

TP 5 - Systèmes de fichiers, partitions et disques



Dans ce cinquième TP, nous allons manipuler divers outils de gestion des disques et partitions, ainsi que le partitionnement LVM. Nous verrons comment créer des points de montage et monter des partitions.

Exercice 1. Disques et partitions

1. Dans l'interface de configuration de votre VM, créez un second disque dur, de 5 Go dynamiquement alloués ; puis démarrez la VM
2. Vérifiez que ce nouveau disque dur est bien détecté par le système
Plusieurs solutions possibles : on constate la présence d'un fichier sdb dans /dev ; ou encore lsblk et fdisk -l mentionnent un disque de 5 Go nommé sdb
3. Partitionnez ce disque en utilisant `fdisk` : créez une première partition de 2 Go de type *Linux* (n°83), et une seconde partition de 3 Go en NTFS (n°7)
fdisk /dev/sdb puis choisir n, p, +2G pour la première partition ; répéter pour la 2ème, puis changer son type avec t et choisir le type 7. Prévisualiser le résultat avec p puis écrire les modifications avec w. Vérifier ensuite avec fdisk -l que les partitions ont bien été créées
4. A ce stade, les partitions ont été créées, mais elles n'ont pas été formatées avec leur système de fichiers. A l'aide de la commande `mkfs`, formatez vos deux partitions (⚠ pensez à consulter le manuel!)
Le manuel nous informe que la commande mkfs est dépréciée en faveur des commandes spécialisées mkfs.* ; on utilise donc mkfs.ext4 /dev/sdb1 pour la première partition, et mkfs.ntfs /dev/sdb2 pour la seconde
5. Pourquoi la commande `df -T`, qui affiche le type de système de fichier des partitions, ne fonctionne-t-elle pas sur notre disque ?
Parce qu'il n'a pas encore été monté ! (cf. manuel)
6. Faites en sorte que les deux partitions créées soient montées automatiquement au démarrage de la machine, respectivement dans les points de montage `/data` et `/win` (vous pourrez vous passer des UUID en raison de l'impossibilité d'effectuer des copier-coller)
Créer les deux dossiers à la racine, puis éditer /etc/fstab :

```
/dev/sdb1 /data ext4 defaults 0 0
/dev/sdb2 /win ntfs defaults 0 0
```

Ensuite, valider par mount -a puis mount
7. Utilisez la commande `mount` puis redémarrez votre VM pour valider la configuration
8. Montez votre clé USB dans la VM
Penser à créer le point de montage
9. Créez un dossier partagé entre votre VM et votre système hôte (rem. il peut être nécessaire d'installer les *Additions invité* de VirtualBox).
cf. Section 4.3 du manuel de VirtualBox

Exercice 2. Partitionnement LVM

Dans cet exercice, nous allons aborder le partitionnement LVM, beaucoup plus flexible pour manipuler les disques et les partitions.

1. On va réutiliser le disque de 5 Gio de l'exercice précédent. Commencez par démonter les systèmes de fichiers montés dans `/data` et `/win` s'ils sont encore montés, et supprimez les lignes correspondantes du fichier `/etc/fstab`

2. Supprimez les deux partitions du disque, et créez une partition unique de type LVM

💡 La création d'une partition LVM n'est pas indispensable, mais vivement recommandée quand on utilise LVM sur un disque entier. En effet, elle permet d'indiquer à d'autres OS ou logiciels de gestion de disques (qui ne reconnaissent pas forcément le format LVM) qu'il y a des données sur ce disque.

⚠ Attention à ne pas supprimer la partition système !

Lancer `fdisk`, créer une nouvelle partition puis changer le type (commande `t`) et donner le type `8e` (LVM)

3. A l'aide de la commande `pvcreate`, créez un *volume physique* LVM. Validez qu'il est bien créé, en utilisant la commande `pvdisplay`.

💡 Toutes les commandes concernant les volumes physiques commencent par `pv`. Celles concernant les groupes de volumes commencent par `vg`, et celles concernant les volumes logiques par `lv`.

`sudo pvcreate /dev/sdb1`

4. A l'aide de la commande `vgcreate`, créez un *groupe de volumes*, qui pour l'instant ne contiendra que le volume physique créé à l'étape précédente. Vérifiez à l'aide de la commande `vgdisplay`.

💡 Par convention, on nomme généralement les groupes de volumes `vgxx` (où `xx` représente l'indice du groupe de volume, en commençant par 00, puis 01...)

`sudo vgcreate vg00 /dev/sdb1`

5. Créez un volume logique appelé `lvData` occupant l'intégralité de l'espace disque disponible.

💡 On peut renseigner la taille d'un volume logique soit de manière absolue avec l'option `-L` (par exemple `-L 10G` pour créer un volume de 10 Gio), soit de manière relative avec l'option `-l` : `-l 60%VG` pour utiliser 60% de l'espace total du groupe de volumes, ou encore `-l 100%FREE` pour utiliser la totalité de l'espace libre.

`sudo lvcreate -n lvData -l 100%FREE vg00`

6. Dans ce volume logique, créez une partition que vous formateriez en `ext4`, puis procédez comme dans l'exercice 1 pour qu'elle soit montée automatiquement, au démarrage de la machine, dans `/data`.

`fdisk /dev/mapper/vg00-lvData`

`mkfs.ext4 /dev/mapper/vg00-lvData`

Ubuntu *mappe* les périphériques physiques en périphériques virtuels de plus haut niveau

💡 A ce stade, l'utilité de LVM peut paraître limitée. Il trouve tout son intérêt quand on veut par exemple agrandir une partition à l'aide d'un nouveau disque.

7. Eteignez la VM pour ajouter un second disque (peu importe la taille pour cet exercice). Redémarrez la VM, vérifiez que le disque est bien présent. Puis, répétez les questions 2 et 3 sur ce nouveau disque.

`fdisk /dev/sdc`

`pvcreate /dev/sdc1`

8. Utilisez la commande `vgextend <nom_vg> <nom_pv>` pour ajouter le nouveau disque au groupe de volumes

`sudo vgextend vg00 /dev/sdc1`

9. Utilisez la commande `lvresize` (ou `lvextend`) pour agrandir le volume logique. Enfin, il ne faut pas oublier de redimensionner le système de fichiers à l'aide de la commande `resize2fs`.

`sudo lvresize -l 2046 /dev/mapper/vg00-lvData` (2046 correspond à 8Gio)

`sudo e2fsck -f /dev/mapper/vg00-lvData`

`sudo resize2fs /dev/mapper/vg00-lvData`

💡 Il est possible d'aller beaucoup plus loin avec LVM, par exemple en créant des volumes par bandes (l'équivalent du RAID 0) ou du mirroring (RAID 1). Le but de cet exercice n'était que de présenter les fonctionnalités de base.