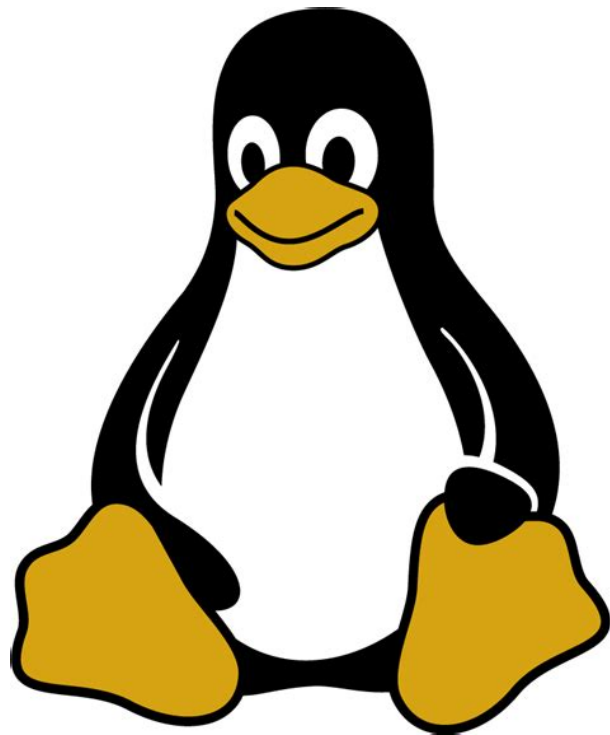




# Partitionnement avec fdisk et LVM sous Debian 9

Superviseur : Patrice Krzanik

Tunui Franken



---

## Table des matières

<b>1</b>	<b>But de la manœuvre</b>	<b>3</b>
<b>2</b>	<b>Prérequis</b>	<b>3</b>
<b>3</b>	<b>Création des partitions avec fdisk</b>	<b>3</b>
3.1	Vérification de la table des partitions . . . . .	3
3.2	Création des volumes physiques . . . . .	3
3.3	Modification des types de partition . . . . .	4
3.4	Ajout du flag d’amorçage sur /dev/sda1 . . . . .	5
3.5	Enregistrement sur le disque . . . . .	5
<b>4</b>	<b>Gestion d’un espace disque</b>	<b>6</b>
4.1	Création des volumes physiques . . . . .	6
4.2	Création des groupes de volume . . . . .	6
4.3	Création du volume logique perso . . . . .	7
4.4	Gestion du système de fichiers . . . . .	7
4.5	Changer la taille du LV, modifier le FS et vérification . . . . .	8
4.6	Modifier le fichier <code>fstab</code> . . . . .	9
<b>5</b>	<b>À vous de jouer</b>	<b>9</b>
5.1	Cahier des charges . . . . .	9
5.2	Mise en œuvre . . . . .	9

## 1 But de la manœuvre

Cette procédure est une transposition pour Debian d'un TP sous CentOS, qui utilise `fdisk` et LVM pour le partitionnement.

## 2 Prérequis

Une VM sous Debian 9. Le disque fait 20Go, et il est prépartitionné de la manière suivante :

- `/dev/sda1` de 500 MiB, formaté en `ext4` et monté sur `/boot`.
- `/dev/sda2` de 3 GiB, volume physique LVM dans le groupe de volume `vg_1`, qui est formaté en `ext4` et monté sur `/`.
- `/dev/sda3` de 1 GiB, volume physique LVM dans le groupe de volume `vg_1`, qui est formaté en `ext4` et monté sur `/`.
- `/dev/sda4`, partition étendue, qui s'étend jusqu'à la fin du disque pour inclure les futures partitions que l'on va créer.
- `/dev/sda5` de 6 GiB, volume physique LVM dans le groupe de volume `vg_2`, qui est formaté en `ext4` et monté sur `/usr`.
- `/dev/sda6` de 1 GiB, pour le swap.
- `/dev/sda7` de 1 GiB, volume physique LVM dans le groupe de volume `vg_3`, qui est formaté en `ext4` et monté sur `/home`.

## 3 Création des partitions avec `fdisk`

### 3.1 Vérification de la table des partitions

On ouvre une session avec le compte `root` sur une console terminale.

On entre dans l'utilitaire `fdisk` avec la commande suivante :

```
# fdisk /dev/sda
```

On nous demande la commande `fdisk` à utiliser. On va taper `p` pour `print`, ce qui affiche la table des partitions :

```
Commande (m pour l'aide) : p
```

Rappel : sur les systèmes Intel 32 bits on peut créer jusqu'à 4 partitions primaires ou 3 primaires et 1 étendue dans laquelle on créera des partitions logiques.

On dispose de la commande `m` pour afficher l'aide (le menu) qui présente toutes les commandes `fdisk` disponibles.

### 3.2 Création des volumes physiques

On veut créer 3 nouvelles partitions `/dev/sda8` `/dev/sda9` et `/dev/sda10`.

On peut créer une nouvelle partition avec `n` :

Comme nous avons déjà 3 partitions primaires, le numéro de partition se sélectionne automatiquement en numéro 8 de type logique.

```
Commande (m pour l'aide) : n
Toutes les partitions primaires sont utilisees.
Ajout de la partition logique 8
Premier secteur (224418304-41943039, 224418304 par default) :
```

On peut taper `Entrée` pour garder la valeur par défaut. Il est important quand on crée des partitions de respecter la taille des blocs du disque. Le plus prudent est donc de créer les partitions dans l'ordre en sélectionnant à chaque fois la valeur par défaut pour le premier secteur.

```
Dernier secteur, +secteurs ou +taille{K,M,G,T,P} (24418304-41943039,
41943039 par default) : +1500M
```

Une nouvelle partition 8 de type `<< Linux >>` et de taille 1,5 GiB a ete creee.

```
Commande (m pour l'aide) : n
Toutes les partitions primaires sont utilisees.
Ajout de la partition logique 9
Premier secteur (27492352-41943039, 27492352 par default) :
Dernier secteur, +secteurs ou +taille{K,M,G,T,P} (27492352-41943039,
41943039 par default) : +1G
```

Une nouvelle partition 9 de type `<< Linux >>` et de taille 1 GiB a ete creee.

```
Commande (m pour l'aide) : n
Toutes les partitions primaires sont utilisees.
Ajout de la partition logique 10
Premier secteur (29591552-41943039, 29591552 par default) :
Dernier secteur, +secteurs ou +taille{K,M,G,T,P} (29591552-41943039,
41943039 par default) : +2,1G
```

Une nouvelle partition 10 de type `<< Linux >>` et de taille 2,1 GiB a ete creee.

On peut maintenant afficher ce que l'on vient de créer :

```
Commande (m pour l'aide) : p
```

### 3.3 Modification des types de partition

Par défaut, `fdisk` a créé les partitions avec le type 83, Linux. Pour créer des volumes logiques dans un groupe de volumes il faut leur donner le type 8e.

Pour cela, on utilise la commande `t`, puis on sélectionne les partitions à modifier.

Les 7 premières partitions sont déjà correctement configurées, il faut changer les partitions 8 à 10 :

```
Commande (m pour l'aide) : t
Numero de partition (1-10, 10 par default) : 8
Type de partition (taper L pour afficher tous les types) : 8e
```

```
Type de partition << Linux >> modifie en << Linux LVM >>.
```

```
Commande (m pour l'aide) : t
Numero de partition (1-10, 10 par default) : 9
Type de partition (taper L pour afficher tous les types) : 8e
```

```
Type de partition << Linux >> modifie en << Linux LVM >>.
```

```
Commande (m pour l'aide) : t
Numero de partition (1-10, 10 par default) :
Type de partition (taper L pour afficher tous les types) : 8e
```

```
Type de partition << Linux >> modifie en << Linux LVM >>.
```

Toutes les partitions devraient maintenant être de type `8e` (LVM Linux), sauf `/dev/sda1` qui est de type `83` (Linux), `/dev/sda4` qui est de type `5` (Étendue) et `/dev/sda6` qui est de type `82` (swap).

### 3.4 Ajout du flag d'amorçage sur `/dev/sda1`

Si ce n'est déjà fait, il faut marquer la première partition avec l'indicateur d'amorçage :

```
Commande (m pour l'aide) : a
Numero de partition (1-7, 7 par default) : 1
```

```
L'indicateur d'amorçage de la partition 1 est maintenant active.
```

### 3.5 Enregistrement sur le disque

Si on a bien vérifié notre nouvelle table des partitions, on peut écrire la table sur le disque et quitter `fdisk` :

```
Commande (m pour l'aide) : w
```

La table sera utilisée au prochain démarrage, puisque nous sommes déjà loggés. On va donc redémarrer :

```
# reboot
```

## 4 Gestion d'un espace disque

Avant de pouvoir utiliser les volumes logiques, il faut installer le paquet `lvm2` :

```
# apt install lvm2
```

### 4.1 Création des volumes physiques

Commandes `pvscan`, `pvcreate` et `pvremove`.

Pour créer un groupe de volume il faut d'abord avoir des volumes physiques disponibles. La commande `pvscan` liste les PV (Physical Volumes, volumes physiques) reconnus par le système.

```
# pvscan
PV /dev/sda7 VG vg_3      lvm2
PV /dev/sda2 VG vg_1      lvm2
PV /dev/sda3 VG vg_1      lvm2
PV /dev/sda5 VG vg_2      lvm2
```

Les partitions `/dev/sda8`, `/dev/sda9` et `/dev/sda10` n'apparaissent pas. Les PV listés sont déjà intégrés dans des groupes de volumes.

La commande `pvcreate` permet de taguer les partitions pour utilisation par LVM. Voir le manuel de `pvcreate` pour la syntaxe : `man pvcreate`.

On veut ajouter les partitions 8 à 10. La commande sera donc :

```
# pvcreate /dev/sda8 /dev/sda9 /dev/sda10
```

Ou bien de manière plus simple :

```
# pvcreate /dev/sda{8..10}
```

Puis avec la commande `pvremove`, retirer `/dev/sda10` de la liste :

```
# pvremove /dev/sda10
```

### 4.2 Création des groupes de volume

Commandes `vgscan`, `vgcreate`, `vgremove` et `vgextend`.

La commande `vgscan` affiche les groupes de volumes (VG, Volume Group) actifs mais les informations sont succinctes. Utilisez plutôt la commande `pvscan` pour vérifier les différentes étapes de la création du VG.

À l'aide de la commande `vgcreate`, créer le groupe de volume `vg_4` comprenant le PV `/dev/sda8`.

```
# vgcreate vg_4 /dev/sda8
```

On peut vérifier la création avec `pvscan` :

```
# pvscan
PV /dev/sda7 VG vg_3          lvm2
PV /dev/sda2 VG vg_1          lvm2
PV /dev/sda3 VG vg_1          lvm2
PV /dev/sda5 VG vg_2          lvm2
PV /dev/sda8 VG vg_4          lvm2
PV /dev/sda9                   lvm2
Total: 6 [12,70 GiB] / in use: 5 [11,70 GiB] / in no VG: 1 [1,00 GiB]
```

En cas d'erreur, détruire le VG avec `vgremove`.

À l'aide de la commande `vgextend`, étendre le VG `vg_4` pour qu'il contienne le PV `/dev/sda9`.

```
# vgextend vg_4 /dev/sda9
```

Avec `pvscan`, on peut vérifier que tous nos PV sont maintenant dans des VG.

### 4.3 Création du volume logique perso

Commandes `lvscan`, `lvcreate`, `lvremove` et `lvextend`.

À l'aide de la commande `lvcreate`, créer le LV (Logical Volume, volume logique) `perso` d'une taille de 1500 Mo sur le VG `vg_4` :

```
# lvcreate -L 1500M -n perso /dev/vg_4
```

En cas d'erreur, détruire le volume avec `lvremove`.

On vérifie avec `lvscan` :

```
# lvscan
ACTIVE          '/dev/vg_3/utilisateurs'
ACTIVE          '/dev/vg_1/system'
ACTIVE          '/dev/vg_2/applications'
ACTIVE          '/dev/vg_4/perso'
```

### 4.4 Gestion du système de fichiers

Pour pouvoir accéder à ce nouvel espace disque il faut :

1. installer un système de fichiers utilisable par le système
2. le monter dans l'arborescence du système d'exploitation (création du point de montage).

Commandes `mkdir`, `mkfs`, `mount`, `df`.

Créer le point de montage `/secret` avec la commande `mkdir` :

```
# mkdir /secret
```

Installer le système de fichiers `ext4` sur le volume logique `perso` avec la commande `mkfs` :

```
# mkfs.ext4 /dev/vg_4/perso
```

Monter le volume logique sur le point de montage :

```
# mount /dev/vg_4/perso /secret
```

Vérifier le montage avec la commande `df` :

```
# df
Sys. de fichiers      blocs de 1K Utilise Disponible Uti% Monte sur
...
/dev/mapper/vg_4-perso 1479088    4512    1381392    1% /secret
```

## 4.5 Changer la taille du LV, modifier le FS et vérification

Commandes `lvextend`, `resize2fs`, `umount`, `mount` et `df`.

Démonter le volume logique `perso` :

```
# umount /dev/vg_4/perso
```

Augmenter la taille du volume logique de 500 Mo :

```
# lvextend -L +500M /dev/vg_4/perso
```

On peut vérifier la nouvelle taille avec `lvscan`.

Monter le volume logique sur son point de montage `/secret` :

```
# mount /dev/vg_4/perso /secret
```

Redimensionner le système de fichier pour occuper la totalité du volume logique `perso`.

```
# resize2fs /dev/vg_4/perso
```

Vérifier la taille du volume logique :

```
# df
```

Répéter les différentes étapes pour que le système de fichiers occupe la totalité du VG (2,46 GiB).

```
# lvextend -l 100%PV /dev/vg_4/perso
```

Nous augmentons la taille du LV `/dev/vg_4/perso` en prenant 100% du PV `/dev/sda9`.



## 4.6 Modifier le fichier `fstab`

L'espace disque doit être remonté à chaque reboot du système. Pour cela il faut modifier le fichier `fstab` qui est dans le répertoire `/etc`.

On ouvre `/etc/fstab` avec par exemple `nano`, et on ajoute la ligne suivante :

```
/dev/mapper/vg_4-perso /secret ext4 defaults 0 0
```

## 5 À vous de jouer

### 5.1 Cahier des charges

Vous devez installer plusieurs applications sur votre serveur mais le répertoire `/usr` est déjà occupé à 86%. On vous demande de récupérer l'espace disque `/dev/sda10` pour augmenter la taille de ce répertoire.

### 5.2 Mise en œuvre

1. Taguer la partition `/dev/sda10` pour être incluse dans un VG :

```
# pvcreate /dev/sda10
```

On vérifie avec `pvscan`. La partition `/dev/sda10` apparaît bien dans la liste.

2. Augmenter la taille du VG correspondant à `/usr` pour inclure la partition `/dev/sda10` :

```
# vgextend vg_2 /dev/sda10
```

On vérifie avec `pvscan`. Le volume a augmenté de la taille de la partition `/dev/sda10`.

3. Redimensionner le volume logique :

```
# lvextend /dev/vg_2/applications /dev/sda10
```

On vérifie avec `vgscan`. Le volume logique a la même taille que le groupe de volume.

4. Redimensionner le système de fichiers :

```
# resize2fs /dev/vg_2/applications
```

On vérifie avec `df`. Le système a la même taille que le volume logique.