00000-565701167.vg  
VG name:    vg0
```

Description: Created *before* executing 'lvcreate -L 350 -n lv0 vg0'
Backup Time: Wed Oct 23 15:19:32 2024

File: /etc/lvm/archive/vg0_00001-944764243.vg
VG name: vg0
Description: Created *before* executing 'lvextend -L +100M /dev/vg0/lv0'
Backup Time: Wed Oct 23 15:24:34 2024

File: /etc/lvm/archive/vg0_00002-556906173.vg
VG name: vg0
Description: Created *before* executing 'lvcreate -s -L 5M -n testsnap /dev/vg0/lv0'
Backup Time: Wed Oct 23 15:28:18 2024

File: /etc/lvm/archive/vg0_00003-1616314336.vg
VG name: vg0
Description: Created *before* executing 'lvcreate -s -L 5M -n testsnap /dev/vg0/lv0'
Backup Time: Wed Oct 23 15:28:18 2024

File: /etc/lvm/archive/vg0_00004-1230709712.vg
VG name: vg0
Description: Created *before* executing 'lvremove /dev/vg0/testsnap'
Backup Time: Wed Oct 23 15:33:25 2024

File: /etc/lvm/archive/vg0_00005-1097824442.vg
VG name: vg0
Description: Created *before* executing 'lvremove /dev/vg0/testsnap'
Backup Time: Wed Oct 23 15:33:25 2024

File: /etc/lvm/archive/vg0_00006-1896107975.vg
VG name: vg0
Description: Created *before* executing 'lvremove /dev/vg0/lv0'
Backup Time: Wed Oct 23 15:33:54 2024

File: /etc/lvm/archive/vg0_00007-341212008.vg

```
VG name:      vg0
Description:   Created *before* executing 'lvcreate -m 1 -L 100M -n lv1 vg0'
Backup Time:   Wed Oct 23 15:42:34 2024

File:         /etc/lvm/archive/vg0_00008-439634752.vg
VG name:      vg0
Description:   Created *before* executing 'lvcreate -m 1 -L 100M -n lv1 vg0'
Backup Time:   Wed Oct 23 15:42:34 2024

File:         /etc/lvm/archive/vg0_00009-689917891.vg
VG name:      vg0
Description:   Created *before* executing 'lvconvert -m 0 /dev/vg0/lv1 /dev/sdb7'
Backup Time:   Wed Oct 23 15:44:01 2024

File:         /etc/lvm/archive/vg0_00010-1717657247.vg
VG name:      vg0
Description:   Created *before* executing 'lvconvert -m 1 /dev/vg0/lv1'
Backup Time:   Wed Oct 23 15:44:22 2024

File:         /etc/lvm/archive/vg0_00011-1035270871.vg
VG name:      vg0
Description:   Created *before* executing 'lvconvert -m 0 /dev/vg0/lv1 /dev/sdb7'
Backup Time:   Wed Oct 23 15:44:39 2024

File:         /etc/lvm/archive/vg0_00012-219256591.vg
VG name:      vg0
Description:   Created *before* executing 'lvcreate -i2 -I64 -n lv2 -L 100M vg0 /dev/sdb7 /dev/sdb9'
Backup Time:   Wed Oct 23 15:49:12 2024

File:         /etc/lvm/backup/vg0
VG name:      vg0
Description:   Created *after* executing 'vgcfgbackup vg0'
Backup Time:   Wed Oct 23 15:53:54 2024
```

RAID Logiciel

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.

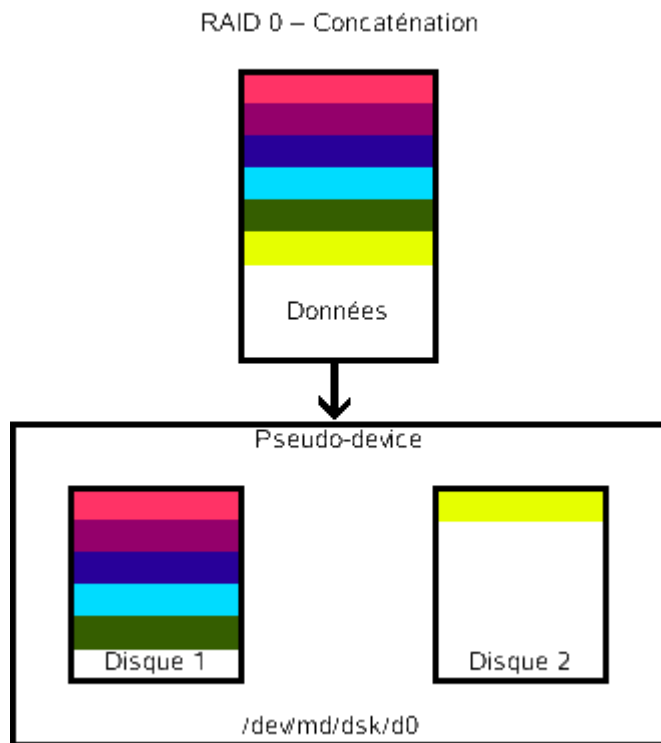
Bandes de données

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

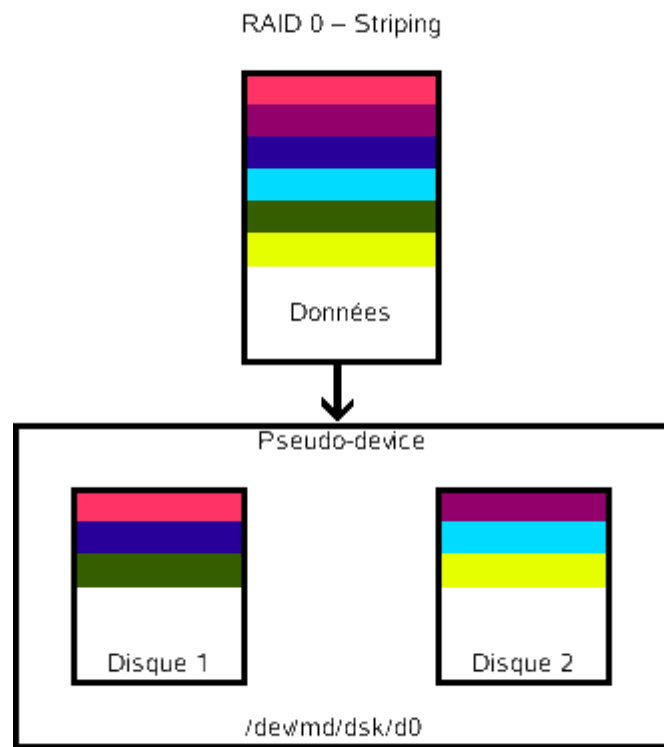
- Récupération de l'espace disque.

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

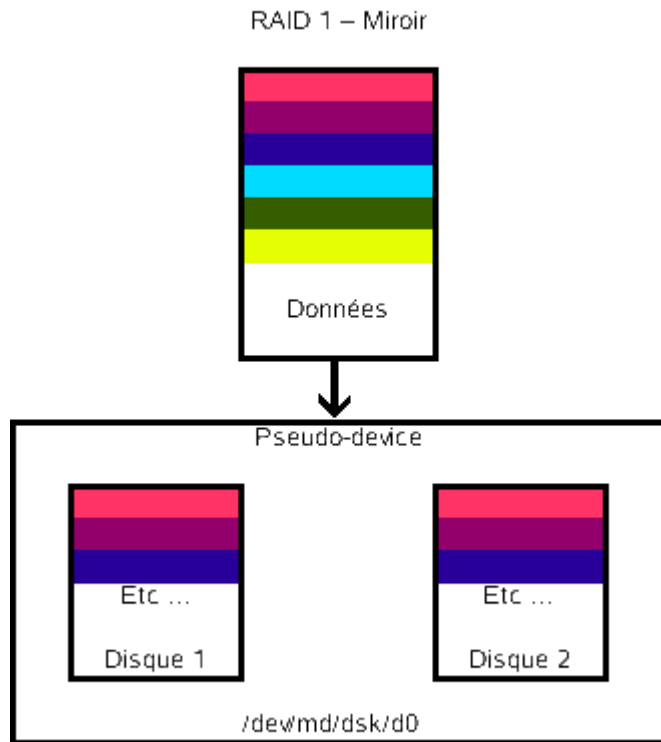
- Augmentation des performances d'E/S par écriture en parallèle sur les disques.

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

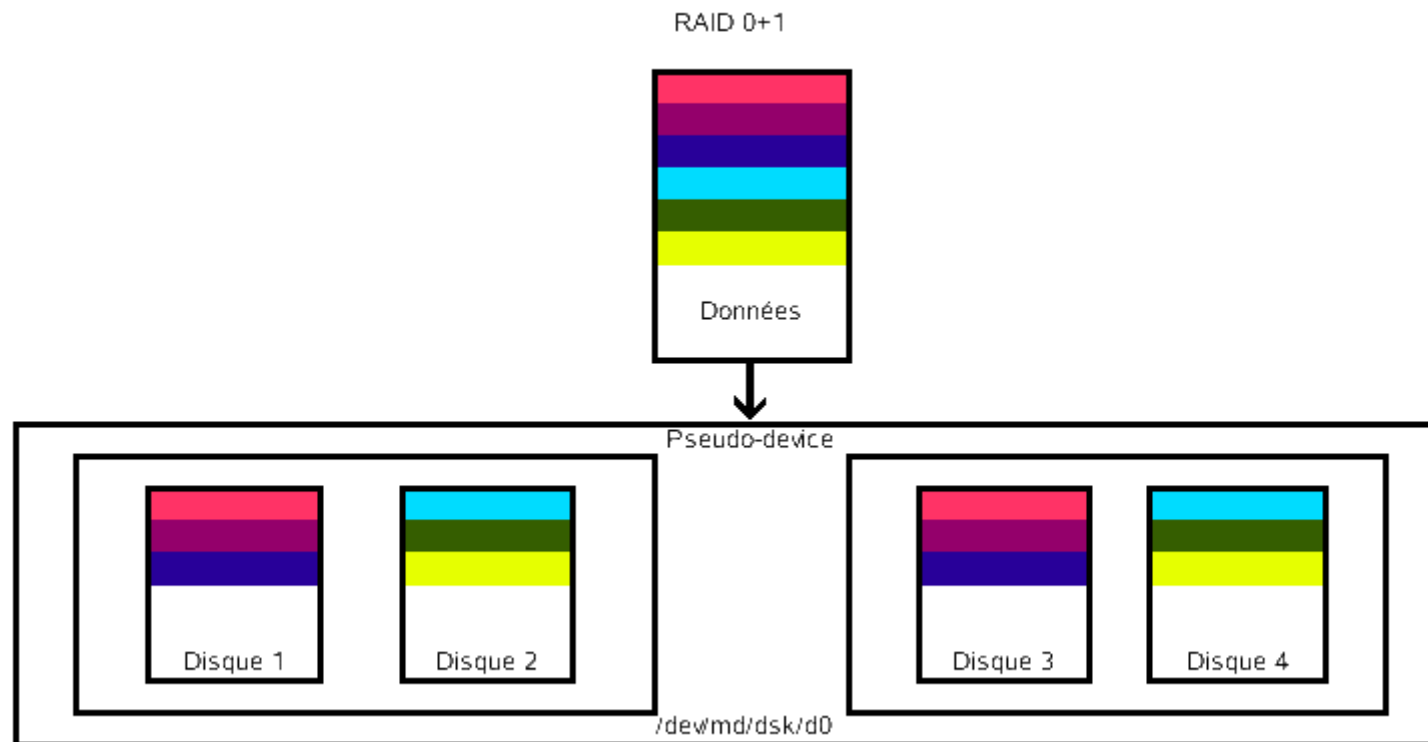
- Protection des données contre une défaillance d'un disque.

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

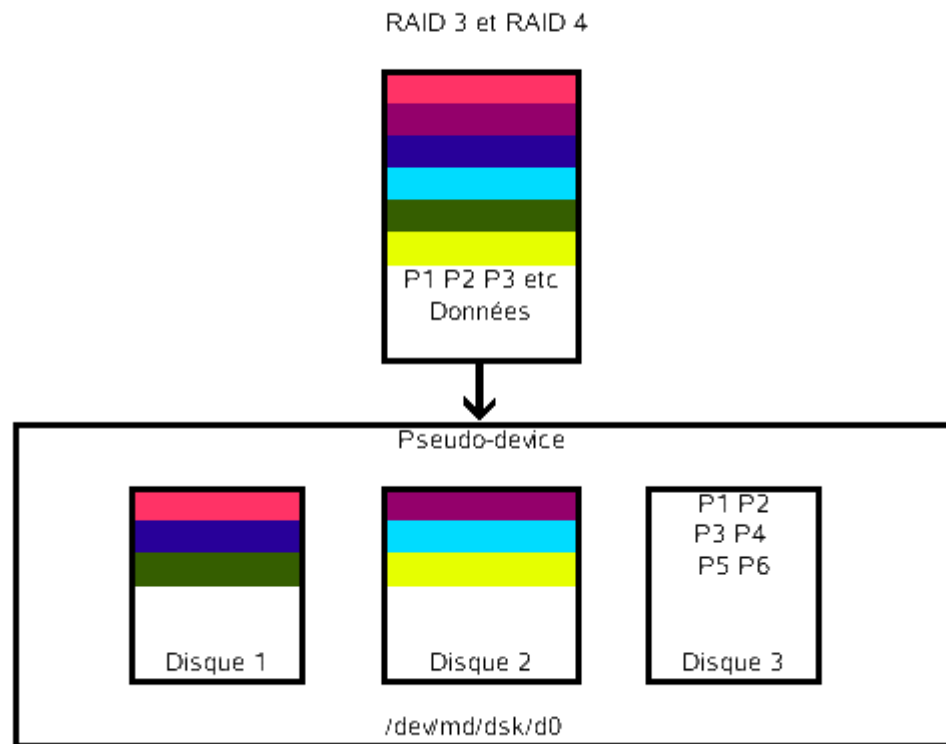
- Coûteux à cause de l'augmentation du nombre de disques.

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 :



En RAID 3, la taille des segments n'est pas modifiable et est fixée à 512 octets (en RAID 3 : un segment = un secteur de disque dur = 512 octets).

En RAID 4, la taille des segments est variable et se modifie en temps réel. Cela implique que les informations de parité doivent être mise à jour à chaque écriture afin de vérifier si la taille des segments a été modifiée.

Avantages

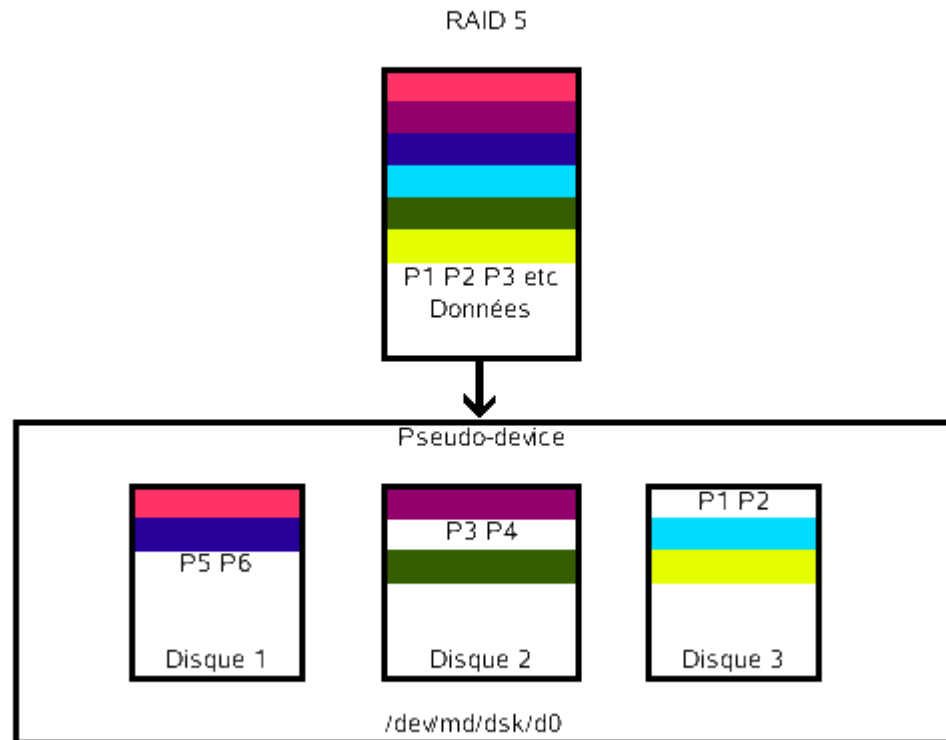
- Protection des données contre une défaillance d'un disque.

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, toutes deux issues de la technologie RAID 5 :

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

LAB #11 - Mise en Place du RAID 5 Logiciel

11.1 - Préparer le disque

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

```
[root@redhat9 ~]# fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 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: 0xd00dfc8a

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	2099199	2097152	1G	83	Linux
/dev/sda2		2099200	104857599	102758400	49G	8e	Linux LVM

Disk /dev/sdb: 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: 0x23a56c2e

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended
/dev/sdb5		618496	1642495	1024000	500M	fd	Linux raid autodetect
/dev/sdb6		1644544	2054143	409600	200M	8e	Linux LVM
/dev/sdb7		2056192	2670591	614400	300M	8e	Linux LVM
/dev/sdb8		2672640	3696639	1024000	500M	fd	Linux raid autodetect
/dev/sdb9		3698688	4517887	819200	400M	8e	Linux LVM
/dev/sdb10		4519936	5543935	1024000	500M	fd	Linux raid autodetect
/dev/sdb11		5545984	6569983	1024000	500M	fd	Linux raid autodetect
/dev/sdb12		6572032	6981631	409600	200M	83	Linux

Disk /dev/sdc: 6 GiB, 6442450944 bytes, 12582912 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/sdd: 6 GiB, 6442450944 bytes, 12582912 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/sde: 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

Disk /dev/mapper/rhel-root: 44 GiB, 47240445952 bytes, 92266496 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/rhel-swap: 5 GiB, 5368709120 bytes, 10485760 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-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
```

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

```
/dev/sdb5      618496  1642495  1024000  500M fd Linux raid autodetect
/dev/sdb8      2672640  3696639  1024000  500M fd Linux raid autodetect
/dev/sdb10     4519936  5543935  1024000  500M fd Linux raid autodetect
/dev/sdb11     5545984  6569983  1024000  500M fd Linux raid autodetect
```

11.2 - 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>
```

Saisissez maintenant la commande suivante :

```
[root@redhat9 ~]# mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sdb5 /dev/sdb8 /dev/sdb10
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@redhat9 ~]# 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@redhat9 ~]# 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@redhat9 ~]# 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.
--write-zeroes    : Write zeroes to the disks before creating. This will bypass initial sync.
```

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

```
[root@redhat9 ~]# 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@redhat9 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md1 : active raid5 sdb10[3] sdb8[1] sdb5[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@redhat9 ~]# 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@redhat9 ~]# mdadm --detail /dev/md1
/dev/md1:
  Version : 1.2
  Creation Time : Wed Oct 23 16:14:41 2024
  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 : Wed Oct 23 16:14:55 2024
State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0
```

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

```
Consistency Policy : resync
```

```
Name : redhat9.ittraining.loc:1 (local to host redhat9.ittraining.loc)
UUID : 3645cd8c:03ae466e:b42c4849:03d1eae5
Events : 18
```

Number	Major	Minor	RaidDevice	State	
0	8	21	0	active sync	/dev/sdb5
1	8	24	1	active sync	/dev/sdb8
3	8	26	2	active sync	/dev/sdb10

Le nom **redhat9.ittraining.loc:1** n'est pas compatible avec la norme POSIX. Pour rectifier ceci, renommez l'array :

```
[root@redhat9 ~]# mdadm --stop /dev/md1
mdadm: stopped /dev/md1
```

```
[root@redhat9 ~]# mdadm --assemble /dev/md1 --name=redhat.ittraining.loc --update=name /dev/sdb5 /dev/sdb8
/dev/sdb10
mdadm: /dev/md1 has been started with 3 drives.
```

```
[root@redhat9 ~]# mdadm --query /dev/md1
/dev/md1: 996.00MiB raid5 3 devices, 0 spares. Use mdadm --detail for more detail.
```

```
[root@redhat9 ~]# mdadm --detail /dev/md1
/dev/md1:
```

```
    Version : 1.2
  Creation Time : Wed Oct 23 16:14:41 2024
    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 : Wed Oct 23 16:14:55 2024
      State : clean
  Active Devices : 3
Working Devices : 3
  Failed Devices : 0
  Spare Devices : 0


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

```
Consistency Policy : resync
```

```
    Name : redhat.ittraining.loc
    UUID : 3645cd8c:03ae466e:b42c4849:03d1eae5
  Events : 18
```

Number	Major	Minor	RaidDevice	State	
0	8	21	0	active sync	/dev/sdb5
1	8	24	1	active sync	/dev/sdb8
3	8	26	2	active sync	/dev/sdb10

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@redhat9 ~]# echo 'DEVICE /dev/sdb5 /dev/sdb8 /dev/sdb10' > /etc/mdadm.conf

[root@redhat9 ~]# mdadm --detail --scan >> /etc/mdadm.conf

[root@redhat9 ~]# cat /etc/mdadm.conf
DEVICE /dev/sdb5 /dev/sdb8 /dev/sdb10
ARRAY /dev/md1 metadata=1.2 name=redhat.ittraining.loc UUID=3645cd8c:03ae466e:b42c4849:03d1eae5
```

Mettez à jour l'initramfs :

```
[root@redhat9 ~]# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).bak.$(date +%m-%d-%H%M%S).img

[root@redhat9 ~]# dracut -f -v
...
dracut: *** Creating initramfs image file '/boot/initramfs-5.14.0-427.37.1.el9_4.x86_64.img' done ***

[root@redhat9 ~]# grep initrd /boot/loader/entries/*
/boot/loader/entries/5a35a3eb625c45cea1d33535723e791f-0-rescue.conf:initrd /initramfs-0-
rescue-5a35a3eb625c45cea1d33535723e791f.img
/boot/loader/entries/5a35a3eb625c45cea1d33535723e791f-5.14.0-284.11.1.el9_2.x86_64.conf:initrd
/initramfs-5.14.0-284.11.1.el9_2.x86_64.img $tuned_initrd
/boot/loader/entries/5a35a3eb625c45cea1d33535723e791f-5.14.0-427.37.1.el9_4.x86_64.conf:initrd
/initramfs-5.14.0-427.37.1.el9_4.x86_64.img $tuned_initrd
```

Chaque unité peut être examinée individuellement :

```
[root@redhat9 ~]# mdadm --examine /dev/sdb5
/dev/sdb5:
    Magic : a92b4efc
```

```
Version : 1.2
Feature Map : 0x0
Array UUID : 3645cd8c:03ae466e:b42c4849:03d1eae5
Name : redhat.ittraining.loc
Creation Time : Wed Oct 23 16:14:41 2024
Raid Level : raid5
Raid Devices : 3

Avail Dev Size : 1019904 sectors (498.00 MiB 522.19 MB)
Array Size : 1019904 KiB (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 : 1ef50b0e:fce4333e:95e4f392:e9470824

Update Time : Wed Oct 23 16:14:55 2024
Bad Block Log : 512 entries available at offset 16 sectors
Checksum : b5e67068 - correct
Events : 18

Layout : left-symmetric
Chunk Size : 512K

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

11.3 - 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/sdb5 est devenue défaillante :

```
[root@redhat9 ~]# mdadm --manage --set-faulty /dev/md1 /dev/sdb5
```

```
mdadm: set /dev/sdb5 faulty in /dev/md1
```

L'utilisation de la ligne de commande suivante nous confirme le statut de /dev/sdb5 :

```
[root@redhat9 ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
    Creation Time : Wed Oct 23 16:14:41 2024
    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 : Wed Oct 23 16:59:59 2024
    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 : redhat.ittraining.loc
    UUID : 3645cd8c:03ae466e:b42c4849:03d1eae5
    Events : 20

    Number Major Minor RaidDevice State
    -      0      0        0      removed
    1      8      24        1      active sync  /dev/sdb5
```

3	8	26	2	active sync	/dev/sdb10
0	8	21	-	faulty	/dev/sdb5

Il est maintenant nécessaire de supprimer /dev/sdb5 de notre RAID 5 :

```
[root@redhat9 ~]# mdadm --manage --remove /dev/md1 /dev/sdb5
mdadm: hot removed /dev/sdb5 from /dev/md1
```

A l'examen de notre RAID, on constate que /dev/sdb5 a été supprimé :

```
[root@redhat9 ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Wed Oct 23 16:14:41 2024
    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 : Wed Oct 23 17:01:05 2024
      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 : redhat.ittraining.loc
UUID : 3645cd8c:03ae466e:b42c4849:03d1eae5
Events : 21
```

Number	Major	Minor	RaidDevice	State
-	0	0	0	removed
1	8	24	1	active sync /dev/sdb8
3	8	26	2	active sync /dev/sdb10

Constatez maintenant l'existence de votre RAID :

```
[root@redhat9 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md1 : active raid5 sdb10[3] sdb8[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.

Pour ajouter un autre disque à notre RAID afin de remplacer /dev/sdb5 il convient d'utiliser l'option **-add** :

```
[root@redhat9 ~]# mdadm --manage --add /dev/md1 /dev/sdb11
mdadm: added /dev/sdb11
```

L'examen du RAID indique que /dev/sdb11 a été ajouté en tant que *spare* :

```
[root@redhat9 ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Wed Oct 23 16:14:41 2024
    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 : Wed Oct 23 17:03:30 2024
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 : 1% complete
```

```
Name : redhat.ittraining.loc
UUID : 3645cd8c:03ae466e:b42c4849:03d1eae5
Events : 23
```

Number	Major	Minor	RaidDevice	State	
4	8	27	0	spare rebuilding	/dev/sdb11
1	8	24	1	active sync	/dev/sdb8
3	8	26	2	active sync	/dev/sdb10

Vérifiez la prise en compte de la configuration :

```
[root@redhat9 ~]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda              8:0    0   50G  0 disk
```

```

└─sda1      8:1    0    1G  0 part  /boot
└─sda2      8:2    0   49G  0 part
   └─rhel-root 253:0    0   44G  0 lvm  /
   └─rhel-swap 253:1    0    5G  0 lvm  [SWAP]
sdb         8:16    0   32G  0 disk
└─sdb1      8:17    0  100M  0 part
└─sdb2      8:18    0  100M  0 part
└─sdb3      8:19    0  100M  0 part
└─sdb4      8:20    0    1K  0 part
└─sdb5      8:21    0  500M  0 part
└─sdb6      8:22    0  200M  0 part
   └─vg0-lv1 253:2    0  104M  0 lvm
└─sdb7      8:23    0  300M  0 part
   └─vg0-lv2 253:3    0  112M  0 lvm
└─sdb8      8:24    0  500M  0 part
   └─md1      9:1     0  996M  0 raid5
└─sdb9      8:25    0  400M  0 part
   └─vg0-lv2 253:3    0  112M  0 lvm
└─sdb10     8:26    0  500M  0 part
   └─md1      9:1     0  996M  0 raid5
└─sdb11     8:27    0  500M  0 part
   └─md1      9:1     0  996M  0 raid5
└─sdb12     8:28    0  200M  0 part
sdc         8:32    0    6G  0 disk
sdd         8:48    0    6G  0 disk
sde         8:64    0   32G  0 disk
sr0         11:0    1 1024M  0 rom

```

```
[root@redhat9 ~]# cat /proc/mdstat
```

```
Personalities : [raid6] [raid5] [raid4] [raid1]
```

```
md1 : active raid5 sdb11[4] sdb10[3] sdb8[1]
```

```
1019904 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
```

```
[=====>.....] recovery = 31.1% (159380/509952) finish=3.8min speed=1517K/sec
```

```
unused devices: <none>
```

```
[root@redhat9 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md1 : active raid5 sdb11[4] sdb10[3] sdb8[1]
      1019904 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
      [=====>.....]  recovery = 50.4% (257956/509952) finish=2.6min speed=1560K/sec
unused devices: <none>

[root@redhat9 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md1 : active raid5 sdb11[4] sdb10[3] sdb8[1]
      1019904 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
      [=====>..]  recovery = 92.4% (472552/509952) finish=0.3min speed=1605K/sec
unused devices: <none>

[root@redhat9 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md1 : active raid5 sdb11[4] sdb10[3] sdb8[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 :

```
<code> [root@redhat9 ~]# echo 'DEVICE /dev/sdb8 /dev/sdb10 /dev/sdb11' > /etc/mdadm.conf
```

```
[root@redhat9 ~]# mdadm -detail -scan » /etc/mdadm.conf
```

```
[root@redhat9 ~]# cat /etc/mdadm.conf DEVICE /dev/sdb8 /dev/sdb10 /dev/sdb11 ARRAY /dev/md1 metadata=1.2 name=redhat.ittraining.loc
UUID=3645cd8c:03ae466e:b42c4849:03d1eae5 </code>
```

Mettez à jour l'initramfs :

```
[root@redhat9 ~]# dracut -f -v
...
dracut: *** Creating initramfs image file '/boot/initramfs-5.14.0-427.37.1.el9_4.x86_64.img' done ***
```

LAB #12 - Stratis

12.1 - Présentation

Stratis est une solution de gestion de volumes et de stockage pour Linux, développée par Red Hat, qui combine des technologies comme **LVM (Logical Volume Manager)** et **XFS** pour fournir une gestion simplifiée des volumes de stockage. Il est conçu pour offrir une solution robuste et facile à utiliser, tout en offrant une gestion avancée du stockage comme la gestion des snapshots, le thin provisioning, et la gestion des volumes logiques.

Voici un aperçu de la gestion du stockage avec Stratis :

Concepts clés de Stratis

- **Pool de stockage** : Un pool est une collection de disques physiques (ou d'autres types de périphériques de stockage) qui forment une base pour la création de volumes logiques. Avec Stratis, les disques physiques sont regroupés dans un pool. Vous pouvez ajouter ou retirer des disques au fur et à mesure de vos besoins,
- **Volumes logiques** : Une fois que vous avez un pool, vous pouvez créer des volumes logiques (ou systèmes de fichiers). Ces volumes logiques peuvent être utilisés comme tout autre système de fichiers dans Linux. Ils peuvent être agrandis ou réduits dynamiquement,
- **Thin provisioning** : Stratis permet de créer des volumes logiques "thin provisioned", c'est-à-dire que vous pouvez créer un volume avec une taille apparente plus grande que l'espace de stockage réellement disponible. Le volume utilisera l'espace physique réel uniquement lorsque les données sont effectivement écrites,
- **Snapshots** : Stratis permet de créer des snapshots, qui sont des copies instantanées des volumes à un moment donné. Ces snapshots peuvent être utilisés pour des sauvegardes ou pour restaurer un état précédent du système de fichiers.

Gestion simplifiée

L'un des principaux objectifs de Stratis est de simplifier la gestion des volumes. Il combine plusieurs technologies complexes (comme LVM, D-Bus, et XFS) sous une interface de gestion unique, accessible via des commandes simples.

Avantages de Stratis

- **Facilité d'utilisation** : L'abstraction des technologies sous-jacentes et l'interface simple réduisent la complexité pour les administrateurs systèmes,
- **Thin provisioning** et **snapshots** intégrés,
- **Gestion dynamique** des disques et des volumes, sans avoir besoin de redémarrer le système,
- **Snapshots** rapides pour des sauvegardes ou restaurations faciles.

Stratis vs ZFS/Btrfs

Stratis est parfois comparé à d'autres systèmes de gestion de stockage comme **ZFS** ou **Btrfs**, qui offrent aussi des fonctionnalités similaires (volumes logiques, snapshots, etc.). Cependant, Stratis se distingue en s'appuyant sur des technologies bien établies de Linux, comme **LVM** et **XFS**, plutôt que de réinventer entièrement un nouveau système de fichiers.

En résumé, Stratis simplifie la gestion du stockage en regroupant plusieurs fonctionnalités avancées dans un cadre simple et accessible pour les administrateurs Linux, tout en utilisant des technologies éprouvées.

12.2 - Travailler avec Stratis

Installation

Le support de Stratis peut être obtenu en installant les paquets **stratisd** et **stratis-cli** :

```
[root@redhat9 ~]# dnf install stratisd stratis-cli -y
```

Activez et démarrez le service stratis :

```
[root@redhat9 ~]# systemctl enable --now stratisd
```

```
[root@redhat9 ~]# systemctl status stratisd
```

```
● stratisd.service - Stratis daemon
   Loaded: loaded (/usr/lib/systemd/system/stratisd.service; enabled; preset: enabl>
   Active: active (running) since Thu 2024-10-24 06:51:25 CEST; 10s ago
     Docs: man:stratisd(8)
  Main PID: 2676 (stratisd)
    Tasks: 8 (limit: 48799)
   Memory: 2.0M
      CPU: 22ms
   CGroup: /system.slice/stratisd.service
           └─2676 /usr/libexec/stratisd --log-level debug

Oct 24 06:51:25 redhat9.ittraining.loc systemd[1]: Starting Stratis daemon...
Oct 24 06:51:25 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T04:51:25Z INFO st>
Oct 24 06:51:25 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T04:51:25Z INFO st>
Oct 24 06:51:25 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T04:51:25Z INFO st>
Oct 24 06:51:25 redhat9.ittraining.loc systemd[1]: Started Stratis daemon.
Oct 24 06:51:25 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T04:51:25Z INFO st>
[root@redhat9 ~]#
```

Création d'un Pool

Créez le pool de stockage **pool1** sur **/dev/sdc** :

```
[root@redhat9 ~]# stratis pool create pool1 /dev/sdc
```

Consultez la taille du pool :

```
[root@redhat9 ~]# stratis pool list
```

Name	Total / Used / Free	Properties	UUID	Alerts
pool1	6 GiB / 526 MiB / 5.49 GiB	~Ca,~Cr, Op	4cfd12ac-57e7-4c9d-8f48-6c33f482fc46	WS001

Dans la colonne **Properties**, le caractère ~ indique que la propriété indiquée n'est pas présente. Les propriétés sont :

- **Ca** - Cache,
- **Cr** - Chiffrement,
- **Op** - Surprovisionnement.

L'alerte Stratis « WS001 » signifie que tous les périphériques du pool ont été entièrement alloués. Cela signifie que la capacité du pool de stockage est entièrement utilisée et qu'il n'y a plus d'espace disponible pour de nouvelles attributions. Cette alerte est critique car elle indique que le pool est effectivement plein et qu'il peut être nécessaire de le gérer ou de l'étendre pour éviter des problèmes opérationnels.

Ajoutez donc **/dev/sdd** au pool :

```
[root@redhat9 ~]# stratis pool add-data pool1 /dev/sdd
[root@redhat9 ~]# stratis pool list
```

Name	Total / Used / Free	Properties	UUID	Alerts
pool1	12 GiB / 534 MiB / 11.48 GiB	~Ca,~Cr, Op	4cfd12ac-57e7-4c9d-8f48-6c33f482fc46	

```
[root@redhat9 ~]# stratis blockdev list pool1
```

Pool Name	Device Node	Physical Size	Tier	UUID
pool1	/dev/sdc	6 GiB	DATA	bdab7f32-db56-4fcf-b78f-af1cda92a72c
pool1	/dev/sdd	6 GiB	DATA	bcd0f548-e4b3-45b1-b334-97d9b16e5030

Création d'un Système de Fichiers

Créez le système de fichiers en provisionnement fin **filesystem1** :

```
[root@redhat9 ~]# stratis filesystem create pool1 filesystem1
```

Consultez la liste des systèmes de fichiers :

```
root@redhat9 ~]# stratis filesystem list
```

Pool	Filesystem	Total / Used / Free / Limit	Created	Device
pool1	filesystem1	1 TiB / 545 MiB / 1023.47 GiB / None	Oct 24 2024 07:12	/dev/stratis/pool1/filesystem1

```
e80a1eeb-95d0-413a-8641-2a07ea685b6a
```

Montage du System de Fichiers

Créez maintenant le point de montage **/volume** :

```
[root@redhat9 ~]# mkdir /volume
```

Montez **filesystem1** sur **/volume** :

```
[root@redhat9 ~]# mount /dev/stratis/pool1/filesystem1 /volume
```

Vérifiez que **/dev/stratis/pool1/filesystem1** a été monté :

```
[root@redhat9 ~]# mount | grep volume
/dev/mapper/stratis-1-4cfd12ac57e74c9d8f486c33f482fc46-thin-fs-e80a1eeb95d0413a86412a07ea685b6a on /volume type
xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=128k,sunit=256,swidth=2048,noquota)
```

Obtenez l'UUID de **/dev/stratis/pool1/filesystem1** :

```
[root@redhat9 ~]# lsblk --output=UUID /dev/stratis/pool1/filesystem1
UUID
e80a1eeb-95d0-413a-8641-2a07ea685b6a
```

Modifiez le fichier **/etc/fstab** :

```
[root@redhat9 ~]# vi /etc/fstab
[root@redhat9 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Oct 19 16:05:58 2023
```

```
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root    /                xfs      defaults        0 0
UUID=6f6c5bb9-30be-4734-bc23-03fed8541616 /boot            xfs      defaults        0 0
/dev/mapper/rhel-swap    none             swap     defaults        0 0
UUID=e80a1eeb-95d0-413a-8641-2a07ea685b6a /volume xfs defaults,x-systemd.requires=stratisd.service 0 0
```

Démontez maintenant **/volume** :

```
[root@redhat9 ~]# umount /volume
[root@redhat9 ~]# mount | grep volume
```

Montez **/volume** à partir de **/etc/fstab** :

```
[root@redhat9 ~]# mount -a
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.

[root@redhat9 ~]# mount | grep volume
/dev/mapper/stratis-1-4cfd12ac57e74c9d8f486c33f482fc46-thin-fs-e80a1eeb95d0413a86412a07ea685b6a on /volume type
xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=128k,sunit=256,swidth=2048,noquota,x-
systemd.requires=stratisd.service)

[root@redhat9 ~]# systemctl daemon-reload
```

Croissance Dynamique du Système de Fichiers

Consultez l'utilisation de l'espace disque de **filesystem1** :

```
[root@redhat9 ~]# stratis filesystem list
```

Pool	Filesystem	Total / Used / Free / Limit	Created	Device
UUID				
pool1	filesystem1	1 TiB / 545 MiB / 1023.47 GiB / None	Oct 24 2024 07:12	/dev/stratis/pool1/filesystem1

```
e80a1eeb-95d0-413a-8641-2a07ea685b6a
```

Créez un fichier de 2GiB dans le système de fichiers **filesystem1** :

```
[root@redhat9 ~]# dd if=/dev/urandom of=/volume/file bs=1M count=2048
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 26.3089 s, 81.6 MB/s
```

Consultez l'utilisation de l'espace disque de **filesystem1** :

```
[root@redhat9 ~]# stratis filesystem list
```

Pool	Filesystem	Total / Used / Free / Limit	Created	Device
UUID				
pool1	filesystem1	1 TiB / 6.53 GiB / 1017.47 GiB / None	Oct 24 2024 07:12	/dev/stratis/pool1/filesystem1

```
e80a1eeb-95d0-413a-8641-2a07ea685b6a
```



Important : Notez que l'utilisation du disque a augmentée.

Création d'un Snapshot

Créez le snapshot **filesystem1-snap** de **filesystem1** :

```
[root@redhat9 ~]# stratis filesystem snapshot pool1 filesystem1 filesystem1-snap
```

Vérifiez la présence du snapshot :

```
[root@redhat9 ~]# stratis filesystem list
```

Pool	Filesystem	Total / Used / Free / Limit	Created	Device
pool1	filesystem1	1 TiB / 6.53 GiB / 1017.47 GiB / None	Oct 24 2024 07:12	/dev/stratis/pool1/filesystem1
pool1	filesystem1-snap	1 TiB / 6.53 GiB / 1017.47 GiB / None	Oct 24 2024 07:44	/dev/stratis/pool1/filesystem1-snap

Supprimez le fichier **/volume/file** :

```
[root@redhat9 ~]# rm -f /volume/file
```

Vérifiez la suppression de **/volume/file**:

```
[root@redhat9 ~]# ls -l /volume
```

total 0

Créez le point de montage **/filesystem1-snap** :

```
[root@redhat9 ~]# mkdir /filesystem1-snap
```

Montez **/dev/stratis/pool1/filesystem1-snap** sur **/filesystem1-snap** :

```
[root@redhat9 ~]# mount /dev/stratis/pool1/filesystem1-snap /filesystem1-snap
```

Vérifiez la présence du fichier **/filesystem1-snap/file** :

```
[root@redhat9 ~]# ls -l /filesystem1-snap
```

total 2097152

```
-rw-r--r--. 1 root root 2147483648 Oct 24 07:37 file
```

Suppression des Systèmes de Fichiers

Démontez les deux systèmes de fichiers :

```
[root@redhat9 ~]# umount /volume
```

```
[root@redhat9 ~]# umount /filesystem1-snap
```

Supprimez les deux systèmes de fichiers :

```
[root@redhat9 ~]# stratis filesystem destroy pool1 filesystem1-snap
```

```
[root@redhat9 ~]# stratis filesystem destroy pool1 filesystem1
```

```
[root@redhat9 ~]# stratis filesystem list
```

Pool	Filesystem	Total / Used / Free / Limit	Created	Device	UUID
------	------------	-----------------------------	---------	--------	------

Editez le fichier **/etc/fstab** :

```
[root@redhat9 ~]# vi /etc/fstab
```

```
[root@redhat9 ~]# cat /etc/fstab
```

```
#  
# /etc/fstab  
# Created by anaconda on Thu Oct 19 16:05:58 2023  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
#  
# After editing this file, run 'systemctl daemon-reload' to update systemd
```

```
# units generated from this file.
#
/dev/mapper/rhel-root    /                xfs      defaults        0 0
UUID=6f6c5bb9-30be-4734-bc23-03fed8541616 /boot          xfs      defaults        0 0
/dev/mapper/rhel-swap    none           swap     defaults        0 0
```

Supprimez les deux points de montage :

```
[root@redhat9 ~]# rm -rf /volume /filesystem1-snap
```

Désactiver et arrêtez le service **stratisd** :

```
[root@redhat9 ~]# systemctl disable --now stratisd
Removed "/etc/systemd/system/multi-user.target.wants/stratisd.service".

[root@redhat9 ~]# systemctl status stratisd
○ stratisd.service - Stratis daemon
   Loaded: loaded (/usr/lib/systemd/system/stratisd.service; disabled; preset: enabled)
   Active: inactive (dead)
     Docs: man:stratisd(8)

Oct 24 07:58:07 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T05:58:07Z INFO
stratisd::dbus_api::pool::pool_3_0::methods] Filesystems with UUIDs 58604780-5686-4f92-b4dc-48ca6a35a64f were
successfully del>
Oct 24 07:58:14 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T05:58:14Z INFO
stratisd::dbus_api::pool::pool_3_0::methods] Filesystems with UUIDs e80a1eeb-95d0-413a-8641-2a07ea685b6a were
successfully del>
Oct 24 08:02:37 redhat9.ittraining.loc systemd[1]: Stopping Stratis daemon...
Oct 24 08:02:37 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T06:02:37Z INFO stratisd::stratis::run] Caught
SIGINT; exiting...
Oct 24 08:02:37 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T06:02:37Z INFO stratisd::dbus_api::util] D-
Bus tree handler was notified to exit
Oct 24 08:02:37 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T06:02:37Z INFO
stratisd::stratis::udev_monitor] udev thread was notified to exit
```

```
Oct 24 08:02:37 redhat9.ittraining.loc stratisd[2676]: [2024-10-24T06:02:37Z INFO
stratisd::dbus_api::connection] D-Bus connection handler thread notified to exit
Oct 24 08:02:37 redhat9.ittraining.loc systemd[1]: stratisd.service: Deactivated successfully.
Oct 24 08:02:37 redhat9.ittraining.loc systemd[1]: Stopped Stratis daemon.
Oct 24 08:02:37 redhat9.ittraining.loc systemd[1]: stratisd.service: Consumed 3.273s CPU time.
```

Systèmes de Fichiers Journalisés

Présentation

Un journal est la partie d'un système de fichiers journalisé qui trace les opérations d'écriture tant qu'elles ne sont pas terminées et cela en vue de garantir l'intégrité des données en cas d'arrêt brutal.

L'intérêt est de pouvoir plus facilement et plus rapidement récupérer les données en cas d'arrêt brutal du système d'exploitation (coupure d'alimentation, plantage du système, etc.), alors que les partitions n'ont pas été correctement synchronisées et démontées.

Sans un tel fichier journal, un outil de récupération de données après un arrêt brutal doit parcourir l'intégralité du système de fichier pour vérifier sa cohérence. Lorsque la taille du système de fichiers est importante, cela peut durer très longtemps pour un résultat moins efficace car entraînant des pertes de données.

Linux peut utiliser un des systèmes de fichiers journalisés suivants :

Système de fichier	Taille maximum - fichier	Taille maximum - système de fichier
Ext3	2 To	32 To
Ext4	16 To	1 EiB
XFS	8 EiB	16 EiB
ReiserFS v3	8 To	16 To
JFS	4 Po	32 Po
Btrfs	16 EiB	16 EiB



A faire : Pour comparer ces six systèmes de fichier, veuillez consulter [cette page](#)

LAB #13 - Ext3

Ext3 est une évolution de Ext2 et a pour principale différence d'utiliser un fichier journal. Il peut :

- être utilisé à partir d'une partition Ext2, sans avoir à sauvegarder et à restaurer des données,
- utiliser tous les utilitaires de maintenance pour les systèmes de fichiers ext2, comme fsck,
- utiliser le logiciel dump, ce qui n'est pas le cas avec ReiserFS.

Pour plus d'information concernant Ext3, consultez [cette page](#)

13.1 - Gestion d'Ext3

Notez maintenant le numéro de la dernière partition que vous avez précédemment créée :

```
[root@redhat9 ~]# fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 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: 0xd00dfc8a

Device      Boot    Start        End    Sectors   Size Id Type
/dev/sda1   *         2048    2099199    2097152    1G 83 Linux
/dev/sda2             2099200 104857599 102758400    49G 8e Linux LVM

Disk /dev/sdb: 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: 0x23a56c2e

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	206847	204800	100M	83	Linux
/dev/sdb2		206848	411647	204800	100M	83	Linux
/dev/sdb3		411648	616447	204800	100M	83	Linux
/dev/sdb4		616448	67108863	66492416	31.7G	5	Extended
/dev/sdb5		618496	1642495	1024000	500M	fd	Linux raid autodetect
/dev/sdb6		1644544	2054143	409600	200M	8e	Linux LVM
/dev/sdb7		2056192	2670591	614400	300M	8e	Linux LVM
/dev/sdb8		2672640	3696639	1024000	500M	fd	Linux raid autodetect
/dev/sdb9		3698688	4517887	819200	400M	8e	Linux LVM
/dev/sdb10		4519936	5543935	1024000	500M	fd	Linux raid autodetect
/dev/sdb11		5545984	6569983	1024000	500M	fd	Linux raid autodetect
/dev/sdb12		6572032	6981631	409600	200M	83	Linux

Disk /dev/sdc: 6 GiB, 6442450944 bytes, 12582912 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/sdd: 6 GiB, 6442450944 bytes, 12582912 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/sde: 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

Disk /dev/mapper/rhel-root: 44 GiB, 47240445952 bytes, 92266496 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/rhel-swap: 5 GiB, 5368709120 bytes, 10485760 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/md1: 996 MiB, 1044381696 bytes, 2039808 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 524288 bytes / 1048576 bytes

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

Créez un filesystem Ext3 sur /dev/sdb12 en utilisant la commande **mke2fs -j** :

```
[root@redhat9 ~]# mke2fs -j /dev/sdb12
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: f489fe51-2f8b-444a-af89-9bb5b6301214
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```



Important : Lors de la mise en place d'un filesystem ext2/ext3/ext4, le système réserve 5% de l'espace disque pour root. Sur des disques de grande taille il est parfois préférable de récupérer une partie de cet espace en utilisant la commande **tune2fs -m n /dev/sdXY** ou **n** est le nouveau pourcentage à réserver.

13.2 - Convertir un Système de Fichiers Ext3 en Ext2

Pour vérifier si un système de fichiers Ext2 est journalisé, utilisez la commande **dumpe2fs** :

```
[root@redhat9 ~]# dumpe2fs -h /dev/sdb12
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem volume name:    <none>
Last mounted on:          <not available>
Filesystem UUID:          f489fe51-2f8b-444a-af89-9bb5b6301214
Filesystem magic number:  0xEF53
```

```
Filesystem revision #: 1 (dynamic)
Filesystem features:  has_journal ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:     signed_directory_hash
Default mount options: user_xattr acl
Filesystem state:     clean
Errors behavior:      Continue
Filesystem OS type:   Linux
Inode count:          51200
Block count:          204800
Reserved block count: 10240
Overhead clusters:    18495
Free blocks:          186274
Free inodes:          51189
First block:          1
Block size:           1024
Fragment size:        1024
Reserved GDT blocks:  256
Blocks per group:     8192
Fragments per group:  8192
Inodes per group:      2048
Inode blocks per group: 512
Filesystem created:   Thu Oct 24 08:25:48 2024
Last mount time:      n/a
Last write time:      Thu Oct 24 08:26:00 2024
Mount count:          0
Maximum mount count:  -1
Last checked:         Thu Oct 24 08:25:48 2024
Check interval:       0 (<none>)
Reserved blocks uid:  0 (user root)
Reserved blocks gid:  0 (group root)
First inode:          11
Inode size:           256
Required extra isize: 32
Desired extra isize:  32
```

```
Journal inode:      8
Default directory hash: half_md4
Directory Hash Seed: e7f0145e-3007-4ee4-b433-4ac7d3480c57
Journal backup:     inode blocks
Journal features:    (none)
Total journal size:  4096k
Total journal blocks: 4096
Max transaction length: 4096
Fast commit length:  0
Journal sequence:    0x00000001
Journal start:       0
```



Important : Le drapeau **Filesystem features: has_journal ...** démontre que Ext3 est utilisé sur cette partition.

Pour supprimer le journal du système de fichier Ext3 sur cette partition, il convient d'utiliser la commande **tune2fs**

```
[root@redhat9 ~]# tune2fs -O ^has_journal /dev/sdb12
tune2fs 1.46.5 (30-Dec-2021)
```

Constatez le résultat de cette commande :

```
[root@redhat9 ~]# dumpe2fs -h /dev/sdb12
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         f489fe51-2f8b-444a-af89-9bb5b6301214
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:         signed_directory_hash
```

```
Default mount options:  user_xattr acl
Filesystem state:       clean
Errors behavior:        Continue
Filesystem OS type:     Linux
Inode count:            51200
Block count:            204800
Reserved block count:   10240
Overhead clusters:     14399
Free blocks:            190387
Free inodes:            51189
First block:            1
Block size:             1024
Fragment size:          1024
Reserved GDT blocks:    256
Blocks per group:       8192
Fragments per group:    8192
Inodes per group:       2048
Inode blocks per group: 512
Filesystem created:     Thu Oct 24 08:25:48 2024
Last mount time:        n/a
Last write time:        Thu Oct 24 08:27:14 2024
Mount count:            0
Maximum mount count:    -1
Last checked:           Thu Oct 24 08:25:48 2024
Check interval:         0 (<none>)
Reserved blocks uid:    0 (user root)
Reserved blocks gid:    0 (group root)
First inode:            11
Inode size:             256
Required extra isize:   32
Desired extra isize:    32
Default directory hash: half_md4
Directory Hash Seed:    e7f0145e-3007-4ee4-b433-4ac7d3480c57
Journal backup:         inode blocks
```



Important : Notez que le drapeau **Filesystem features: has_journal ...** a été supprimé.

Supprimez maintenant l'inode du journal :

```
[root@redhat9 ~]# fsck /dev/sdb12
fsck from util-linux 2.37.4
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb12: clean, 11/51200 files, 14413/204800 blocks
```

Créez maintenant un point de montage pour /dev/sdb12 :

```
[root@redhat9 ~]# mkdir /mnt/sdb12
```

Essayez de monter /dev/sdb12 en tant que système de fichiers Ext3. Vous obtiendrez un résultat similaire à celui-ci :

```
[root@redhat9 ~]# mount -t ext3 /dev/sdb12 /mnt/sdb12
mount: /mnt/sdb12: wrong fs type, bad option, bad superblock on /dev/sdb12, missing codepage or helper program,
or other error.
```



Important : Notez l'erreur due au mauvais système de fichiers qui suit l'option **-t**.

Montez maintenant le système de fichiers en tant que Ext2 :

```
[root@redhat9 ~]# mount -t ext2 /dev/sdb12 /mnt/sdb12
```

13.3 - Convertir un Système de Fichiers Ext2 en Ext3

Pour remplacer le journal sur /dev/sdb12, il convient d'utiliser la commande **tune2fs** :

```
[root@redhat9 ~]# umount /mnt/sdb12

[root@redhat9 ~]# tune2fs -j /dev/sdb12
tune2fs 1.46.5 (30-Dec-2021)
Creating journal inode: done
```



Important : Notez que vous avez du démonter la partition avant d'exécuter la commande **tune2fs**.

13.4 - Placer le Journal sur un autre Partition

Le journal d'un système de fichiers peut être placé sur un autre périphérique bloc.

Créez un système de fichiers sur /dev/sdb11 :

```
[root@redhat9 ~]# mdadm --stop /dev/md1
mdadm: stopped /dev/md1

[root@redhat9 ~]# mke2fs -O journal_dev /dev/sdb11
mke2fs 1.46.5 (30-Dec-2021)
/dev/sdb11 contains a linux_raid_member file system labelled 'redhat.ittraining.loc'
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 512000 1k blocks and 0 inodes
Filesystem UUID: 5ab9820b-f652-4e79-99a9-07f5581cb312
```

Superblock backups stored on blocks:

Zeroing journal device:



Important : Notez l'utilisation de l'option **-O**.

Créez maintenant un système de fichiers Ext3 sur /dev/sdb12 en plaçant le journal sur /dev/sdb11 :

```
[root@redhat9 ~]# mke2fs -j -J device=/dev/sdb11 /dev/sdb12
mke2fs 1.46.5 (30-Dec-2021)
Using journal device's blocksize: 1024
/dev/sdb12 contains a ext3 file system
    last mounted on Thu Oct 24 08:29:20 2024
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: ac7c133a-3f15-4fa4-81df-b1a0c808a4d0
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Adding journal to device /dev/sdb11: done
Writing superblocks and filesystem accounting information: done
```



Important : Notez que le journal a été placé sur /dev/sdb11 grâce à l'utilisation de l'option **-J**.

13.5 - Modifier la Fréquence de Vérification du Système de Fichiers Ext3

Pour modifier la fréquence de vérification du système de fichiers sur /dev/sdb12, il convient d'utiliser soit l'option **-c**, soit l'option **-i** :

```
[root@redhat9 ~]# tune2fs -i 100d /dev/sdb12
tune2fs 1.46.5 (30-Dec-2021)
Setting interval between checks to 8640000 seconds
```

Dernièrement, pour obtenir seul l'UUID du système de fichiers, utilisez les commandes **dumpe2fs** et **grep** :

```
[root@redhat9 ~]# dumpe2fs /dev/sdb12 | grep UUID
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem UUID:          ac7c133a-3f15-4fa4-81df-b1a0c808a4d0
Journal UUID:             5ab9820b-f652-4e79-99a9-07f5581cb312
```

LAB #14 - Ext4

Le système de fichiers **Ext4** a été introduit dans le noyau **2.6.19** en mode expérimental et est devenu stable dans le noyau **2.6.28**.

Ext4 n'est pas une évolution de Ext3. Cependant il a une compatibilité ascendante avec Ext3.

Les fonctionnalités majeures d'Ext4 sont :

- la gestion des volumes d'une taille allant jusqu'à **1 024 pébioctets**,
- l'allocation par **extents** qui permettent la pré-allocation d'une zone contiguë pour un fichier afin de minimiser la fragmentation.

L'option **extents** est activée par défaut depuis le noyau **2.6.23**.

La compatibilité ascendante avec ext3 comprend :

- la possibilité de monter une partition Ext3 en tant que partition Ext4,
- la possibilité de monter une partition Ext4 en tant que partition Ext3 mais **uniquement** dans le cas où la partition Ext4 n'ait jamais utilisé l'allocation par **extents** pour enregistrer des fichiers, mais l'allocation binaire comprise par ext3.

Pour plus d'informations concernant Ext4, consultez [cette page](#).

14.1 - Créer un Système de Fichiers Ext4

Créez un système de fichiers Ext4 sur **/dev/sdb11** :

```
[root@redhat9 ~]# mkfs.ext4 /dev/sdb11
mke2fs 1.46.5 (30-Dec-2021)
/dev/sdb11 contains a jbd file system
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 512000 1k blocks and 128016 inodes
Filesystem UUID: 52c3fea6-d28b-40db-b7e5-d3c289b51043
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Consultez maintenant les caractéristiques du système de fichier :

```
[root@redhat9 ~]# dumpe2fs /dev/sdb11 | more
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          52c3fea6-d28b-40db-b7e5-d3c289b51043
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent 64bit flex_bg sparse_super
                           large_file huge_file dir_nlink extra_isize metadata_csum
```

```
Filesystem flags:      signed_directory_hash
Default mount options: user_xattr acl
Filesystem state:      clean
Errors behavior:       Continue
Filesystem OS type:    Linux
Inode count:           128016
Block count:           512000
Reserved block count:  25600
Overhead clusters:     42672
Free blocks:           469314
Free inodes:           128005
First block:           1
Block size:            1024
Fragment size:         1024
Group descriptor size: 64
Reserved GDT blocks:   256
Blocks per group:      8192
Fragments per group:   8192
Inodes per group:      2032
Inode blocks per group: 508
Flex block group size: 16
Filesystem created:    Thu Oct 24 08:35:02 2024
Last mount time:       n/a
Last write time:       Thu Oct 24 08:35:02 2024
Mount count:           0
Maximum mount count:   -1
Last checked:          Thu Oct 24 08:35:02 2024
Check interval:        0 (<none>)
Lifetime writes:       286 kB
Reserved blocks uid:   0 (user root)
Reserved blocks gid:   0 (group root)
First inode:           11
Inode size:            256
Required extra isize:  32
```

```
Desired extra isize:      32
Journal inode:           8
Default directory hash:  half_md4
Directory Hash Seed:     6bf0e2d7-9fb0-4c7b-8d32-b0ca3113c0cd
Journal backup:          inode blocks
Checksum type:           crc32c
Checksum:                0xed59106b
Journal features:        (none)
Total journal size:      8M
Total journal blocks:    8192
Max transaction length:  8192
Fast commit length:      0
Journal sequence:        0x00000001
Journal start:           0

--More--
```

14.2 - Ajouter une Etiquette au Système de Fichiers Ext4

Utilisez la commande **e2label** pour associer une étiquette au système de fichiers :

```
[root@redhat9 ~]# e2label /dev/sdb11 my_ext4

[root@redhat9 ~]# dumpe2fs /dev/sdb11 | more
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem volume name:   my_ext4
Last mounted on:          <not available>
Filesystem UUID:          52c3fea6-d28b-40db-b7e5-d3c289b51043
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent 64bit flex_bg sparse_super
                           large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
```

```
Default mount options: user_xattr acl
Filesystem state:      clean
Errors behavior:       Continue
Filesystem OS type:    Linux
Inode count:           128016
Block count:           512000
Reserved block count:  25600
Overhead clusters:     42672
Free blocks:           469314
Free inodes:           128005
First block:           1
Block size:            1024
Fragment size:         1024
Group descriptor size: 64
Reserved GDT blocks:   256
Blocks per group:      8192
Fragments per group:   8192
Inodes per group:      2032
Inode blocks per group: 508
Flex block group size: 16
Filesystem created:    Thu Oct 24 08:35:02 2024
Last mount time:       n/a
Last write time:       Thu Oct 24 08:36:06 2024
Mount count:           0
Maximum mount count:   -1
Last checked:          Thu Oct 24 08:35:02 2024
Check interval:        0 (<none>)
Lifetime writes:       286 kB
Reserved blocks uid:   0 (user root)
Reserved blocks gid:   0 (group root)
First inode:           11
Inode size:            256
Required extra isize:  32
Desired extra isize:   32
```

```
Journal inode:      8
Default directory hash: half_md4
Directory Hash Seed: 6bf0e2d7-9fb0-4c7b-8d32-b0ca3113c0cd
Journal backup:     inode blocks
Checksum type:      crc32c
Checksum:           0x22e9a63d
Journal features:    (none)
Total journal size:  8M
Total journal blocks: 8192
Max transaction length: 8192
Fast commit length:  0
Journal sequence:    0x00000001
Journal start:       0
```

--More--



Important - Notez que l'étiquette doit être de 16 caractères maximum.

Créez un point de montage dans **/mnt** et essayez de monter **/dev/sdb11** en tant qu'Ext3 :

```
[root@redhat9 ~]# mkdir /mnt/sdb11
```

```
[root@redhat9 ~]# mount -t ext3 /dev/sdb11 /mnt/sdb11
```

```
mount: /mnt/sdb11: wrong fs type, bad option, bad superblock on /dev/sdb11, missing codepage or helper program,
or other error.
```



Important - Notez l'erreur qui est signalée.

Montez de nouveau la partition **sans** stipuler le type de système de fichiers :

```
[root@redhat9 ~]# mount /dev/sdb11 /mnt/sdb11

[root@redhat9 ~]# mount | grep sdb11
/dev/sdb11 on /mnt/sdb11 type ext4 (rw,relatime,seclabel)
```



Important - Constatez que la partition a été montée en tant qu'Ext4.

14.3 - Convertir un Système de Fichiers Ext3 en Ext4

Créez un système de fichiers ext3 sur /dev/sdb12 :

```
[root@redhat9 ~]# mkfs.ext3 /dev/sdb12
mke2fs 1.46.5 (30-Dec-2021)
/dev/sdb12 contains a ext3 file system
    created on Thu Oct 24 08:33:41 2024
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: aed27e98-82d1-4a9d-a072-5f73162e8f4c
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Montez maintenant **/dev/sdb12** sur /mnt/sdb12 :

```
[root@redhat9 ~]# mount /dev/sdb12 /mnt/sdb12

[root@redhat9 ~]# ls -l /mnt/sdb12
total 12
drwx-----. 2 root root 12288 Oct 24 08:38 lost+found
```

Créez le fichier **/mnt/sdb12/check_file** :

```
[root@redhat9 ~]# touch /mnt/sdb12/check_file
```

Injectez la chaîne **check file** dans le fichier /mnt/sdb12/check_file puis démontez /dev/sdb12 :

```
[root@redhat9 ~]# echo "check file" > /mnt/sdb12/check_file

[root@redhat9 ~]# umount /dev/sdb12
```

Exécutez e2fsck sur /dev/sdb12 :

```
[root@redhat9 ~]# e2fsck /dev/sdb12
e2fsck 1.46.5 (30-Dec-2021)
/dev/sdb12: clean, 12/51200 files, 18527/204800 blocks
```

Convertissez /dev/sdb12 en Ext4 :

```
[root@redhat9 ~]# tune2fs -0 extents,uninit_bg,dir_index /dev/sdb12
tune2fs 1.46.5 (30-Dec-2021)
```

Optimisez le système de fichiers :

```
[root@redhat9 ~]# e2fsck -fDC0 /dev/sdb12
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
```

```
Pass 3: Checking directory connectivity
Pass 3A: Optimizing directories
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb12: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sdb12: 12/51200 files (0.0% non-contiguous), 18527/204800 blocks
```

Essayez de monter **/dev/sdb12** en tant qu'Ext3 :

```
[root@redhat9 ~]# mount -t ext3 /dev/sdb12 /mnt/sdb12
mount: /mnt/sdb12: wrong fs type, bad option, bad superblock on /dev/sdb12, missing codepage or helper program,
or other error.
```

Montez /dev/sdb12 sans spécifier le type de fichiers Ext3 et vérifiez le contenu du fichier **check_file** :

```
[root@redhat9 ~]# mount /dev/sdb12 /mnt/sdb12

[root@redhat9 ~]# ls -l /mnt/sdb12
total 13
-rw-r--r--. 1 root root    11 Oct 24 08:39 check_file
drwx-----. 2 root root 12288 Oct 24 08:38 lost+found

[root@redhat9 ~]# cat /mnt/sdb12/check_file
check file
```

Dernièrement, pour obtenir seul l'UUID du système de fichiers, utilisez les commandes **dumpe2fs** et **grep** :

```
[root@redhat9 ~]# dumpe2fs /dev/sdb11 | grep UUID
dumpe2fs 1.46.5 (30-Dec-2021)
Filesystem UUID:          52c3fea6-d28b-40db-b7e5-d3c289b51043

[root@redhat9 ~]# dumpe2fs /dev/sdb12 | grep UUID
dumpe2fs 1.46.5 (30-Dec-2021)
```

Filesystem UUID: aed27e98-82d1-4a9d-a072-5f73162e8f4c

LAB #15 - XFS

XFS est un système de fichiers 64-bit journalisé de haute performance créée par SGI pour son système d'exploitation IRIX. XFS est inclus par défaut avec les versions du noyau Linux 2.5.xx et 2.6.xx. XFS est le système de fichiers par défaut de RedHat 9.



Important : Le système de fichiers XFS permet l'augmentation de sa taille à chaud. Par contre, la taille d'un système de fichiers XFS ne peut pas être réduite.

Pour plus d'informations concernant XFS, consultez [cette page](#).

15.1 - Créer un Système de Fichiers XFS

Démontez **/dev/sdb12** :

```
[root@redhat9 ~]# umount /dev/sdb12
```

Créez un système de fichiers XFS sur la partition **/dev/sdb12** :

```
[root@redhat9 ~]# mkfs.xfs -f /dev/sdb12
Filesystem should be larger than 300MB.
Log size should be at least 64MB.
Support for filesystems like this one is deprecated and they will not be supported in future releases.
meta-data=/dev/sdb12          isize=512    agcount=4, agsize=12800 blks
      =                       sectsz=512   attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=1, rmapbt=0
      =                       reflink=1    bigtime=1 inobtcount=1 nnext64=0
```

```

data      =          bsize=4096   blocks=51200, imaxpct=25
          =          sunit=0      swidth=0 blks
naming    =version 2   bsize=4096   ascii-ci=0, ftype=1
log       =internal log bsize=4096   blocks=1368, version=2
          =          sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none       extsz=4096    blocks=0, rtextents=0
Discarding blocks...Done.

```



Important - Notez l'utilisation de l'option **-f** afin d'écraser le système de fichiers Ext4 existant.

Consultez maintenant les caractéristiques du système de fichier :

```

[root@redhat9 ~]# xfs_info /dev/sdb12
meta-data=/dev/sdb12      isize=512    agcount=4, agsize=12800 blks
                     =          sectsz=512   attr=2, projid32bit=1
                     =          crc=1        finobt=1, sparse=1, rmapbt=0
                     =          reflink=1     bigtime=1 inobtcount=1 nnext64=0
data      =          bsize=4096   blocks=51200, imaxpct=25
                     =          sunit=0      swidth=0 blks
naming    =version 2   bsize=4096   ascii-ci=0, ftype=1
log       =internal log bsize=4096   blocks=1368, version=2
                     =          sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none       extsz=4096    blocks=0, rtextents=0

```

15.2 - Ajouter une Etiquette au Système de Fichiers XFS

Utilisez la commande **xfs_admin** pour associer une étiquette au système de fichiers :

```
[root@redhat9 ~]# xfs_admin -L my_xfs /dev/sdb12
```

```
writing all SBs  
new label = "my_xfs"
```

Pour voir l'étiquette, utilisez la commande suivante :

```
root@redhat9 ~]# xfs_admin -l /dev/sdb12  
label = "my_xfs"
```



Important - Notez que l'étiquette doit être de 12 caractères maximum.

Dernièrement, pour obtenir seul l'UUID du système de fichiers, utilisez la commande **xfs-admin** et l'option **-u** :

```
[root@redhat9 ~]# xfs_admin -u /dev/sdb12  
UUID = 0b61a609-6997-4c63-9de9-d3d316c4d5d7
```



Important - La commande **xfs_metadump** est utilisée pour sauvegarder les méta-données du système de fichiers, tandis que la commande **xfs_mdrestore** est utilisée pour restaurer les les méta-données du système de fichiers.

Autres Systèmes de Fichiers



Important - Veuillez noter que le support des systèmes de fichiers **ReiserFS**, **JFS** et **Btrfs** est **absent** du noyau des distributions de Red Hat.

ReiserFS

ReiserFS permet :

- de meilleurs temps d'accès à des sous-répertoires que Ext3, même ceux contenant des dizaines de milliers de fichiers,
- une plus grande efficacité pour ce qui concerne le stockage des fichiers moins de quelques ko. Le gain d'espace peut aller jusqu'à 10% par rapport à Ext2/Ext3.

Pour plus d'informations concernant ReiserFS, consultez [cette page](#).

JFS

JFS Journalized File System est un système de fichiers journalisé mis au point par IBM et disponible sous licence GPL.

Pour plus d'informations concernant JFS, consultez [cette page](#).

Btrfs

Btrfs, (B-tree file system, prononcé ButterFS) est un système de fichiers expérimental basé sur la copie sur écriture sous licence GNU GPL, développé principalement par Oracle, Red Hat, Fujitsu, Intel, SUSE et STRATO AG, qui s'inspire grandement du système de fichiers ZFS utilisé par Solaris.

A noter sont les points suivants :

- Btrfs utilise des extents,
 - Btrfs stocke les données des très petits fichiers directement dans l'extent du fichier répertoire, et non dans un extent séparé,
 - Btrfs gère une notion de « sous-volumes » permettant ainsi des snapshots,
 - Btrfs possède ses techniques propres de protection des données,
 - Btrfs permet de redimensionner à chaud la taille du système de fichiers,
 - Btrfs gère le RAID 0 ainsi que le RAID 1 logiciel,
 - Btrfs gère la compression du système de fichiers.
-

Comparaison des Commandes par Système de Fichiers

Description	Ext3	Ext4	XFS	ReiserFS	JFS	Btrfs
Build a Linux filesystem	mkfs.ext3 (mke2fs -j)	mkfs.ext4 (mke4fs)	mkfs.xfs	mkfs.reiserfs (mkreiserfs)	mkfs.jfs (jfs_mkfs)	mkfs.btrfs
Check a Linux filesystem	e2fsck	e2fsck	xfs_check / xfs_repair	reiserfsck	jfs_fsck	btrfsck
Adjust tunable filesystem parameters Linux filesystems	tune2fs	tune2fs	xfs_admin	reiserfstune	jfs_tune	btrfs-show-super, btrfs filesystem show, et btrfs filesystem df
File system resizer	resize2fs	resize2fs	xfs_growfs	resize_reiserfs	S/O	btrfs filesystem resize
Dump filesystem information	dumpe2fs	dumpe2fs	xfs_info / xfs_metadump	debugreiserfs	jfs_tune	btrfstune
File system debugger	debugfs	debugfs	xfs_db	debugreiserfs	jfs_debugfs	btrfs-debug-tree
Change the label on a filesystem	e2label	e2label	xfs_admin	reiserfstune	jfs_tune	btrfs filesystem label

LAB #16 - Créer un Système de Fichiers ISO

16.1 - La Commande mkisofs

Pour créer un fichier ISO à partir d'une arborescence de fichiers, il convient d'utiliser la commande **mkisofs** :

```
[root@redhat9 ~]# cd /tmp

[root@redhat9 tmp]# mkisofs -r -T -o tmp.iso .
bash: mkisofs: command not found...
Install package 'xorriso' to provide command 'mkisofs'? [N/y] y

* Waiting in queue...
* Loading list of packages....
The following packages have to be installed:
```

```
libburn-1.5.4-4.el9.x86_64    Library for reading, mastering and writing optical discs
libisoburn-1.5.4-4.el9.x86_64 Library to enable creation and expansion of ISO-9660 filesystems
libisofs-1.5.4-4.el9.x86_64   Library to create ISO 9660 disk images
xorriso-1.5.4-4.el9.x86_64    ISO-9660 and Rock Ridge image manipulation tool
```

Proceed with changes? [N/y] y

```
* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...
xorriso 1.5.4 : RockRidge filesystem manipulator, libburnia project.
```

```
Drive current: -outdev 'stdio:tmp.iso'
Media current: stdio file, overwriteable
Media status : is blank
Media summary: 0 sessions, 0 data blocks, 0 data, 37.0g free
xorriso : NOTE : -as genisofs: Ignored option '-T'
Added to ISO image: directory '/'='/tmp'
xorriso : UPDATE :      37 files added in 1 seconds
xorriso : UPDATE :      37 files added in 1 seconds
ISO image produced: 202 sectors
Written to medium : 202 sectors at LBA 0
Writing to 'stdio:tmp.iso' completed successfully.
```

Le fichier ISO peut être monter en utilisant un périphérique de type **loop** :

```
[root@redhat9 tmp]# mount tmp.iso /mnt -o loop
mount: /mnt: WARNING: source write-protected, mounted read-only.

[root@redhat9 tmp]# ls /mnt
```

```

copy.html      file2                                systemd-private-
aea59c32b8804212a7e18c402fe2ec29-power-profiles-daemon.service-NoPIRE
dbus-59XI29LyuI systemd-private-aea59c32b8804212a7e18c402fe2ec29-colord.service-pQIfSx      systemd-private-
aea59c32b8804212a7e18c402fe2ec29-rtkit-daemon.service-RLivCo
dbus-K54lUGWRFp systemd-private-aea59c32b8804212a7e18c402fe2ec29-dbus-broker.service-Ca2dJ4    systemd-private-
aea59c32b8804212a7e18c402fe2ec29-switcheroo-control.service-DAQKND
dbus-pLIWTki5B4 systemd-private-aea59c32b8804212a7e18c402fe2ec29-httpd.service-LSYL6G        systemd-private-
aea59c32b8804212a7e18c402fe2ec29-systemd-logind.service-areXk6
dbus-UpdINCi7Cg systemd-private-aea59c32b8804212a7e18c402fe2ec29-kdump.service-T3uYS0        systemd-private-
aea59c32b8804212a7e18c402fe2ec29-upower.service-gbX2j7
file1          systemd-private-aea59c32b8804212a7e18c402fe2ec29-ModemManager.service-y08983  vg0_backup

[root@redhat9 tmp]# ls
copy.html      systemd-private-aea59c32b8804212a7e18c402fe2ec29-colord.service-pQIfSx                  systemd-
private-aea59c32b8804212a7e18c402fe2ec29-switcheroo-control.service-DAQKND
dbus-59XI29LyuI systemd-private-aea59c32b8804212a7e18c402fe2ec29-dbus-broker.service-Ca2dJ4    systemd-
private-aea59c32b8804212a7e18c402fe2ec29-systemd-logind.service-areXk6
dbus-K54lUGWRFp systemd-private-aea59c32b8804212a7e18c402fe2ec29-httpd.service-LSYL6G        systemd-
private-aea59c32b8804212a7e18c402fe2ec29-upower.service-gbX2j7
dbus-pLIWTki5B4 systemd-private-aea59c32b8804212a7e18c402fe2ec29-kdump.service-T3uYS0        tmp.iso
dbus-UpdINCi7Cg systemd-private-aea59c32b8804212a7e18c402fe2ec29-ModemManager.service-y08983
vg0_backup
file1          systemd-private-aea59c32b8804212a7e18c402fe2ec29-power-profiles-daemon.service-NoPIRE
file2          systemd-private-aea59c32b8804212a7e18c402fe2ec29-rtkit-daemon.service-RLivCo

```

Démontez maintenant le fichier ISO :

```
[root@redhat9 tmp]# cd ~
```

```
[root@redhat9 ~]# umount /tmp/tmp.iso
```

Options de la Commande mkisofs

Les options de cette commande sont :

```
[root@redhat9 ~]# mkisofs --help
Usage: xorriso -as mkisofs [options] file...
Note: This is not mkisofs. See xorriso -help, xorriso -version, man xorrisofs
Options:
  -f, -follow-links          Follow symbolic links
  -graft-points              Allow to use graft points for filenames
  -help                     Print option help
  -hfsplus                  Generate HFS+ filesystem
  -hfsplus-file-creator-type CREATOR TYPE iso_rr_path
                           Attach creator and type to a File
  -hfs-bless FOLDER_NAME    Name of Folder to be blessed
  -hfs-bless-by BLESS_TYPE ISO_RR_PATH
                           Bless ISO_RR_PATH by BLESS_TYPE {p,i,s,9,x}
  -hfsplus-serial-no HEXSTRING
                           HFS serial number: 16 characters [0-9a-fA-F]
  -hfsplus-block-size NUMBER Set HFS+ block size
  -apm-block-size NUMBER    Set Apple Partition Map block size
  -hide GLOBFILE            Hide ISO9660/RR file
  -hide-list FILE           File with list of ISO9660/RR files to hide
  -hide-joliet GLOBFILE     Hide Joliet file
  -hide-joliet-list FILE    File with list of Joliet files to hide
  -hide-hfsplus GLOBFILE    Hide HFS+ file
  -hide-hfsplus-list FILE   File with list of HFS+ files to hide
  -input-charset CHARSET    Local input charset for file name conversion
  -output-charset CHARSET   Output charset for file name conversion
  -iso-level LEVEL          Set ISO9660 conformance level (1..3) or 4 for ISO9660 version 2
  -disallow_dir_id_ext      Do not allow dot in ISO directory names
  -J, -joliet               Generate Joliet directory information
  -joliet-long              Allow Joliet file names to be 103 Unicode characters
```

-joliet-utf16	Encode Joliet names in UTF-16BE rather than UCS-2
-U, -untranslated-filenames	Allow Untranslated filenames (for HPUX & AIX - violates IS09660).
-untranslated_name_len LEN	Allow up to LEN (1..96) name characters (heavily violates IS09660).
-allow-lowercase	Allow lower case characters in addition to the current character set (violates IS09660)
-relaxed-filenames	Allow 7 bit ASCII except lower case characters (violates IS09660)
-d, -omit-period	Omit trailing periods from filenames (violates IS09660)
-l, -full-iso9660-filenames	Allow full 31 character filenames for IS09660 names
-max-iso9660-filenames	Allow 37 character filenames for IS09660 names (violates IS09660)
-N, -omit-version-number	Omit version number from IS09660 filename (violates IS09660)
-D, -disable-deep-relocation	Disable deep directory relocation (violates IS09660)
-hide-rr-moved	Relocate deep directories to /.rr_moved
-rr_reloc_dir NAME	Set deep directory relocation target in root
-uid uid	Make the owner of all files this uid.
-gid gid	Make the group owner of all files this gid.
-o FILE, -output FILE	Set output file name
-m GLOBFILE, -exclude GLOBFILE	Exclude file name
-x FILE, -old-exclude FILE	Exclude file name
-exclude-list FILE	File with list of file names to exclude
-pad	Pad output by 300k (default)
-no-pad	Do not pad output
-M FILE, -prev-session FILE	Set path to previous session to merge
-C PARAMS, -cdrecord-params	PARAMS
	Magic paramters from cdrecord
-dir-mode mode	Make the mode of all directories this mode.
-file-mode mode	Make the mode of all plain files this mode.
-path-list FILE	File with list of pathnames to process
--quoted_path_list FILE	File with list of quoted pathnames to process
-print-size	Print estimated filesystem size and exit
-quiet	Run quietly
-gui	Switch behaviour for GUI
-R, -rock	Generate Rock Ridge directory information

-r, -rational-rock	Generate rationalized Rock Ridge directory information
--norock	Disable Rock Ridge. (Strongly discouraged !)
-file_name_limit LEN	Set truncation limit for Rock Ridge names
--hardlinks	Record eventual hard link relations of files
--acl	Record eventual ACLs of files
--xattr	Record eventual user space xattr of files
--xattr-any	Record xattr of any namespace, not only user.
--md5	Compute and record MD5 checksums of data files
--scdbbackup_tag PATH NAME	With --md5 record a scdbbackup checksum tag
--for_backup	Use all options which improve backup fidelity
-V ID, -volid ID	Set Volume ID
-volset ID	Set Volume set ID
-publisher PUB	Set Volume publisher
-A ID, -appid ID	Set Application ID
-sysid ID	Set System ID
-p PREP, -preparer PREP	Set Volume preparer
-abstract FILE	Set Abstract filename
-biblio FILE	Set Bibliographic filename
-copyright FILE	Set Copyright filename
--application_use CHAR PATH	Set content of Application Use field
-jigdo-jigdo FILE	Produce a jigdo .jigdo file as well as the .iso
-jigdo-template FILE	Produce a jigdo .template file as well as the .iso
-jigdo-min-file-size SIZE	Minimum size for a file to be listed in the jigdo file
-jigdo-force-checksum PTRN	Pattern(s) where files MUST match an externally-supplied checksum
-jigdo-force-md5 PATTERN	Outdated alias of -jigdo-force-checksum
-jigdo-exclude PATTERN	Pattern(s) to exclude from the jigdo file
-jigdo-map PATTERN1=PATTERN2	Pattern(s) to map paths (e.g. Debian=/mirror/debian)
-checksum-list FILE	File containing checksums of the files that should be checked
-md5-list FILE	Outdated alias of -checksum-list
-jigdo-checksum-algorithm ALGORITHM	Choose algorithm for file matching checksums: md5, sha256 Expected in the -checksum-list FILE, written into .jigdo file.
-jigdo-template-compress ALGORITHM	

```
Choose to use gzip or bzip2 compression for template data; default is gzip
-checksum_algorithm_iso alg1,alg2,...
    Specify the checksum types desired for the output image (in .jigdo)
-checksum_algorithm_template alg1,alg2,...
    Specify the checksum types desired for the output jigdo template
-eltorito-platform FILE
    Set El Torito platform id for the next boot entry
-b FILE, -eltorito-boot FILE
    Set El Torito boot image name
-eltorito-alt-boot
    Start specifying alternative El Torito boot parameters
--efi-boot FILE
    Set El Torito EFI boot image name and type
-e FILE
    Set EFI boot image name (more rawly)
-c FILE, -eltorito-catalog FILE
    Set El Torito boot catalog name
--boot-catalog-hide
    Hide boot catalog from ISO9660/RR and Joliet
-boot-load-size #
    Set numbers of load sectors
-hard-disk-boot
    Boot image is a hard disk image
-no-emul-boot
    Boot image is 'no emulation' image
-boot-info-table
    Patch boot image with info table
--grub2-boot-info
    Patch boot image at byte 2548
-eltorito-id ID
    Set El Torito Id String
-eltorito-selcrit HEXBYTES
    Set El Torito Selection Criteria
-isohybrid-gpt-basdat
    Mark El Torito boot image as Basic Data in GPT
-isohybrid-gpt-hfsplus
    Mark El Torito boot image as HFS+ in GPT
-isohybrid-apm-hfsplus
    Mark El Torito boot image as HFS+ in APM
-part_like_isohybrid
    Mark in MBR, GPT, APM without -isohybrid-mbr
-iso_mbr_part_type
    Set type byte or GUID of ISO partition in MBR
    or type GUID if a GPT ISO partition emerges.
--gpt_disk_guid GUID
    Set GPT disk GUID or choose automatic GUID
-G FILE, -generic-boot FILE
    Set generic boot image name
--embedded-boot FILE
    Alias of -G
--protective-msdos-label
    Patch System Area by partition table
--mbr-force-bootable
    Enforce existence of bootable flag in MBR
-partition_offset LBA
    Make image mountable by first partition, too
-partition_sec_hd NUMBER
    Define number of sectors per head
```

```

-partition_hd_cyl NUMBER    Define number of heads per cylinder
-partition_cyl_align MODE   Control cylinder alignment: off, on, auto, all
-mips-boot FILE             Set mips boot image name (relative to image root)
-mipsel-boot FILE           Set mipsel boot image name (relative to image root)
-B FILES, -sparc-boot FILES Set sparc boot image names
-sparc-label label text     Set sparc boot disk label
-hppa-cmdline CMDLINE       Set hppa boot command line
-hppa-kernel-32 FILE        Set hppa 32-bit image name (relative to image root)
-hppa-kernel-64 FILE        Set hppa 64-bit image name (relative to image root)
-hppa-bootloader FILE       Set hppa boot loader file name (relative to image root)
-hppa-ramdisk FILE          Set hppa ramdisk file name (relative to image root)
-hppa-hdrverson NUMBER      Set hppa PALO header version to 4 or 5
-alpha-boot FILE            Set alpha boot image name (relative to image root)
--grub2-sparc-core FILE     Set path of core file for disk label patching
-efi-boot-part DISKFILE|--efi-boot-image
                             Set data source for EFI System Partition
-chrp-boot-part             Mark ISO image size by MBR partition type 0x96
-chrp-boot                  Alias of -chrp-boot-part
-prep-boot-part DISKFILE    Set data source for MBR partition type 0x41
-append_partition NUMBER TYPE FILE
                             Append FILE after image. TYPE is hex: 0x.. or
                             a GUID to be used if -appended_part_as_gpt.
-appended_part_as_gpt       mark appended partitions in GPT instead of MBR.
-appended_part_as_apm       mark appended partitions in APM.
--modification-date=YYYYMMDDhhmmsscc
                             Override date of creation and modification
--set_all_file_dates TIME   Override mtime, atime, ctime in all files
-isohybrid-mbr FILE         Set SYSLINUX mbr/isohdp[fp]x*.bin for isohybrid
--grub2-mbr FILE            Set GRUB2 MBR for boot image address patching
--sort-weight NUMBER FILE   Set LBA weight number to file or file tree
--sort-weight-list DISKFILE Read list of NUMBER FILE pairs for --sort-weight
--sort-weight-patterns DISKFILE --sort-weight-list with pattern expansion
--stdio_sync on|off|number  Control forced output to disk files
--no-emul-toc               Save 64 kB size on random access output files

```

```
--emul-toc           Multi-session history on such output files
--old-empty          Use old style block addresses for empty files
-z, -transparent-compression
                      Enable transparent compression of files
--zisofs-version-2   Enable processing of zisofs version 2 files
--zisofs2-susp-z2     Produce Z2 entries for zisofs version 2
--zisofs2-susp-zf     Produce ZF entries for zisofs version 2
-root DIR            Set root directory for all new files and directories
-old-root DIR        Set root directory in previous session that is searched for files
--old-root-no-md5     Do not record and use MD5 with -old-root
--old-root-no-ino     Do not use disk inode numbers with -old-root
--old-root-devno      Use disk device numbers with -old-root
-log-file LOG_FILE   Re-direct messages to LOG_FILE
--no_rc              Do not execute startup files
-v, -verbose         Verbose
-version             Print the current version
```

Report any bugs to bug-xorriso@gnu.org or in private to scdbackup@gmx.net .

LAB #17 - Systèmes de Fichiers Chiffrés sous RedHat 9

17.1 - Créer un Système de Fichiers Chiffré avec LUKS sous RedHat 9

Présentation

LUKS (Linux Unified Key Setup) permet de chiffrer l'intégralité d'un disque de telle sorte que celui-ci soit utilisable sur d'autres plates-formes et distributions de Linux (voire d'autres systèmes d'exploitation). Il supporte jusqu'à 8 mots de passe, afin que plusieurs utilisateurs soient en mesure de déchiffrer le même volume sans partager leur mot de passe.

Mise en Place

Remplissez la partition /dev/sdb12 avec des données aléatoires :

```
[root@redhat9 ~]# shred -v --iterations=1 /dev/sdb12
shred: /dev/sdb12: pass 1/1 (random)...
shred: /dev/sdb12: pass 1/1 (random)...5.7MiB/200MiB 2%
shred: /dev/sdb12: pass 1/1 (random)...10MiB/200MiB 5%
shred: /dev/sdb12: pass 1/1 (random)...14MiB/200MiB 7%
shred: /dev/sdb12: pass 1/1 (random)...18MiB/200MiB 9%
shred: /dev/sdb12: pass 1/1 (random)...23MiB/200MiB 11%
shred: /dev/sdb12: pass 1/1 (random)...27MiB/200MiB 13%
shred: /dev/sdb12: pass 1/1 (random)...31MiB/200MiB 15%
shred: /dev/sdb12: pass 1/1 (random)...36MiB/200MiB 18%
shred: /dev/sdb12: pass 1/1 (random)...41MiB/200MiB 20%
shred: /dev/sdb12: pass 1/1 (random)...45MiB/200MiB 22%
shred: /dev/sdb12: pass 1/1 (random)...49MiB/200MiB 24%
shred: /dev/sdb12: pass 1/1 (random)...53MiB/200MiB 26%
shred: /dev/sdb12: pass 1/1 (random)...57MiB/200MiB 28%
shred: /dev/sdb12: pass 1/1 (random)...61MiB/200MiB 30%
shred: /dev/sdb12: pass 1/1 (random)...64MiB/200MiB 32%
shred: /dev/sdb12: pass 1/1 (random)...68MiB/200MiB 34%
shred: /dev/sdb12: pass 1/1 (random)...73MiB/200MiB 36%
shred: /dev/sdb12: pass 1/1 (random)...77MiB/200MiB 38%
shred: /dev/sdb12: pass 1/1 (random)...81MiB/200MiB 40%
shred: /dev/sdb12: pass 1/1 (random)...86MiB/200MiB 43%
shred: /dev/sdb12: pass 1/1 (random)...90MiB/200MiB 45%
shred: /dev/sdb12: pass 1/1 (random)...94MiB/200MiB 47%
shred: /dev/sdb12: pass 1/1 (random)...98MiB/200MiB 49%
shred: /dev/sdb12: pass 1/1 (random)...105MiB/200MiB 52%
shred: /dev/sdb12: pass 1/1 (random)...108MiB/200MiB 54%
shred: /dev/sdb12: pass 1/1 (random)...112MiB/200MiB 56%
shred: /dev/sdb12: pass 1/1 (random)...113MiB/200MiB 56%
```

```
shred: /dev/sdb12: pass 1/1 (random)...117MiB/200MiB 58%
shred: /dev/sdb12: pass 1/1 (random)...121MiB/200MiB 60%
shred: /dev/sdb12: pass 1/1 (random)...125MiB/200MiB 62%
shred: /dev/sdb12: pass 1/1 (random)...129MiB/200MiB 64%
shred: /dev/sdb12: pass 1/1 (random)...133MiB/200MiB 66%
shred: /dev/sdb12: pass 1/1 (random)...137MiB/200MiB 68%
shred: /dev/sdb12: pass 1/1 (random)...141MiB/200MiB 70%
shred: /dev/sdb12: pass 1/1 (random)...145MiB/200MiB 72%
shred: /dev/sdb12: pass 1/1 (random)...148MiB/200MiB 74%
shred: /dev/sdb12: pass 1/1 (random)...152MiB/200MiB 76%
shred: /dev/sdb12: pass 1/1 (random)...156MiB/200MiB 78%
shred: /dev/sdb12: pass 1/1 (random)...159MiB/200MiB 79%
shred: /dev/sdb12: pass 1/1 (random)...163MiB/200MiB 81%
shred: /dev/sdb12: pass 1/1 (random)...167MiB/200MiB 83%
shred: /dev/sdb12: pass 1/1 (random)...170MiB/200MiB 85%
shred: /dev/sdb12: pass 1/1 (random)...174MiB/200MiB 87%
shred: /dev/sdb12: pass 1/1 (random)...178MiB/200MiB 89%
shred: /dev/sdb12: pass 1/1 (random)...182MiB/200MiB 91%
shred: /dev/sdb12: pass 1/1 (random)...187MiB/200MiB 93%
shred: /dev/sdb12: pass 1/1 (random)...190MiB/200MiB 95%
shred: /dev/sdb12: pass 1/1 (random)...194MiB/200MiB 97%
shred: /dev/sdb12: pass 1/1 (random)...198MiB/200MiB 99%
shred: /dev/sdb12: pass 1/1 (random)...200MiB/200MiB 100%
```



Important : L'étape ci-dessus est très importante parce que elle permet de s'assurer qu'aucune donnée ne reste sur la partition.

Initialisez la partition avec LUKS :

```
[root@redhat9 ~]# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb12
```



```
data = crc=1 finobt=1, sparse=1, rmapbt=0
      = reflink=1 bigtime=1 inobtcount=1 nnext64=0
      = bsize=4096 blocks=47104, imaxpct=25
      = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0, ftype=1
log =internal log bsize=4096 blocks=1368, version=2
      = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
```

Montez la partition LUKS :

```
[root@redhat9 ~]# mount /dev/mapper/sdb12 /mnt/sdb12
```

Vérifiez la présence du montage :

```
[root@redhat9 ~]# df -h | grep sdb12
/dev/mapper/sdb12      179M   11M  168M   6% /mnt/sdb12
```

Editez le fichier **/etc/crypttab/** :

```
[root@redhat9 ~]# vi /etc/crypttab

[root@redhat9 ~]# cat /etc/crypttab
sdb12 /dev/sdb12 none
```

Modifiez le fichier **/etc/fstab** :

```
[root@redhat9 ~]# vi /etc/fstab

[root@redhat9 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Oct 19 16:05:58 2023
```

```
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root    /                xfs      defaults    0 0
UUID=6f6c5bb9-30be-4734-bc23-03fed8541616 /boot            xfs      defaults    0 0
/dev/mapper/rhel-swap    none             swap     defaults    0 0
/dev/mapper/sdb12        /mnt/sdb12       xfs      defaults    0 0
```

Restaurer les SC par défaut de SELinux :

```
[root@redhat9 ~]# /sbin/restorecon -v -R /mnt/sdb12
Relabeled /mnt/sdb12 from system_u:object_r:unlabeled_t:s0 to system_u:object_r:mnt_t:s0
```

Redémarrez votre machine virtuelle :

```
[root@centos8 ~]# shutdown -r now
```



Important : Lors du démarrage de la machine virtuelle, le système devrait vous demander d'entrer la passphrase **fenestros123456789** pour permettre le montage de /dev/sda12.

Ajouter une deuxième Passphrase

Pour ajouter une deuxième passphrase, utilisez la commande cryptsetup avec la sous-commande **luksAddKey** :

```
[trainee@redhat9 ~]$ su -  
Password: fenestros  
[root@redhat9 ~]# cryptsetup luksAddKey /dev/sdb12  
Enter any existing passphrase: fenestros123456789  
Enter new passphrase for key slot: redhat123456789  
Verify passphrase: redhat123456789
```



Important : Les passphrases ne seront pas en claire. Elle le sont ici pour vous montrer des mots de passe acceptables pour LUKS.

Supprimer une Passphrase

Pour supprimer une passphrase, utilisez la commande cryptsetup avec la sous-commande **luksRemoveKey** :

```
[root@redhat9 ~]# cryptsetup luksRemoveKey /dev/sdb12  
Enter passphrase to be deleted: redhat123456789
```

Supprimer LUKS

Constatez le statut de LUKS :

```
[root@redhat9 ~]# umount /mnt/sdb12  
  
[root@redhat9 ~]# cryptsetup status sdb12  
/dev/mapper/sdb12 is active.  
type:    LUKS2  
cipher:  aes-xts-plain64  
keysize: 512 bits
```

```
key location: keyring
device: /dev/sdb12
sector size: 512
offset: 32768 sectors
size: 376832 sectors
mode: read/write
```

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

```
[root@redhat9 ~]# cryptsetup luksRemoveKey /dev/sdb12
Enter passphrase to be deleted:
```

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@redhat9 ~]# cryptsetup remove /dev/mapper/sdb12
```

Vérifiez de nouveau le statut :

```
[root@redhat9 ~]# cryptsetup status sdb12
/dev/mapper/sdb12 is inactive.
```

```
[root@redhat9 ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINTS
sda	8:0	0	50G	0	disk	
└─sda1	8:1	0	1G	0	part	/boot
└─sda2	8:2	0	49G	0	part	
└─rhel-root	253:0	0	44G	0	lvm	/
└─rhel-swap	253:1	0	5G	0	lvm	[SWAP]

```
sdb          8:16    0   32G    0 disk
├─sdb1       8:17    0  100M    0 part
├─sdb2       8:18    0  100M    0 part
├─sdb3       8:19    0  100M    0 part
├─sdb4       8:20    0    1K    0 part
├─sdb5       8:21    0  500M    0 part
├─sdb6       8:22    0  200M    0 part
│   └─vg0-lv1 253:2    0  104M    0 lvm
├─sdb7       8:23    0  300M    0 part
│   └─vg0-lv2 253:3    0  112M    0 lvm
├─sdb8       8:24    0  500M    0 part
│   └─md1      9:1     0  996M    0 raid5
├─sdb9       8:25    0  400M    0 part
│   └─vg0-lv2 253:3    0  112M    0 lvm
├─sdb10      8:26    0  500M    0 part
│   └─md1      9:1     0  996M    0 raid5
├─sdb11      8:27    0  500M    0 part
└─sdb12      8:28    0  200M    0 part
sdc          8:32    0    6G    0 disk
sdd          8:48    0    6G    0 disk
sr0         11:0     1 1024M    0 rom
```

Editez les fichiers **/etc/fstab** :

```
[root@redhat9 ~]# vi /etc/fstab
```

```
[root@redhat9 ~]# cat /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Thu Oct 19 16:05:58 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
```

```
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root    /                xfs      defaults        0 0
UUID=6f6c5bb9-30be-4734-bc23-03fed8541616 /boot           xfs      defaults        0 0
/dev/mapper/rhel-swap    none            swap     defaults        0 0
```

Supprimez le fichier **/etc/crypttab** :

```
[root@redhat9 ~]# rm -f /etc/crypttab
```

LAB #18 - Le Swap

18.1 - Taille du swap

Le tableau suivant résume la taille du swap recommandée en fonction de la RAM de la machine :

RAM	Taille du swap	Taille du Swap si Hibernation
2 Go ou moins	2 * RAM	3 * RAM
2 Go à 8 Go	Identique à la RAM	2 * RAM
8 Go à 64 Go	Au moins 4 Go	1.5 * RAM
64 Go et plus	Au moins 4 Go	Hibernation n'est pas recommandée

18.2 - Partitions de swap

Une partition de swap peut être créée sur :

- une partition du disque dur
- un RAID logiciel
- un Volume Logique

18.3 - La Commande swapon

Pour préparer un espace de swap, il convient d'utiliser la commande **mkswap**. Pour activer une partition de swap, il convient d'utiliser la commande **swapon**. Pour consulter la liste des partitions swap, il convient d'utiliser la commande **swapon** avec l'option **-s**.

```
[root@redhat9 ~]# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/dm-1	partition	5242876	0	-2



Important : Vous noterez que dans l'exemple ci-dessus, le swap n'est pas utilisé. Notez aussi qu'il existe une notion de **priorité** pour les partitions de swap.

Options de la Commande

Les options de la commande swapon sont :

```
[root@redhat9 ~]# swapon --help
```

Usage:

```
swapon [options] [<spec>]
```

Enable devices and files for paging and swapping.

Options:

-a, --all	enable all swaps from /etc/fstab
-d, --discard[=<policy>]	enable swap discards, if supported by device
-e, --ifexists	silently skip devices that do not exist
-f, --fixpgsz	reinitialize the swap space if necessary
-o, --options <list>	comma-separated list of swap options

```
-p, --priority <prio>    specify the priority of the swap device
-s, --summary            display summary about used swap devices (DEPRECATED)
  --show[=<columns>]    display summary in definable table
  --noheadings          don't print table heading (with --show)
  --raw                 use the raw output format (with --show)
  --bytes               display swap size in bytes in --show output
-v, --verbose            verbose mode

-h, --help              display this help
-V, --version           display version
```

The <spec> parameter:

```
-L <label>              synonym for LABEL=<label>
-U <uuid>               synonym for UUID=<uuid>
LABEL=<label>            specifies device by swap area label
UUID=<uuid>             specifies device by swap area UUID
PARTLABEL=<label>       specifies device by partition label
PARTUUID=<uuid>         specifies device by partition UUID
<device>                name of device to be used
<file>                  name of file to be used
```

Available discard policy types (for --discard):

```
once    : only single-time area discards are issued
pages   : freed pages are discarded before they are reused
```

If no policy is selected, both discard types are enabled (default).

Available output columns:

```
NAME    device file or partition path
TYPE    type of the device
SIZE    size of the swap area
USED    bytes in use
PRIO    swap priority
UUID    swap uuid
LABEL   swap label
```

For more details see `swapon(8)`.



Important : L'option **-p** de la commande **swapon** permet de régler la priorité.

18.4 - La Commande `swapoff`

Dans le cas de notre exemple, la partition de swap se trouve sur **/dev/dm-1**. Pour la désactiver, il convient de saisir la commande suivante :

```
[root@redhat9 ~]# swapoff /dev/dm-1
```

```
[root@redhat9 ~]# swapon -s
```

```
[root@redhat9 ~]#
```

Options de la Commande

```
[root@redhat9 ~]# swapoff --help
```

Usage:

```
swapoff [options] [<spec>]
```

Disable devices and files for paging and swapping.

Options:

<code>-a, --all</code>	disable all swaps from <code>/proc/swaps</code>
<code>-v, --verbose</code>	verbose mode
<code>-h, --help</code>	display this help
<code>-V, --version</code>	display version

The <spec> parameter:

-L <label>	LABEL of device to be used
-U <uuid>	UUID of device to be used
LABEL=<label>	LABEL of device to be used
UUID=<uuid>	UUID of device to be used
<device>	name of device to be used
<file>	name of file to be used

For more details see `swapoff(8)`.

18.5 - Créer un Fichier de Swap

Sous Linux, vous pouvez aussi bien utiliser un fichier de swap qu'une partition. La mise en place de ce fichier est faite en utilisant la commande **dd**.

La commande **dd** copie le fichier passé en entrée dans le fichier de sortie en limitant le nombre d'octets copiés par l'utilisation de deux options :

- **count**
 - le nombre
- **bs**
 - la taille du bloc à copier

Dans le cas du fichier swap il convient d'utiliser le fichier spécial **/dev/zero** en tant que fichier d'entrée. Le fichier **/dev/zero** contient une valeur **null**.

Pour créer votre fichier de swap de 256 MiB, appelé **swap**, saisissez la commande suivante :

```
[root@redhat9 ~]# dd if=/dev/zero of=/swap bs=1024k count=256
256+0 records in
256+0 records out
268435456 bytes (268 MB, 256 MiB) copied, 1.45212 s, 185 MB/s
```

Pour préparer le fichier en tant qu'espace de swap, saisissez la commande suivante :

```
[root@redhat9 ~]# mkswap /swap
```

```
mkswap: /swap: insecure permissions 0644, fix with: chmod 0600 /swap
Setting up swapspace version 1, size = 256 MiB (268431360 bytes)
no label, UUID=6bc62766-8662-478a-8556-8b17b26fc5ae
```

Pour activer le fichier avec une priorité de **1**, saisissez la commande suivante :

```
[root@redhat9 ~]# swapon -p1 /swap
swapon: /swap: insecure permissions 0644, 0600 suggested.

[root@redhat9 ~]# swapon /dev/dm-1
```

Pour visualiser les espaces swap, saisissez la commande suivante :

```
[root@redhat9 ~]# swapon -s
```

Filename	Type	Size	Used	Priority
/swap	file	262140	0	1
/dev/dm-1	partition	5242876	0	-2



Important : Le fichier de swap ayant une priorité de 1 sera utilisé avant la partition de swap ayant une priorité de -2.



Important : Pour activer le fichier swap d'une manière permanente, il convient d'ajouter une ligne au fichier **/etc/fstab**. Ne modifiez pas votre fichier **/etc/fstab** car vous allez supprimer le fichier de swap.

Désactivez maintenant le fichier swap :

```
[root@redhat9 ~]# swapoff /swap
```

```
[root@redhat9 ~]# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/dm-1	partition	5242876	0	-2

Supprimez maintenant le fichier de swap :

```
[root@redhat9 ~]# rm /swap
rm: remove regular file '/swap'? y
```