

Version : **2024.01**

Dernière mise-à-jour : 2024/04/22 13:58

LDF504 - Gestion des Disques, des Systèmes de Fichiers et du Swap

Contenu du Module

- **LDF504 - 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
- Logical Volume Manager (LVM)
 - LAB #2 - Volumes Logiques Linéaires
 - LAB #3 - Étendre un Volume Logique à Chaud
 - LAB #4 - Snapshots
 - LAB #5 - Suppression des Volumes
 - LAB #6 - Volumes Logiques en Miroir
 - LAB #7 - Modifier les Attributs LVM
 - LAB #8 - Volumes Logiques en Bandes
 - LAB #9 - Gérer les Métadonnées
- Systèmes de Fichiers Journalisés
 - Présentation
 - Ext3
 - Gestion d'Ext3
 - LAB #10 - Convertir un Système de Fichiers Ext3 en Ext2
 - LAB #11 - Convertir un Système de Fichiers Ext2 en Ext3

- LAB #12 - Placer le Journal sur un autre Partition
- LAB #13 - Modifier la Fréquence de Vérification du Système de Fichiers Ext3
- Ext4
 - LAB #14 - Créer un Système de Fichiers Ext4
 - LAB #15 - Ajouter une Étiquette au Système de Fichiers Ext4
 - LAB #16 - Convertir un Système de Fichiers Ext3 en Ext4
- XFS
 - LAB #17 - Créer un Système de Fichiers XFS
 - LAB #18 - Ajouter une Étiquette au Système de Fichiers XFS
- ReiserFS
 - LAB #19 - Créer un Système de Fichiers ReiserFS
 - LAB #20 - Ajouter une Étiquette au Système de Fichiers ReiserFS
- JFS
 - LAB #21 - Créer un Système de Fichiers JFS
 - LAB #22 - Ajouter une Étiquette au Système de Fichiers JFS
- Btrfs
 - LAB #23 - Créer un Système de Fichiers Btrfs
 - LAB #24 - Convertir un Système de Fichiers Ext4 en Btrfs
 - LAB #25 - Créer un Snapshot
 - LAB #26 - Ajouter une Étiquette au Système de Fichiers Btrfs
- Comparaison des Commandes par Système de Fichiers
- Systèmes de Fichiers Chiffrés
 - LAB #27 - Créer un Système de Fichiers Chiffré avec encryptfs
 - LAB #28 - Créer un Système de Fichiers Chiffré avec LUKS
- Raid Logiciel
 - LAB #29 - Mise en Place du RAID 5 Logiciel
 - 29.1 - Préparer le disque
 - 29.2 - Créer une Unité RAID
 - 29.3 - Remplacer une Unité Défaillante
- Le Swap
 - Taille du swap
 - Partitions de swap
 - La Commande swapon
 - La Commande swapoff

- LAB #30 - 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** - disque SATA ou SCSI
 - Esclave
 - **hdd** - disque IDE
 - **sdd** - disque SATA ou SCSI

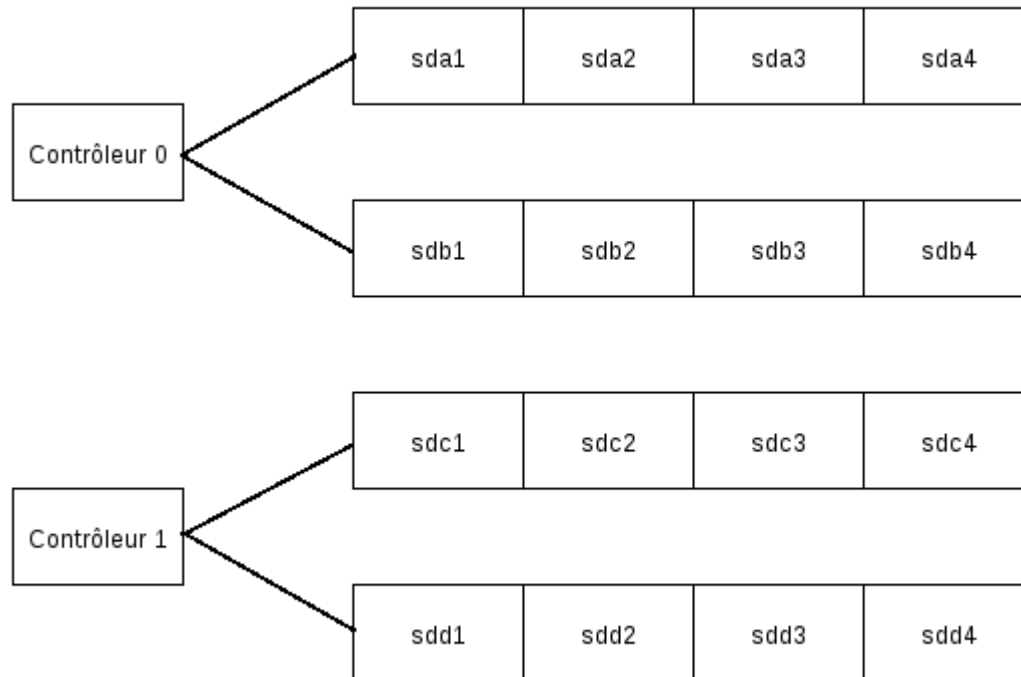
Un disque peut comporter trois types de partitions :

- **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 Etendues**,

- 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

LAB #1 - Partitionnement de votre Disque avec fdisk

Pour procéder au partitionnement de votre disque ou de vos disques, Debian 11 possède l'outil dénommé **fdisk**.

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

```
root@debian11:~# fdisk /dev/sdc
```

```
Welcome to fdisk (util-linux 2.36.1).
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 0x304308a3.
```

```
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
0  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/sdc1	Primaire	100 Mo
/dev/sdc2	Primaire	100 Mo
/dev/sdc3	Primaire	100 Mo
/dev/sdc4	Extended	Du premier secteur disponible au dernier secteur du disque
/dev/sdc5	Logique	500 Mo
/dev/sdc6	Logique	200 Mo
/dev/sdc7	Logique	300 Mo
/dev/sdc8	Logique	500 Mo
/dev/sdc9	Logique	400 Mo
/dev/sdc10	Logique	500 Mo
/dev/sdc11	Logique	500 Mo

Partition	Type	Taille de la Partition
/dev/sdc12	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 (2048-8388607, default 2048):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-8388607, default 8388607): +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 (206848-8388607, default 206848):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (206848-8388607, default 8388607): +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 (411648-8388607, default 411648):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (411648-8388607, default 8388607): +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 (616448-8388607, default 616448):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (616448-8388607, default 8388607):

Created a new partition 4 of type 'Extended' and of size 3.7 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 (618496-8388607, default 618496):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (618496-8388607, default 8388607): +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 (1644544-8388607, default 1644544):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1644544-8388607, default 8388607): +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 (2056192-8388607, default 2056192):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2056192-8388607, default 8388607): +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 (2672640-8388607, default 2672640):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2672640-8388607, default 8388607): +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 (3698688-8388607, default 3698688):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (3698688-8388607, default 8388607): +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 (4519936-8388607, default 4519936):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4519936-8388607, default 8388607): +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 (5545984-8388607, default 5545984):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (5545984-8388607, default 8388607): +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 (6572032-8388607, default 6572032):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (6572032-8388607, default 8388607): +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/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
```

```
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	83	Linux
/dev/sdc6		1644544	2054143	409600	200M	83	Linux
/dev/sdc7		2056192	2670591	614400	300M	83	Linux
/dev/sdc8		2672640	3696639	1024000	500M	83	Linux
/dev/sdc9		3698688	4517887	819200	400M	83	Linux
/dev/sdc10		4519936	5543935	1024000	500M	83	Linux
/dev/sdc11		5545984	6569983	1024000	500M	83	Linux
/dev/sdc12		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@debian11:~# partprobe
Warning: Unable to open /dev/sr0 read-write (Read-only file system). /dev/sr0 has been opened read-only.
```

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

```
root@debian11:~# fdisk /dev/sdc
```

```
Welcome to fdisk (util-linux 2.36.1).
```

```
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/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
```

```
Disk model: QEMU HARDDISK
```

```
Units: sectors of 1 * 512 = 512 bytes
```

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

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: dos
```

```
Disk identifier: 0x304308a3
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	83	Linux
/dev/sdc6		1644544	2054143	409600	200M	83	Linux
/dev/sdc7		2056192	2670591	614400	300M	83	Linux
/dev/sdc8		2672640	3696639	1024000	500M	83	Linux
/dev/sdc9		3698688	4517887	819200	400M	83	Linux
/dev/sdc10		4519936	5543935	1024000	500M	83	Linux
/dev/sdc11		5545984	6569983	1024000	500M	83	Linux
/dev/sdc12		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):
```

```
Partition 12 has been deleted.
```

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

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	83	Linux
/dev/sdc6		1644544	2054143	409600	200M	83	Linux
/dev/sdc7		2056192	2670591	614400	300M	83	Linux
/dev/sdc8		2672640	3696639	1024000	500M	83	Linux
/dev/sdc9		3698688	4517887	819200	400M	83	Linux
/dev/sdc10		4519936	5543935	1024000	500M	83	Linux
/dev/sdc11		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@debian11:~# fdisk /dev/sdc
```

```
Welcome to fdisk (util-linux 2.36.1).
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/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	83	Linux
/dev/sdc6		1644544	2054143	409600	200M	83	Linux
/dev/sdc7		2056192	2670591	614400	300M	83	Linux
/dev/sdc8		2672640	3696639	1024000	500M	83	Linux
/dev/sdc9		3698688	4517887	819200	400M	83	Linux
/dev/sdc10		4519936	5543935	1024000	500M	83	Linux
/dev/sdc11		5545984	6569983	1024000	500M	83	Linux
/dev/sdc12		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/sdc: 4 GiB, 4294967296 bytes, 8388608 sectors

Disk model: QEMU HARDDISK

Units: sectors of 1 * 512 = 512 bytes

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x304308a3
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	fd	Linux raid autodetect
/dev/sdc6		1644544	2054143	409600	200M	8e	Linux LVM
/dev/sdc7		2056192	2670591	614400	300M	8e	Linux LVM
/dev/sdc8		2672640	3696639	1024000	500M	fd	Linux raid autodetect
/dev/sdc9		3698688	4517887	819200	400M	8e	Linux LVM
/dev/sdc10		4519936	5543935	1024000	500M	fd	Linux raid autodetect
/dev/sdc11		5545984	6569983	1024000	500M	fd	Linux raid autodetect
/dev/sdc12		6572032	6981631	409600	200M	83	Linux

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

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

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

```
root@debian11:~# partprobe
Warning: Unable to open /dev/sr0 read-write (Read-only file system). /dev/sr0 has been opened read-only.
```

Options de la Commande fdisk

Les options de cette commande sont :

```
root@debian11:~# fdisk --help
```

Usage:

```
fdisk [options] <disk>          change partition table
fdisk [options] -l [<disk>...] list partition table(s)
```

Display or manipulate a disk partition table.

Options:

```
-b, --sector-size <size>      physical and logical sector size
-B, --protect-boot            don't erase bootbits when creating a new label
-c, --compatibility[=<mode>]   mode is 'dos' or 'nondos' (default)
-L, --color[=<when>]          colorize output (auto, always or never)
                               colors are enabled by default

-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 #2 - Volumes Logiques Linéaires

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

Sous Debian 11, installez le paquet `lvm2` :

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

Nous allons travailler avec les partitions suivantes :

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

Pour initialiser le LVM saisissez la commande suivante :

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

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

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

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

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

Common options for lvm:

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

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

Physical Volume (PV)

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

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

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

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

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

Common options for lvm:

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

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

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

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

```
VG Name
PV Size          300.00 MiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          VAREmm-RCo9-qP7f-0gGP-I6Ym-b494-RXfNjC
"/dev/sdc9" is a new physical volume of "400.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sdc9
VG Name
PV Size          400.00 MiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          Q3Vqjd-rGf1-20vs-S0An-0gY2-eGKr-qmyCY2
```

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

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

pvdiskplay
  [ -a|--all ]
  [ -c|--colon ]
  [ -C|--columns ]
  [ -m|--maps ]
  [ -o|--options String ]
  [ -S|--select String ]
  [ -s|--short ]
  [ -O|--sort String ]
```

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

Common options for lvm:

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

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

Volume Group (VG) et Physical Extent (PE)

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

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

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

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

vgcreate VG_new PV ...
[ -A|--autobackup y|n ]
[ -c|--clustered y|n ]
[ -l|--maxlogicalvolumes Number ]
[ -p|--maxphysicalvolumes Number ]
[ -M|--metadatatype lvm2 ]
[ -s|--physicalextentsize Size[m|UNIT] ]
[ -f|--force ]
[ -Z|--zero y|n ]
[ --addtag Tag ]
[ --alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit ]
[ --metadataprofile String ]
[ --labelsector Number ]
[ --metadatasize Size[m|UNIT] ]
[ --pvmetadatacopies 0|1|2 ]
[ --vgmetadatacopies all|unmanaged|Number ]
[ --reportformat basic|json ]
[ --dataalignment Size[k|UNIT] ]
[ --dataalignmentoffset Size[k|UNIT] ]
[ --shared ]
[ --systemid String ]
```

```
[ --locktype sanlock|dlm|none ]  
[ COMMON_OPTIONS ]
```

Common options for lvm:

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

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

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

```
root@debian11:~# vgdisplay vg0  
--- Volume group ---  
VG Name                vg0  
System ID  
Format                 lvm2  
Metadata Areas         3  
Metadata Sequence No   1  
VG Access              read/write  
VG Status              resizable  
MAX LV                 0  
Cur LV                0
```

Open LV	0
Max PV	0
Cur PV	3
Act PV	3
VG Size	880.00 MiB
PE Size	8.00 MiB
Total PE	110
Alloc PE / Size	0 / 0
Free PE / Size	110 / 880.00 MiB
VG UUID	d7zxKd-eVpl-Zrp1-sN0e-HbyL-zjIV-2xpDua

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

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

vgdisplay
[ -A|--activevolumegroups ]
[ -c|--colon ]
[ -C|--columns ]
[ -o|--options String ]
[ -S|--select String ]
[ -s|--short ]
[ -O|--sort String ]
[ --aligned ]
[ --binary ]
[ --configreport log|vg|lv|pv|pvseg|seg ]
[ --foreign ]
[ --ignorelockingfailure ]
[ --logonly ]
[ --noheadings ]
[ --nosuffix ]
[ --readonly ]
[ --reportformat basic|json ]
```

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

Common options for lvm:

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

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

Logical Volumes (LV)

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

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



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

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

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

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

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

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

Create a raid LV (a specific raid level must be used, e.g. raid1).

```
lvcreate --type raid -L|--size Size[m|UNIT] VG
[ -l|--extents Number[PERCENT] ]
[ -m|--mirrors Number ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ -R|--regionsize Size[m|UNIT] ]
[ --minrecoveryrate Size[k|UNIT] ]
[ --maxrecoveryrate Size[k|UNIT] ]
[ --raidintegrity y|n ]
[ --raidintegritymode String ]
[ --raidintegrityblocksize Number ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

Create a raid10 LV.

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

Create a COW snapshot LV of an origin LV.

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

Create a thin pool.

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

Create a cache pool.

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

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

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

```
[ COMMON_OPTIONS ]
```

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

```
lvcreate -s|--snapshot LV_thin  
[ --type thin ]  
[ --discards passdown|nopassdown|ignore ]  
[ --errorwhenfull y|n ]  
[ COMMON_OPTIONS ]
```

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

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

Create a LV that returns VDO when used.

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

Create a thin LV, first creating a thin pool for it,

where the new thin pool is named by the `--thinpool` arg.

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

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

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

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

```
lvcreate --type cache -L|--size Size[m|UNIT] --cachevol LV VG
```

```
[ -l|--extents Number[PERCENT] ]
[ -c|--chunksize Size[k|UNIT] ]
[ -i|--stripes Number ]
[ -I|--stripesize Size[k|UNIT] ]
[ --cachemode writethrough|writeback|passthrough ]
[ --cachepolicy String ]
[ --cachesettings String ]
[ --cachemetadadataformat auto|1|2 ]
[ COMMON_OPTIONS ]
[ PV ... ]
```

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

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

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

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

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

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

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

Common options for command:

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

```
[ --nosync ]  
[ --noudevsync ]  
[ --reportformat basic|json ]
```

Common options for lvm:

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

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

Créez maintenant un répertoire dans /mnt pour monter lv0 :

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

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

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

```
8193, 24577, 40961, 57345, 73729, 204801, 221185
```

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

Montez votre lv0 :

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

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

Saisissez donc la commande suivante :

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

Constatez ensuite le contenu de /mnt/lvm :

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

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

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

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

LAB #3 - Etendre un Volume Logique à Chaud

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

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



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

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

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

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

Extend an LV by specified PV extents.
lvextend LV PV ...
    [ -r|--resizefs ]
    [ -i|--stripes Number ]
```

```
[ -I|--stripesize Size[k|UNIT] ]  
[ COMMON_OPTIONS ]
```

Extend a pool metadata SubLV by a specified size.

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

Extend an LV according to a predefined policy.

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

Common options for command:

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

Common options for lvm:

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

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

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

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

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

Constatez maintenant la modification de la taille du volume :

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

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

LAB #4 - Snapshots

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

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

```
root@debian11:~# dd if=/dev/zero of=/mnt/lvm/10M bs=1048576 count=10
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.023862 s, 439 MB/s
```

Créez maintenant un snapshot :

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

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

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



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

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

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

lvs
    [ -H|--history ]
```

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

Common options for lvm:

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

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

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

Créez maintenant un répertoire pour monter le snapshot :

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

Montez le snapshot :

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

Comparez le volume d'origine et le snapshot :

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

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

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

Constatez le résultat de cette suppression :

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



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

LAB #5 - Suppression des Volumes

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

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

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

root@debian11:~# umount /mnt/lvm

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



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

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

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

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

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

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

Le Volume Group peut aussi être supprimé :

```
root@debian11:~# vgremove vg0
Volume group "vg0" successfully removed
```

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

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

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

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

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

Ainsi que le volume physique :

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

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

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

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

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

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

LAB #6 - Volumes Logiques en Miroir

Re-créez maintenant votre Volume Group :

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

Créez le VG **vg0** :

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

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

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

Constatez maintenant la présence du miroir :

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

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

Le miroir s'étend sur plusieurs volumes physiques :

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

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

```

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

```

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

```

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

```

```

└─sdc10          8:42    0  500M    0 part
└─sdc11          8:43    0  500M    0 part
└─sdc12          8:44    0  200M    0 part
sr0             11:0     1   378M    0 rom

```

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

```

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

```

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

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

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

Supprimez de nouveau votre miroir :

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

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

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

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

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

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

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

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

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

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

Convert raid LV to change number of stripe images.

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

Convert raid LV to change the stripe size.

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

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

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

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

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

Merge LV images that were split from a raid1 LV.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Convert LV to type thin-pool.

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

Convert LV to type cache-pool.

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

Convert LV to type vdo pool.

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

Detach a cache from an LV.

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

Merge thin LV into its origin LV.

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

Merge COW snapshot LV into its origin.

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

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

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

Replace failed PVs in a raid or mirror LV.

Repair a thin pool.

Repair a cache pool.

```
lvconvert --repair LV_thinpool_cache_cachepool_mirror_raid
```

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

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

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

Poll LV to continue conversion.

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

Add or remove data integrity checksums to raid images.

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

Common options for command:

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

Common options for lvm:

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

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

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

LAB #7 - Modifier les Attributs LVM

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

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



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

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

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





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

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

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



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

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

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

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

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

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

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

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

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

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

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

Resynchronize a mirror or raid LV.

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

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

Resynchronize or check a raid LV.

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

Reconstruct data on specific PVs of a raid LV.

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

Activate or deactivate an LV.

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

Reactivate an LV using the latest metadata.

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

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

Start or stop monitoring an LV from dmeventd.

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

Start or stop processing an LV conversion.

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

Make the minor device number persistent for an LV.

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

Common options for command:

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

Common options for lvm:

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

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

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

LAB #8 - Volumes Logiques en Bandes

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

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

Saisissez donc la commande suivante :

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

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

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

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

```
--- Segments ---
```

```
Logical extents 0 to 13:
```

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

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

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	64G	0	disk	
sdb	8:16	0	32G	0	disk	
└sdb1	8:17	0	31G	0	part	/
└sdb2	8:18	0	1K	0	part	

```

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

```

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

```

root@debian11:~# lvs -o +devices
LV VG Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
lv1 vg0 -wi-a----- 104.00m
lv2 vg0 -wi-a----- 112.00m

```

LAB #9 - Gérer les Métadonnées

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

```

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

```

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

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

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

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

    physical_volumes {

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

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

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

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

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

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

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

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

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

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

        stripes = [
            "pv0", 1
        ]
    }
}

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

    segment1 {
        start_extent = 0
        extent_count = 14          # 112 Megabytes

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

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

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

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

```
root@debian11:~# vgcfgrestore --list vg0
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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](#)

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](#)

Gestion d'Ext3

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

```
root@debian11:~# fdisk -l
Disk /dev/sda: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

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

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sdc1                2048    206847    204800    100M 83 Linux
/dev/sdc2           206848    411647    204800    100M 83 Linux
/dev/sdc3           411648    616447    204800    100M 83 Linux
/dev/sdc4           616448   8388607   7772160    3.7G  5 Extended
/dev/sdc5           618496   1642495   1024000    500M fd Linux raid autodetect
/dev/sdc6          1644544   2054143    409600    200M 8e Linux LVM
```

```
/dev/sdc7      2056192 2670591  614400  300M 8e Linux LVM
/dev/sdc8      2672640 3696639 1024000  500M fd Linux raid autodetect
/dev/sdc9      3698688 4517887  819200  400M 8e Linux LVM
/dev/sdc10     4519936 5543935 1024000  500M fd Linux raid autodetect
/dev/sdc11     5545984 6569983 1024000  500M fd Linux raid autodetect
/dev/sdc12     6572032 6981631  409600  200M 83 Linux
```

Disk /dev/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: 0xf2e3a71a

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1	*	2048	65107967	65105920	31G	83	Linux
/dev/sdb2		65110014	67106815	1996802	975M	5	Extended
/dev/sdb5		65110016	67106815	1996800	975M	82	Linux swap / Solaris

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 ce cas, il s'agit de **/dev/sdc12**.

Créez un filesystem Ext3 sur /dev/sdc12 en utilisant la commande **mke2fs -j** :

```
root@debian11:~# mke2fs -j /dev/sdc12
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
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
```

Les options de la commande **mke2fs** sont :

```
root@debian11:~# mke2fs --help
mke2fs: invalid option -- '-'
Usage: mke2fs [-c|-l filename] [-b block-size] [-C cluster-size]
    [-i bytes-per-inode] [-I inode-size] [-J journal-options]
    [-G flex-group-size] [-N number-of-inodes] [-d root-directory]
    [-m reserved-blocks-percentage] [-o creator-os]
    [-g blocks-per-group] [-L volume-label] [-M last-mounted-directory]
    [-O feature[,...]] [-r fs-revision] [-E extended-option[,...]]
    [-t fs-type] [-T usage-type] [-U UUID] [-e errors_behavior][-z undo_file]
    [-jnvDFSV] device [blocks-count]
```



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.

LAB #10 - Convertir un Système de Fichiers Ext3 en Ext2

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

```
root@debian11:~# dumpe2fs -h /dev/sdc12
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
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:        12095
Free blocks:              192674
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:    256
Filesystem created:        Tue Apr 26 16:41:16 2022
Last mount time:          n/a
```

```
Last write time:      Tue Apr 26 16:41:17 2022
Mount count:         0
Maximum mount count:  -1
Last checked:        Tue Apr 26 16:41:16 2022
Check interval:      0 (<none>)
Reserved blocks uid:  0 (user root)
Reserved blocks gid:  0 (group root)
First inode:         11
Inode size:          128
Journal inode:        8
Default directory hash: half_md4
Directory Hash Seed:  705a3710-674b-4734-bf64-35a6ca7aabc5
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 Ext3 sur cette partition, il convient d'utiliser la commande **tune2fs**

```
root@debian11:~# tune2fs -0 ^has_journal /dev/sdc12
tune2fs 1.46.2 (28-Feb-2021)
```

Constatez le résultat de cette commande :

```
root@debian11:~# dumpe2fs -h /dev/sdc12
```

```
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:    <none>
Last mounted on:          <not available>
Filesystem UUID:          bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
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:        12095
Free blocks:              196787
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:   256
Filesystem created:       Tue Apr 26 16:41:16 2022
Last mount time:          n/a
Last write time:          Tue Apr 26 16:44:45 2022
Mount count:              0
Maximum mount count:      -1
Last checked:             Tue Apr 26 16:41:16 2022
Check interval:           0 (<none>)
Reserved blocks uid:      0 (user root)
```

```
Reserved blocks gid:    0 (group root)
First inode:           11
Inode size:            128
Default directory hash: half_md4
Directory Hash Seed:   705a3710-674b-4734-bf64-35a6ca7aabc5
Journal backup:        inode blocks
```



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

Supprimez maintenant l'inode du journal :

```
root@debian11:~# fsck /dev/sdc12
fsck from util-linux 2.36.1
e2fsck 1.46.2 (28-Feb-2021)
/dev/sdc12: clean, 11/51200 files, 8013/204800 blocks
```

Créez un point de montage pour /dev/sdc12 :

```
root@debian11:~# mkdir /mnt/sdc12
```

Essayez de monter /dev/sdc12 en tant que système de fichiers Ext3. Vous obtiendrez un résultat similaire à celui-ci :

```
root@debian11:~# mount -t ext3 /dev/sdc12 /mnt/sdc12
mount: /mnt/sdc12: wrong fs type, bad option, bad superblock on /dev/sdc12, 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@debian11:~# mkdir /mnt/sdc12
root@debian11:~# mount -t ext3 /dev/sdc12 /mnt/sdc12
mount: /mnt/sdc12: wrong fs type, bad option, bad superblock on /dev/sdc12, missing codepage or helper program,
or other error.
root@debian11:~# mount -t ext2 /dev/sdc12 /mnt/sdc12

root@debian11:~# cat /etc/mtab | grep sdc12
/dev/sdc12 /mnt/sdc12 ext2 rw,relatime 0 0
```

LAB #11 - Convertir un Système de Fichiers Ext2 en Ext3

Pour remplacer le journal sur /dev/sdc12, il convient d'utiliser la commande **tune2fs** :

```
root@debian11:~# umount /mnt/sdc12

root@debian11:~# tune2fs -j /dev/sdc12
tune2fs 1.46.2 (28-Feb-2021)
Creating journal inode: done

root@debian11:~# dumpe2fs -h /dev/sdc12 | head
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:    <none>
Last mounted on:          <not available>
Filesystem UUID:          bdea8b16-8d0b-480c-ab37-68d7d2bdb0b8
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



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

LAB #12 - 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/sdc11 :

```
root@debian11:~# mke2fs -O journal_dev /dev/sdc11
```



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

Créez maintenant un système de fichiers Ext3 sur /dev/sdc12 en plaçant le journal sur /dev/sdc11 :

```
root@debian11:~# mke2fs -j -J device=/dev/sdc11 /dev/sdc12
mke2fs 1.46.2 (28-Feb-2021)
Using journal device's blocksize: 1024
/dev/sdc12 contains a ext3 file system
    last mounted on Tue Apr 26 16:47:45 2022
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 84363ced-42b4-49b1-8627-49c39bd71ac3
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729
```

```
Allocating group tables: done  
Writing inode tables: done  
Adding journal to device /dev/sdc11: done  
Writing superblocks and filesystem accounting information: done
```



Important : Notez que le journal a été placé sur /dev/sdc11 grâce à l'utilisation de l'option **-J**.

LAB #13 - Modifier la Fréquence de Vérification du Système de Fichiers Ext3

Pour modifier la fréquence de vérification du système de fichiers sur /dev/sdc12, il convient d'utiliser soit l'option **-c**, soit l'option **-i** :

```
root@debian11:~# tune2fs -i 100d /dev/sdc12  
tune2fs 1.46.2 (28-Feb-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@debian11:~# dumpe2fs /dev/sdc12 | grep UUID  
dumpe2fs 1.46.2 (28-Feb-2021)  
Filesystem UUID:      84363ced-42b4-49b1-8627-49c39bd71ac3  
Journal UUID:         a9765f56-d8d4-499b-8fc4-db1d665cdfca
```

Ext4

Le système de fichiers **Ext4** fut 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 pébiotets**,
- 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](#).

LAB #14 - Créer un Système de Fichiers Ext4

Créez un système de fichiers Ext4 sur **/dev/sdc11** :

```
root@debian11:~# mkfs.ext4 /dev/sdc11
mke2fs 1.46.2 (28-Feb-2021)
/dev/sdc11 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: a590c318-d9ad-4883-abce-4e154c80e521
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
```

Les options de cette commande sont :

```
root@debian11:~# mkfs.ext4 --help
mkfs.ext4: invalid option -- '-'
Usage: mkfs.ext4 [-c|-l filename] [-b block-size] [-C cluster-size]
      [-i bytes-per-inode] [-I inode-size] [-J journal-options]
      [-G flex-group-size] [-N number-of-inodes] [-d root-directory]
      [-m reserved-blocks-percentage] [-o creator-os]
      [-g blocks-per-group] [-L volume-label] [-M last-mounted-directory]
      [-O feature[,...]] [-r fs-revision] [-E extended-option[,...]]
      [-t fs-type] [-T usage-type ] [-U UUID] [-e errors_behavior][-z undo_file]
      [-jnqvDFSV] device [blocks-count]
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# dumpe2fs /dev/sdc11 | more
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         a590c318-d9ad-4883-abce-4e154c80e521
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:        26670
Free blocks:              485316
Free inodes:              128005
First block:              1
```

```
Block size:          1024
Fragment size:       1024
Group descriptor size: 64
Reserved GDT blocks: 256
--More--
[q]
```

LAB #15 - Ajouter une Etiquette au Système de Fichiers Ext4

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

```
root@debian11:~# e2label /dev/sdc11 my_ext4
root@debian11:~# dumpe2fs /dev/sdc11 | more
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem volume name:   my_ext4
Last mounted on:          <not available>
Filesystem UUID:          a590c318-d9ad-4883-abce-4e154c80e521
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:       26670
Free blocks:              485316
Free inodes:              128005
First block:              1
```

```
Block size:          1024
Fragment size:       1024
Group descriptor size: 64
Reserved GDT blocks: 256
--More--
[q]
```



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

Créez un point de montage dans **/mnt** et essayez de monter **/dev/sdc11** en tant qu'Ext3 :

```
root@debian11:~# mkdir /mnt/sdc11

root@debian11:~# mount -t ext3 /dev/sdc11 /mnt/sdc11
mount: /mnt/sdc11: wrong fs type, bad option, bad superblock on /dev/sdc11, 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@debian11:~# mount /dev/sdc11 /mnt/sdc11

root@debian11:~# mount | grep sdc11
/dev/sdc11 on /mnt/sdc11 type ext4 (rw,relatime)
```



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

LAB #16 - Convertir un Système de Fichiers Ext3 en Ext4

Créez un système de fichiers ext3 sur /dev/sdc12 :

```
root@debian11:~# mkfs.ext3 /dev/sdc12
mke2fs 1.46.2 (28-Feb-2021)
/dev/sdc12 contains a ext3 file system
    created on Wed Apr 27 11:57:30 2022
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 62cb630b-25d5-4ca0-bbab-e855fba96a80
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/sdc12** sur /mnt/sdc12 :

```
root@debian11:~# mount /dev/sdc12 /mnt/sdc12

root@debian11:~# ls -l /mnt/sdc12
total 12
drwx----- 2 root root 12288 Apr 27 12:11 lost+found
```

Créez le fichier **/mnt/sdc12/check_file** :

```
root@debian11:~# touch /mnt/sdc12/check_file
```

Injectez la chaîne **check file** dans le fichier /mnt/sdc12/check_file puis démontez /dev/sdc12 :

```
root@debian11:~# echo "check file" > /mnt/sdc12/check_file  
  
root@debian11:~# umount /mnt/sdc12
```

Exécutez e2fsck sur /dev/sdc12 :

```
root@debian11:~# e2fsck /dev/sdc12  
e2fsck 1.46.2 (28-Feb-2021)  
/dev/sdc12: clean, 12/51200 files, 12127/204800 blocks
```

Convertissez /dev/sdc12 en Ext4 :

```
root@debian11:~# tune2fs -0 extents,uninit_bg,dir_index /dev/sdc12  
tune2fs 1.46.2 (28-Feb-2021)
```

Optimisez le système de fichiers :

```
root@debian11:~# e2fsck -fDC0 /dev/sdc12  
e2fsck 1.46.2 (28-Feb-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/sdc12: ***** FILE SYSTEM WAS MODIFIED *****  
/dev/sdc12: 12/51200 files (0.0% non-contiguous), 12127/204800 blocks
```

Essayez de monter **/dev/sdc12** en tant qu'Ext3 :

```
root@debian11:~# mount -t ext3 /dev/sdc12 /mnt/sdc12  
mount: /mnt/sdc12: wrong fs type, bad option, bad superblock on /dev/sdc12, missing codepage or helper program,  
or other error.
```

Montez /dev/sdc12 sans spécifier le type de fichiers Ext3 et vérifiez le contenu du fichier **check_file** :

```
root@debian11:~# mount /dev/sdc12 /mnt/sdc12

root@debian11:~# ls -l /mnt/sdc12
total 13
-rw-r--r-- 1 root root    11 Apr 27 12:15 check_file
drwx----- 2 root root 12288 Apr 27 12:11 lost+found

root@debian11:~# cat /mnt/sdc12/check_file
check file
```

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

```
root@debian11:~# dumpe2fs /dev/sdc11 | grep UUID
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem UUID:          a590c318-d9ad-4883-abce-4e154c80e521

root@debian11:~# dumpe2fs /dev/sdc12 | grep UUID
dumpe2fs 1.46.2 (28-Feb-2021)
Filesystem UUID:          62cb630b-25d5-4ca0-bbab-e855fba96a80
```

XFS

XFS est un système de fichiers 64-bit journalisé de haute performance créé 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.

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

LAB #17 - Créer un Système de Fichiers XFS

Installez le paquet **xfsprogs** :

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

Créez un système de fichiers XFS sur la partition **/dev/sdc10** :

```
root@debian11:~# mkfs.xfs /dev/sdc10
meta-data=/dev/sdc10          isize=512    agcount=4, agsize=32000 blks
                =               sectsz=512   attr=2, projid32bit=1
                =               crc=1        finobt=1, sparse=1, rmapbt=0
                =               reflink=1     bigtime=0
data        =               bsize=4096     blocks=128000, 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.
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# xfs_info /dev/sdc10
meta-data=/dev/sdc10          isize=512    agcount=4, agsize=32000 blks
                =               sectsz=512   attr=2, projid32bit=1
                =               crc=1        finobt=1, sparse=1, rmapbt=0
                =               reflink=1     bigtime=0
data        =               bsize=4096     blocks=128000, 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
```

LAB #18 - Ajouter une Etiquette au Système de Fichiers XFS

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

```
root@debian11:~# xfs_admin -L my_xfs /dev/sdc10
writing all SBs
new label = "my_xfs"
```

Pour voir l'étiquette, utilisez la commande suivante :

```
root@debian11:~# xfs_admin -l /dev/sdc10
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@debian11:~# xfs_admin -u /dev/sdc10
UUID = b4d2c2c3-94bc-4408-8284-5b09ab60204e
```



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.

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](#).

Avant de continuer, installer le paquet **reiserfsprogs** :

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

LAB #19 - Créer un Système de Fichiers ReiserFS

Créez un système de fichiers ReiserFS sur la partition **/dev/sdc8** :

```
root@debian11:~# mkfs.reiserfs /dev/sdc8
mkfs.reiserfs 3.6.27

Guessing about desired format.. Kernel 5.10.0-13-amd64 is running.
Format 3.6 with standard journal
Count of blocks on the device: 128000
Number of blocks consumed by mkreiserfs formatting process: 8215
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 95c81e34-2215-4b96-be47-e5368444f19a
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
          ALL DATA WILL BE LOST ON '/dev/sdc8'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
```

ReiserFS is successfully created on /dev/sdc8.

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# debugreiserfs /dev/sdc8
debugreiserfs 3.6.27

Filesystem state: consistent

Reiserfs super block in block 16 on 0x828 of format 3.6 with standard journal
Count of blocks on the device: 128000
Number of bitmaps: 4
Blocksize: 4096
Free blocks (count of blocks - used [journal, bitmaps, data, reserved] blocks): 119785
Root block: 8211
Filesystem is clean
Tree height: 2
Hash function used to sort names: "r5"
Objectid map size 2, max 972
Journal parameters:
    Device [0x0]
    Magic [0x345d673e]
    Size 8193 blocks (including 1 for journal header) (first block 18)
    Max transaction length 1024 blocks
    Max batch size 900 blocks
    Max commit age 30
Blocks reserved by journal: 0
Fs state field: 0x0:
sb_version: 2
inode generation number: 0
UUID: 95c81e34-2215-4b96-be47-e5368444f19a
LABEL:
Set flags in SB:
```

ATTRIBUTES CLEAN

Mount count: 1

Maximum mount count: 30

Last fsck run: Wed Apr 27 13:37:02 2022

Check interval in days: 180

LAB #20 - Ajouter une Etiquette au Système de Fichiers ReiserFS

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

```
root@debian11:~# reiserfstune -l my_reiserfs /dev/sdc8
reiserfstune: Journal device has not been specified. Assuming journal is on the main device (/dev/sdc8).

Current parameters:

Filesystem state: consistent

Reiserfs super block in block 16 on 0x828 of format 3.6 with standard journal
Count of blocks on the device: 128000
Number of bitmaps: 4
Blocksize: 4096
Free blocks (count of blocks - used [journal, bitmaps, data, reserved] blocks): 119785
Root block: 8211
Filesystem is clean
Tree height: 2
Hash function used to sort names: "r5"
Objectid map size 2, max 972
Journal parameters:
    Device [0x0]
    Magic [0x345d673e]
    Size 8193 blocks (including 1 for journal header) (first block 18)
    Max transaction length 1024 blocks
    Max batch size 900 blocks
```

```
Max commit age 30
Blocks reserved by journal: 0
Fs state field: 0x0:
sb_version: 2
inode generation number: 0
UUID: 95c81e34-2215-4b96-be47-e5368444f19a
LABEL: my_reiserfs
Set flags in SB:
    ATTRIBUTES CLEAN
Mount count: 1
Maximum mount count: 30
Last fsck run: Wed Apr 27 13:37:02 2022
Check interval in days: 180
```



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

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

```
root@debian11:~# debugreiserfs /dev/sdc8 | grep UUID
debugreiserfs 3.6.27

UUID: 95c81e34-2215-4b96-be47-e5368444f19a
```

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](#).

Avant de continuer, installer le paquet **jfsutils** :

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

LAB #21 - Créer un Système de Fichiers JFS

Créez un système de fichiers JFS sur **/dev/sdc5** :

```
root@debian11:~# mkfs.jfs /dev/sdc5
mkfs.jfs version 1.1.15, 04-Mar-2011
Warning! All data on device /dev/sdc5 will be lost!

Continue? (Y/N) y
|

Format completed successfully.

512000 kilobytes total disk space.
```

Consultez maintenant les caractéristiques du système de fichier :

```
root@debian11:~# jfs_tune -l /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011

JFS filesystem superblock:

JFS magic number:      'JFS1'
JFS version:           1
JFS state:              clean
JFS flags:              JFS_LINUX  JFS_COMMIT  JFS_GROUPCOMMIT  JFS_INLINELOG
Aggregate block size:  4096 bytes
Aggregate size:         1019464 blocks
Physical block size:    512 bytes
Allocation group size:  8192 aggregate blocks
```

```
Log device number:    0x0
Filesystem creation:  Wed Apr 27 13:48:33 2022
Volume label:         ''
```

LAB #22 - Ajouter une Etiquette au Système de Fichiers JFS

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

```
root@debian11:~# jfs_tune -L my_jfs /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011
Volume label updated successfully.

root@debian11:~# jfs_tune -l /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011

JFS filesystem superblock:

JFS magic number:      'JFS1'
JFS version:           1
JFS state:             clean
JFS flags:             JFS_LINUX  JFS_COMMIT  JFS_GROUPCOMMIT  JFS_INLINELOG
Aggregate block size:  4096 bytes
Aggregate size:        1019464 blocks
Physical block size:   512 bytes
Allocation group size: 8192 aggregate blocks
Log device number:     0x0
Filesystem creation:   Wed Apr 27 13:48:33 2022
Volume label:          'my_jfs'
```



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

Créez maintenant un UUID pour le système de fichiers :

```
root@debian11:~# jfs_tune -U random /dev/sdc5
jfs_tune version 1.1.15, 04-Mar-2011
UUID updated successfully.
```

Dernièrement, pour visualiser l'UUID du système de fichiers, utilisez les commandes **jfs_tune** et **grep** :

```
root@debian11:~# jfs_tune -l /dev/sdc5 | grep UUID
File system UUID:      9fc2208f-9b6f-400a-8b0f-4b584f3edaaa
External log UUID:     00000000-0000-0000-d01e-21e2fe7f0000
```

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.

Avant de continuer, installer le paquet **btrfs-progs** :

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

LAB #23 - Créer un Système de Fichiers Btrfs

Créez un système de fichiers Btrfs sur **/dev/sdc5** :

```
root@debian11:~# mkfs.btrfs /dev/sdc5
btrfs-progs v5.10.1
See http://btrfs.wiki.kernel.org for more information.

/dev/sdc5 appears to contain an existing filesystem (jfs).
ERROR: use the -f option to force overwrite of /dev/sdc5

root@debian11:~# mkfs.btrfs -f /dev/sdc5
btrfs-progs v5.10.1
See http://btrfs.wiki.kernel.org for more information.

Label:                (null)
UUID:                 6030b3a7-a3f0-4af8-b00e-988e43c91784
Node size:            16384
Sector size:          4096
Filesystem size:      500.00MiB
Block group profiles:
  Data:                single           8.00MiB
  Metadata:            DUP              32.00MiB
  System:              DUP              8.00MiB
SSD detected:         no
Incompat features:    extref, skinny-metadata
Runtime features:
Checksum:             crc32c
Number of devices:    1
Devices:
  ID      SIZE  PATH
  1      500.00MiB /dev/sdc5
```

Montez la partition btrfs sur /mnt :

```
root@debian11:~# mount -t btrfs /dev/sdc5 /mnt/  
root@debian11:~# mount | grep btrfs  
/dev/sdc5 on /mnt type btrfs (rw,relatime,space_cache,subvolid=5,subvol=/)
```

Sous Btrfs, il est possible de créer des sous volumes. Pour comprendre, comparez notre partition Btrfs à un VG et des sous volumes comme des LV :

```
root@debian11:~# btrfs subvolume create /mnt/volume1  
Create subvolume '/mnt/volume1'  
  
root@debian11:~# btrfs subvolume list /mnt/  
ID 256 gen 7 top level 5 path volume1
```



Important - L'ID identifie le volume d'une manière unique.

LAB #24 - Convertir un Système de Fichiers Ext4 en Btrfs

Créez un système de fichiers Ext4 sur **/dev/sdc10** :

```
root@debian11:~# mkfs.ext4 -b 4096 /dev/sdc10  
mke2fs 1.46.2 (28-Feb-2021)  
/dev/sdc10 contains a ext4 file system  
    created on Thu Apr 28 05:36:22 2022  
Proceed anyway? (y,N) y  
Discarding device blocks: done  
Creating filesystem with 128000 4k blocks and 128000 inodes  
Filesystem UUID: 592f92f2-9d1b-40bf-89b3-24b35b59621b  
Superblock backups stored on blocks:
```

32768, 98304

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

Convertissez maintenant le système de fichiers Ext4 en Btrfs :

```
root@debian11:~# btrfs-convert /dev/sdc10
create btrfs filesystem:
    blocksize: 4096
    nodesize: 16384
    features:  extref, skinny-metadata (default)
    checksum:  crc32c
free space report:
    total:      524288000
    free:      390004736 (74.39%)
creating ext2 image file
creating btrfs metadata
copy inodes [0] [          0/          11]
conversion complete
```

Visualisez les systèmes de fichiers Btrfs :

```
root@debian11:~# btrfs filesystem show
Label: none  uuid: 6030b3a7-a3f0-4af8-b00e-988e43c91784
    Total devices 1 FS bytes used 144.00KiB
    devid    1 size 500.00MiB used 88.00MiB path /dev/sdc5

Label: none  uuid: ebdf4e62-3dc5-4bd3-8dc2-988ad5e680e8
    Total devices 1 FS bytes used 32.62MiB
    devid    1 size 500.00MiB used 163.00MiB path /dev/sdc10
```

Ajouter la partition /dev/sdc10 à /mnt :

```
root@debian11:~# btrfs device add /dev/sdc10 /mnt/  
/dev/sdc10 appears to contain an existing filesystem (btrfs).
```

ERROR: use the -f option to force overwrite of /dev/sdc10

```
root@debian11:~# btrfs device add -f /dev/sdc10 /mnt/
```

```
root@debian11:~# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	1.9G	0	1.9G	0%	/dev
tmpfs	394M	904K	393M	1%	/run
/dev/sdb1	31G	3.5G	26G	13%	/
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	394M	44K	394M	1%	/run/user/113
tmpfs	394M	40K	394M	1%	/run/user/1000
/dev/sdc5	1000M	3.6M	918M	1%	/mnt

```
root@debian11:~# btrfs filesystem show
```

```
Label: none  uuid: 6030b3a7-a3f0-4af8-b00e-988e43c91784  
    Total devices 2 FS bytes used 144.00KiB  
    devid    1 size 500.00MiB used 88.00MiB path /dev/sdc5  
    devid    2 size 500.00MiB used 0.00B path /dev/sdc10
```

```
root@debian11:~# btrfs filesystem df /mnt/
```

```
Data, single: total=8.00MiB, used=0.00B  
System, DUP: total=8.00MiB, used=16.00KiB  
Metadata, DUP: total=32.00MiB, used=144.00KiB  
GlobalReserve, single: total=3.25MiB, used=0.00B
```

LAB #25 - Créer un Snapshot

Créez un snapshot du volume /mnt :

```
root@debian11:~# btrfs subvolume snapshot /mnt /mnt/snapshot
Create a snapshot of '/mnt' in '/mnt/snapshot'

root@debian11:~# ls -l /mnt
total 16
drwxr-xr-x 1 root root 14 Apr 28 05:31 snapshot
drwxr-xr-x 1 root root  0 Apr 28 05:33 volume1
```

Démontez le système de fichiers btrfs :

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

Montez ensuite le snapshot :

```
root@debian11:~# mount -o subvol=snapshot /dev/sdc5 /mnt/

root@debian11:~# ls -l /mnt
total 0
drwxr-xr-x 1 root root 0 Apr 28 05:46 volume1

root@debian11:~# btrfs filesystem df /mnt/
Data, single: total=8.00MiB, used=0.00B
System, DUP: total=8.00MiB, used=16.00KiB
Metadata, DUP: total=32.00MiB, used=144.00KiB
GlobalReserve, single: total=3.25MiB, used=0.00B
```

LAB #26 - Ajouter une Etiquette au Système de Fichiers Btrfs

Pour voir les caractéristiques du système de fichiers Btrfs, utilisez la commande suivante :

```
root@debian11:~# btrfs inspect-internal dump-super /dev/sdc5
superblock: bytenr=65536, device=/dev/sdc5
-----
csum_type          0 (crc32c)
csum_size          4
csum               0xc63d381f [match]
bytenr             65536
flags              0x1
                  ( WRITTEN )
magic              _BHRfS_M [match]
fsid               6030b3a7-a3f0-4af8-b00e-988e43c91784
metadata_uuid      6030b3a7-a3f0-4af8-b00e-988e43c91784
label
generation        12
root               30441472
sys_array_size     129
chunk_root_generation 8
root_level         0
chunk_root         22020096
chunk_root_level   0
log_root           0
log_root_transid   0
log_root_level     0
total_bytes        1048576000
bytes_used         163840
sectorsize         4096
nodesize           16384
leafsize (deprecated) 16384
stripesize         4096
```

```
root_dir          6
num_devices       2
compat_flags      0x0
compat_ro_flags   0x0
incompat_flags    0x141
                  ( MIXED_BACKREF |
                    EXTENDED_IREF |
                    SKINNY_METADATA )
cache_generation  12
uuid_tree_generation 12
dev_item.uuid      ee13b782-2ecb-443a-8703-b44206bab1e3
dev_item.fsid      6030b3a7-a3f0-4af8-b00e-988e43c91784 [match]
dev_item.type      0
dev_item.total_bytes 524288000
dev_item.bytes_used 92274688
dev_item.io_align  4096
dev_item.io_width   4096
dev_item.sector_size 4096
dev_item.devid      1
dev_item.dev_group  0
dev_item.seek_speed 0
dev_item.bandwidth  0
dev_item.generation 0
```

Créez une étiquette avec la commande **btrfs filesystem label** :

```
root@debian11:~# btrfs filesystem label /mnt my_btrfs

root@debian11:~# btrfs inspect-internal dump-super /dev/sdc5
superblock: bytenr=65536, device=/dev/sdc5
-----
csum_type          0 (crc32c)
csum_size          4
csum               0x981c9953 [match]
```

bytenr	65536
flags	0x1 (WRITTEN)
magic	_BHRfS_M [match]
fsid	6030b3a7-a3f0-4af8-b00e-988e43c91784
metadata_uuid	6030b3a7-a3f0-4af8-b00e-988e43c91784
label	my_btrfs
generation	13
root	30523392
sys_array_size	129
chunk_root_generation	8
root_level	0
chunk_root	22020096
chunk_root_level	0
log_root	0
log_root_transid	0
log_root_level	0
total_bytes	1048576000
bytes_used	163840
sectorsize	4096
nodesize	16384
leafsize (deprecated)	16384
stripesize	4096
root_dir	6
num_devices	2
compat_flags	0x0
compat_ro_flags	0x0
incompat_flags	0x141 (MIXED_BACKREF EXTENDED_IREF SKINNY_METADATA)
cache_generation	13
uuid_tree_generation	13
dev_item.uuid	ee13b782-2ecb-443a-8703-b44206bab1e3

```

dev_item.fsid      6030b3a7-a3f0-4af8-b00e-988e43c91784 [match]
dev_item.type      0
dev_item.total_bytes 524288000
dev_item.bytes_used 92274688
dev_item.io_align  4096
dev_item.io_width  4096
dev_item.sector_size 4096
dev_item.devid     1
dev_item.dev_group  0
dev_item.seek_speed 0
dev_item.bandwidth  0
dev_item.generation 0

```

Démontez le système de fichiers btrfs :

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

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

Systèmes de Fichiers Chiffrés

LAB #27 - Créer un Système de Fichiers Chiffré avec encryptfs

Commencez par installer le paquet **ecryptfs-utils** dans la machine virtuelle Debian 11 :

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

Remontez /mnt/sdc12 sur lui-même en spécifiant le type de fichiers en tant qu'encryptfs :

```
root@debian11:~# mount -t ecryptfs /mnt/sdc12 /mnt/sdc12
Select key type to use for newly created files:
 1) tspi
 2) passphrase
Selection: 2
Passphrase:
Select cipher:
 1) aes: blocksize = 16; min keysize = 16; max keysize = 32
 2) blowfish: blocksize = 8; min keysize = 16; max keysize = 56
 3) des3_ede: blocksize = 8; min keysize = 24; max keysize = 24
 4) twofish: blocksize = 16; min keysize = 16; max keysize = 32
 5) cast6: blocksize = 16; min keysize = 16; max keysize = 32
 6) cast5: blocksize = 8; min keysize = 5; max keysize = 16
Selection [aes]: 1
Select key bytes:
 1) 16
 2) 32
 3) 24
Selection [16]: 1
Enable plaintext passthrough (y/n) [n]: n
Enable filename encryption (y/n) [n]: y
Filename Encryption Key (FNEK) Signature [91aefde99b5a4977]:
```

Attempting to mount with the following options:

```
ecryptfs_unlink_sigs
ecryptfs_fnek_sig=91aefde99b5a4977
ecryptfs_key_bytes=16
ecryptfs_cipher=aes
ecryptfs_sig=91aefde99b5a4977
```

WARNING: Based on the contents of [/root/.ecryptfs/sig-cache.txt], it looks like you have never mounted with this key before. This could mean that you have typed your passphrase wrong.

Would you like to proceed with the mount (yes/no)? : yes

Would you like to append sig [91aefde99b5a4977] to
[/root/.ecryptfs/sig-cache.txt]

in order to avoid this warning in the future (yes/no)? : yes

Successfully appended new sig to user sig cache file

Mounted eCryptfs

Ce montage est visible dans la sortie de la commande **df** :

```
root@debian11:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0   1.9G   0% /dev
tmpfs           394M  904K   393M   1% /run
/dev/sdb1       31G   3.5G    26G  13% /
tmpfs           2.0G     0   2.0G   0% /dev/shm
tmpfs           5.0M     0   5.0M   0% /run/lock
tmpfs          394M   44K   394M   1% /run/user/113
/dev/sdc11      474M   14K   445M   1% /mnt/sdc11
/mnt/sdc12      189M   32K   175M   1% /mnt/sdc12
tmpfs          394M   40K   394M   1% /run/user/1000
```

Plus de détails sont visibles avec la commande **mount** :

```
root@debian11:~# mount | tail
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/user/113 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=113,gid=121)
/dev/sdc11 on /mnt/sdc11 type ext4 (rw,relatime)
/dev/sdc12 on /mnt/sdc12 type ext4 (rw,relatime)
tmpfs on /run/user/1000 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=1000,gid=1000)
/mnt/sdc12 on /mnt/sdc12 type ecryptfs
(rw,relatime,ecryptfs_fnek_sig=91aefde99b5a4977,ecryptfs_sig=91aefde99b5a4977,ecryptfs_cipher=aes,ecryptfs_key_by
tes=16,ecryptfs_unlink_sigs)
```

Créez maintenant le fichier **encrypt** contenant la chaîne de caractères **fenestros** dans /mnt/sdc12 :

```
root@debian11:~# touch /mnt/sdc12/encrypt

root@debian11:~# echo "fenestros" > /mnt/sdc12/encrypt

root@debian11:~# cat /mnt/sdc12/encrypt
fenestros
```

Démontez maintenant /mnt/sdc12 :

```
root@debian11:~# umount /mnt/sdc12

root@debian11:~# mount | tail
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
```

```
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/user/113 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=113,gid=121)
/dev/sdc11 on /mnt/sdc11 type ext4 (rw,relatime)
/dev/sdc12 on /mnt/sdc12 type ext4 (rw,relatime)
tmpfs on /run/user/1000 type tmpfs
(rw,nosuid,nodev,relatime,size=402556k,nr_inodes=100639,mode=700,uid=1000,gid=1000)
```



Important : Notez que /dev/sdc12 est toujours monté sur /mnt/sdc12.

Constatez maintenant le contenu de /mnt/sdc12 :

```
root@debian11:~# ls -l /mnt/sdc12
total 25
-rw-r--r-- 1 root root    11 Apr 27 12:15 check_file
-rw-r--r-- 1 root root 12288 Apr 28 06:12
ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE- -
drwx----- 2 root root 12288 Apr 27 12:11 lost+found
```



Important : Notez que le nom du fichier **encrypt** a été chiffré.

Constatez maintenant le contenu du fichier

ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE- :

```
root@debian11:~# cat
/mnt/sdc12/ECRYPTFS_FNEK_ENCRYPTED.FWaFfjrdapd7RkRCT3OFIdaE.-6mxquDYm8R4p7VFuV0SGgSJauuQJ2hEE- -
```

KnwM
"3DUfw`6.X0XgR#2 _CONSOLEZIw#t+X
qF3|2m&69CHlú)Tn%A^cu[kL;
N h'
5uxΠRNbb&<[RnIpz,0nKjC\y□eZ5qmE:K:<\rFθUY9^E|hW!|0yd\R^B*/h5)Q□8b PjQ]z[l0q
bhXnM`rAgM}{xc\DçYMuŌ=3u|ivcWV3P6<#F#[<TP&ün;6pF-0!l}uz\$zZENCriFW#LMTVj3
?9xdQp\$X.x`?L"4;5sQf0|%V5□n"DŦ!*MfsPU9R'(eq^𐄂 NU
□e|\Z._NIBik1>UGD.S'}\$9 hdc-`IDX}Qf0\Gv w-Cdfj{+K%m T=pT)( H&d`\/yb1}
FB}ff(JfĀ,~"-0u□4
JIOI/2\H[pi
)B亭v]?~p4=x\0F? *1)
v/RD+□{GB8\IGiw□
uAc~anv,
2□|tA=\{5+4_z+oF L>Xx\$*eVCb p(
L&a0Y'\$_P\b;d}7r(`'+bX/0Kmm5-<,3[RkmWGuLOPΦUG_!34\$ia* U %}(σQ2 B*↘I\$.@s?![o-
}S{=,k2k]κ8dhn@s%? Tz̄,k]jqc
x;F;\$Z0#+0Sd8f\YBc3VG蕞IAq
>BA4XGl fF\$2scB4lo#0Z+yCEZ^_
(@`' 'i\B~[[K.ohc6Qw/| y>8ok
L
M+LMw e>發nb>ryΛpB(i□5VY#DA□?z)òpr(
'3c}%71k !F_Ю`İFs!]LD
p/&8`{?H<UI<|1xU2f3F𠂇1KM𠂇2P)鵯;ي}U?'T9UP@wp娛lkzLlb<1(ux
*a+n
]HDu0l:#]`]^Bÿ3LA:euGqHᄇ.g致rp湮8N!e)}Gs["aS)bX.Ny;6\$□óymMR逦+
𐄂pk#D>' %'A<8뽕/ph`kF{-Noh+Ç+~;pqoqE]d
)m)'_t}aI*Qh%so?%BBo40M7&vEBF9i*aApjm8|=QlwmWtz`#χ~3(=eR3SfU @u2
n
BD^DoskY}N8S7ᄃ3^b]1W-?DqQarpv~>!hIkXá@Cow&XJ<Hᄂ(uΓu{c/md`S&.GFp7euk)0I!-(!%eL0F
x]Ç@0(خ

```

60噓      )G*
m^xveřfo|â'wn^^b[H□_Bg+&;~b      Zvn,Y□4Z2<.}´-sTUn k5(1r\rvf_n{fЩ
                                                    $f6YWE□K4m0|c$!y
                                                    /Y_os3G{zp_+~)/z@h1!vn"h

*ł+uyBex{p%|V)EauM(N mJ=%XUx-bScF&
6q@dc!B□LG)b6tT6'hD4E8-mĤ
d(<v!RX□3d%~Ija4zb)Q'8j#K9□r+9?,H!q4
:U□X*d
KmPem&btzesHc^J91Fplq4CL\ĸF      sIwĤ@*:H& {K.rT7M9<i
q6fuSf93      □a7qP[TrR_V0,Y n9 i
                        pe5
kf,`F'+}VezEuBv,_4 ~g/M90m>⊗ L5□,i)□c%4~Δa-*au`ř0_~;[9"n]m#3PrY~(S3:J66fu%DA|wWFI%r{EŘ(7TK
                                                    eI\1Nx^}
G9+|NhKp-hđ-dJh<F
莛J_FIc|t.],uTłf3vc?4Zy,zBl(]^uZp5"gsBii/:aHak[$kx@6- Ps.@
_4QĴ$Nn:``Ǿc#M□
      73F5
      3ł:9`Qg𐄂3 GwTroot@debian11:~#

```



Important : Notez que le contenu du fichier **encrypt** a été chiffré. Pour pouvoir lire le nom et le contenu de ce fichier de nouveau, il faut remonter /mnt/sdc12 en spécifiant les mêmes options ainsi que la même passphrase. Notez que si vous vous trompez au niveau de la passphrase ceci n'empêchera pas le processus de montage. Par contre vous ne pourrez ni lire le nom ni lire le contenu du fichier chiffré.

LAB #28 - Créer un Système de Fichiers Chiffré avec LUKS

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 des mots de passe multiples, afin que plusieurs utilisateurs soient en mesure de déchiffrer le même volume sans partager leur mot de passe.

Mise en Place

Commencez par installer le paquet **cryptsetup** :

```
root@debian11:/# apt-get -y install cryptsetup
```

Remplissez la partition /dev/sdc11 avec des données aléatoires :

```
root@debian11:~# shred -v --iterations=1 /dev/sdc11
shred: /dev/sdc11: pass 1/1 (random)...
shred: /dev/sdc11: pass 1/1 (random)...474MiB/500MiB 94%
shred: /dev/sdc11: pass 1/1 (random)...500MiB/500MiB 100%
```



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

Démontez /dev/sdc11 :

```
root@debian11:/# umount /mnt/sdc11
```

Initialisez la partition avec LUKS :

```
root@debian11:~# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdc11
```

WARNING!

=====

```
This will overwrite data on /dev/sdc11 irrevocably.
```

```
Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/sdc11: fenestros123456789
Verify passphrase: fenestros123456789
Key slot 0 created.
Command successful.
```



Important : La passphrase ne sera pas en claire. Elle l'est ici pour vous montrer un mot de passe acceptable pour LUKS.

Ouvrez la partition LUKS en lui donnant le nom **sdc11** :

```
root@debian11:~# cryptsetup luksOpen /dev/sdc11 sdc11
Enter passphrase for /dev/sdc11: fenestros123456789
```

Vérifiez que le système voit la partition :

```
root@debian11:~# ls -l /dev/mapper | grep sdc11
lrwxrwxrwx 1 root root      7 Apr 28 06:26 sdc11 -> ../dm-2
```

Créez maintenant un système de fichiers sur **/dev/mapper/sdc11** :

```
root@debian11:~# mkfs.ext4 /dev/mapper/sdc11
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 495616 1k blocks and 123952 inodes
Filesystem UUID: cc35e698-6915-4cb3-a0ca-e76bce705be1
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
```

Montez la partition LUKS :

```
root@debian11:/# mount /dev/mapper/sdc11 /mnt/sdc11
```

Vérifiez la présence du montage :

```
root@debian11:~# df -h | grep sdc11
/dev/mapper/sdc11 459M 14K 431M 1% /mnt/sdc11
```

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

```
root@debian11:~# vi /etc/crypttab

root@debian11:~# cat /etc/crypttab
# <target name> <source device>          <key file>          <options>
sdc11 /dev/sdc11 none
```

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

```
root@debian11:~# vi /etc/fstab

root@debian11:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
```

```
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=1f9439f5-4b19-49b1-b292-60c2c674cee9 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/mapper/sdc11 /mnt/sdc11 ext4 defaults
1 2
```

Ajouter une deuxième Passphrase

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

```
root@debian11:~# cryptsetup luksAddKey /dev/sdc11
Enter any existing passphrase: fenestros123456789
Enter new passphrase for key slot: debian123456789
Verify passphrase: debian123456789
```



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@debian11:~# cryptsetup luksRemoveKey /dev/sdc11
Enter passphrase to be deleted: debian123456789
```





NE PAS FAIRE : Lors du démarrage de la machine virtuelle, le système vous demandera d'entrer la passphrase **fenestros123456789** pour permettre le montage de `/dev/sdc11`.

RAID Logiciel

LAB #29 - Mise en Place du RAID 5 Logiciel

29.1 - Préparer le disque

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

```
root@debian11:~# fdisk -l
Disk /dev/sda: 32 GiB, 34359738368 bytes, 67108864 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf2e3a71a

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

Disk /dev/sdb: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
```

Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

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

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		2048	206847	204800	100M	83	Linux
/dev/sdc2		206848	411647	204800	100M	83	Linux
/dev/sdc3		411648	616447	204800	100M	83	Linux
/dev/sdc4		616448	8388607	7772160	3.7G	5	Extended
/dev/sdc5		618496	1642495	1024000	500M	fd	Linux raid autodetect
/dev/sdc6		1644544	2054143	409600	200M	8e	Linux LVM
/dev/sdc7		2056192	2670591	614400	300M	8e	Linux LVM
/dev/sdc8		2672640	3696639	1024000	500M	fd	Linux raid autodetect
/dev/sdc9		3698688	4517887	819200	400M	8e	Linux LVM
/dev/sdc10		4519936	5543935	1024000	500M	fd	Linux raid autodetect
/dev/sdc11		5545984	6569983	1024000	500M	fd	Linux raid autodetect
/dev/sdc12		6572032	6981631	409600	200M	83	Linux

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

Disk /dev/mapper/vg0-lv2: 112 MiB, 117440512 bytes, 229376 sectors

```
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 65536 bytes / 131072 bytes
```

```
Disk /dev/mapper/sdc11: 484 MiB, 507510784 bytes, 991232 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

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

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

29.2 - Créer une Unité RAID

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

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

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

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

Saisissez maintenant la commande suivante :

```
root@debian11:~# mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sdc5 /dev/sdc8 /dev/sdc10
mdadm: /dev/sdc8 appears to contain a reiserfs file system
size = 512000K
```

```
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
```

Les options dans la ligne de commande sont :

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

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

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

Some common options are:
  --help      -h      : General help message or, after above option,
                        mode specific help message
  --help-options : This help message
  --version   -V      : Print version information for mdadm
  --verbose   -v      : Be more verbose about what is happening
  --quiet     -q      : Don't print un-necessary messages
  --brief     -b      : Be less verbose, more brief
  --export    -Y      : With --detail, --detail-platform or --examine use
                        key=value format for easy import into environment
  --force     -f      : Override normal checks and be more forceful

  --assemble -A      : Assemble an array
  --build     -B      : Build an array without metadata
  --create    -C      : Create a new array
  --detail    -D      : Display details of an array
```

```
--examine      -E    : Examine superblock on an array component
--examine-bitmap -X    : Display the detail of a bitmap file
--examine-badblocks: Display list of known bad blocks on device
--monitor      -F    : monitor (follow) some arrays
--grow         -G    : resize/ reshape and array
--incremental  -I    : add/remove a single device to/from an array as appropriate
--query        -Q    : Display general information about how a
                        device relates to the md driver
--auto-detect   : Start arrays auto-detected by the kernel
```

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

```
root@debian11:~# mdadm --help
mdadm is used for building, managing, and monitoring
Linux md devices (aka RAID arrays)
Usage: mdadm --create device options...
        Create a new array from unused devices.
mdadm --assemble device options...
        Assemble a previously created array.
mdadm --build device options...
        Create or assemble an array without metadata.
mdadm --manage device options...
        make changes to an existing array.
mdadm --misc options... devices
        report on or modify various md related devices.
mdadm --grow options device
        resize/reshape an active array
mdadm --incremental device
        add/remove a device to/from an array as appropriate
mdadm --monitor options...
        Monitor one or more array for significant changes.
mdadm device options...
        Shorthand for --manage.
Any parameter that does not start with '-' is treated as a device name
```

or, for `--examine-bitmap`, a file name.

The first such name is often the name of an md device. Subsequent names are often names of component devices.

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

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

For general help on options use
`mdadm --help-options`

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

```
root@debian11:~# mdadm --create --help
```

```
Usage: mdadm --create device --chunk=X --level=Y --raid-devices=Z devices
```

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

Before devices are added, they are checked to see if they already contain raid superblocks or filesystems. They are also checked to see if the variance in device size exceeds 1%.

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

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

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

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

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

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

```
root@debian8:~# mdadm --help
mdadm is used for building, managing, and monitoring
Linux md devices (aka RAID arrays)
Usage: mdadm --create device options...
        Create a new array from unused devices.
mdadm --assemble device options...
        Assemble a previously created array.
mdadm --build device options...
```

```
Create or assemble an array without metadata.
mdadm --manage device options...
    make changes to an existing array.
mdadm --misc options... devices
    report on or modify various md related devices.
mdadm --grow options device
    resize/reshape an active array
mdadm --incremental device
    add/remove a device to/from an array as appropriate
mdadm --monitor options...
    Monitor one or more array for significant changes.
mdadm device options...
    Shorthand for --manage.
```

Any parameter that does not start with '-' is treated as a device name or, for --examine-bitmap, a file name.

The first such name is often the name of an md device. Subsequent names are often names of component devices.

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

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

For general help on options use

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

Constatez maintenant les informations concernant le RAID 5 créé :

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

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

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

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

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

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


    Layout : left-symmetric
    Chunk Size : 512K

Consistency Policy : resync


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


   Number   Major   Minor   RaidDevice State
    0         8       37         0     active sync   /dev/sdc5
```

1	8	40	1	active sync	/dev/sdc8
3	8	42	2	active sync	/dev/sdc10

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

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

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

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

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

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

# definitions of existing MD arrays

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

Ecrasez le contenu de ce fichier avec les informations suivantes :

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

Mettez à jour l'initramfs :

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

Chaque unité peut être examinée individuellement :

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

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

    Update Time : Sun May  1 13:27:53 2022
    Bad Block Log : 512 entries available at offset 16 sectors
```

```
Checksum : 31909df9 - correct
Events : 18

Layout : left-symmetric
Chunk Size : 512K

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

29.3 - Remplacer une Unité Défaillante

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

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

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

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

    Update Time : Sun May  1 13:43:24 2022
      State : clean, degraded
```

```
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0
```

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

```
Consistency Policy : resync
```

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

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

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

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

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

```
root@debian11:~# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Sun May  1 13:27:48 2022
    Raid Level : raid5
    Array Size : 1019904 (996.00 MiB 1044.38 MB)
```

```
Used Dev Size : 509952 (498.00 MiB 522.19 MB)
Raid Devices : 3
Total Devices : 2
Persistence : Superblock is persistent
```

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

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

```
Consistency Policy : resync
```

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

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

Constatez maintenant l'existence de votre RAID :

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



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

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

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

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

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

```
root@debian11:~# cryptsetup luksRemoveKey /dev/sdc11
Enter passphrase to be deleted: fenestros123456789

WARNING!
=====
This is the last keyslot. Device will become unusable after purging this key.

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

Supprimez maintenant LUKs :

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

Vérifiez de nouveau le statut :

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

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

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	32G	0	disk	
_sda1	8:1	0	31G	0	part	/
_sda2	8:2	0	1K	0	part	
_sda5	8:5	0	975M	0	part	[SWAP]
sdb	8:16	0	64G	0	disk	
sdc	8:32	0	4G	0	disk	
_sdc1	8:33	0	100M	0	part	
_sdc2	8:34	0	100M	0	part	
_sdc3	8:35	0	100M	0	part	
_sdc4	8:36	0	1K	0	part	
_sdc5	8:37	0	500M	0	part	
_sdc6	8:38	0	200M	0	part	
_vg0-lv1	254:0	0	104M	0	lvm	
_sdc7	8:39	0	300M	0	part	
_vg0-lv2	254:1	0	112M	0	lvm	
_sdc8	8:40	0	500M	0	part	
_md1	9:1	0	996M	0	raid5	
_sdc9	8:41	0	400M	0	part	
_vg0-lv2	254:1	0	112M	0	lvm	
_sdc10	8:42	0	500M	0	part	
_md1	9:1	0	996M	0	raid5	
_sdc11	8:43	0	500M	0	part	
_sdc12	8:44	0	200M	0	part	
sr0	11:0	1	378M	0	rom	

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

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

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

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

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


    Layout : left-symmetric
    Chunk Size : 512K

Consistency Policy : resync

    Rebuild Status : 56% complete

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

Events : 32

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

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

Version : 1.2

Creation Time : Sun May 1 13:27:48 2022

Raid Level : raid5

Array Size : 1019904 (996.00 MiB 1044.38 MB)

Used Dev Size : 509952 (498.00 MiB 522.19 MB)

Raid Devices : 3

Total Devices : 3

Persistence : Superblock is persistent

Update Time : Sun May 1 14:03:07 2022

State : clean

Active Devices : 3

Working Devices : 3

Failed Devices : 0

Spare Devices : 0

Layout : left-symmetric

Chunk Size : 512K

Consistency Policy : resync

Name : debian11:1 (local to host debian11)

UUID : c0f945a0:f65b2136:b7913f8a:3707ffa2

Events : 40

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

Vérifiez la prise en compte de la configuration :

```
root@debian11:~# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                  8:0    0   32G  0 disk
├─sda1               8:1    0   31G  0 part /
├─sda2               8:2    0    1K  0 part
└─sda5               8:5    0   975M  0 part [SWAP]
sdb                  8:16    0   64G  0 disk
sdc                  8:32    0    4G  0 disk
├─sdc1               8:33    0   100M  0 part
├─sdc2               8:34    0   100M  0 part
├─sdc3               8:35    0   100M  0 part
├─sdc4               8:36    0    1K  0 part
├─sdc5               8:37    0   500M  0 part
├─sdc6               8:38    0   200M  0 part
│   └─vg0-lv1        254:0    0   104M  0 lvm
├─sdc7               8:39    0   300M  0 part
│   └─vg0-lv2        254:1    0   112M  0 lvm
├─sdc8               8:40    0   500M  0 part
│   └─md1            9:1    0   996M  0 raid5
├─sdc9               8:41    0   400M  0 part
│   └─vg0-lv2        254:1    0   112M  0 lvm
├─sdc10              8:42    0   500M  0 part
│   └─md1            9:1    0   996M  0 raid5
├─sdc11              8:43    0   500M  0 part
│   └─md1            9:1    0   996M  0 raid5
└─sdc12              8:44    0   200M  0 part
sr0                  11:0    1   378M  0 rom
```

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

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

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

Mettez à jour l'initramfs :

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

Le Swap

Taille du swap

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

Mémoire	Taille du swap
4 Go ou moins	2 Go
4 Go à 16 Go	4 Go
16 Go à 64 Go	8 Go
64 Go à 256 Go	16 Go

Partitions de swap

Une partition de swap peut être créée sur :

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

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@debian11:~# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/sdb5	partition	998396	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@debian11:~# 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
PRI0	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.

La Commande `swapoff`

Dans le cas de notre exemple, la partition de swap se trouve sur **/dev/sdb5**. Pour la désactiver, il convient de saisir la commande suivante :

```
root@debian11:~# swapoff /dev/sdb5
```

```
root@debian11:~# swapon -s  
root@debian11:~#
```

Options de la Commande

```
root@debian11:~# swapoff --help
```

Usage:

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

Disable devices and files for paging and swapping.

Options:

-a, --all	disable all swaps from /proc/swaps
-v, --verbose	verbose mode
-h, --help	display this help
-V, --version	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)`.

LAB #30 - 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 268Mo, appelé **swap**, saisissez la commande suivante :

```
root@debian11:~# 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, 0.22828 s, 1.2 GB/s
```

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

```
root@debian8:~# mkswap /swap
Setting up swspace version 1, size = 262140 KiB
no label, UUID=68393385-7543-49ad-ab2a-3cd17725c5cc
```

Pour activer le fichier avec une priorité de **3**, saisissez la commande suivante :

```
root@debian11:~# mkswap /swap
mkswap: /swap: insecure permissions 0644, 0600 suggested.
Setting up swspace version 1, size = 256 MiB (268431360 bytes)
no label, UUID=db698755-d483-46fa-bbd8-137467e2af5e

root@debian11:~# chmod 600 /swap

root@debian11:~# swapon /dev/sdc5

root@debian11:~# swapon -p 3 /swap
```

Pour visualiser les espaces swap, saisissez la commande suivante :

```
root@debian11:~# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/sdb5	partition	998396	0	-2
/swap	file	262140	0	3



Important : Le fichier de swap ayant une priorité de 3 sera utilisé avant la partition de swap ayant une priorité de -2. 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@debian11:~# swapoff /swap
```

```
root@debian11:~# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/sdb5	partition	998396	0	-2

Supprimez maintenant le fichier de swap :

```
root@debian11:~# rm /swap
```