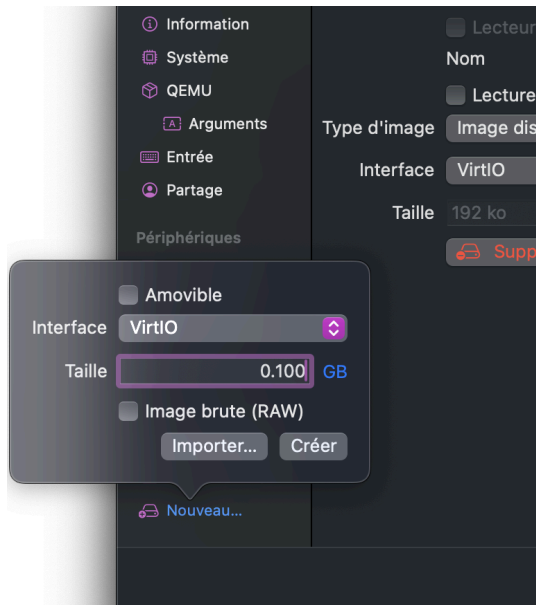


Ajouter un nouveau disque SATA de 100Mo dans la machine



Pour commencer, j'ai éteint ma machine virtuelle (VM) et accédé aux **paramètres** de celle-ci. Sous Mac, il n'est pas possible d'ajouter directement un disque **SATA**, donc j'ai choisi de créer un disque **VirtIO** à la place.

J'ai configuré la taille du disque à 100 Mo pour correspondre à la consigne. Ensuite, j'ai **créé** le disque et **relancé** ma VM afin de pouvoir le **configurer** et **l'utiliser** dans le système.

Trouver le nom de device attribué au disque

Pour identifier le nom de device attribué au nouveau disque de 100 Mo, j'ai utilisé la commande suivante dans le terminal : `lsblk`

Cette commande liste tous les périphériques de **stockage** connectés à la machine, y compris leurs **partitions** et leurs **points de montage**. Grâce à cette commande, j'ai pu repérer le disque récemment ajouté, identifiable par sa taille (environ 100 Mo). Il était nommé `/dev/vdb`

Cette étape est essentielle pour connaître le **chemin** du disque avant de le **manipuler** et de le **partitionner**.

```
ubuntu--vg-ubuntu--lv
253:0    0   38,5G  0 lvm  /var/snap/fi
vdb      252:16  0      0B  0 disk
arthur@ubuntu:~$
```

Partitionner le nouveau disque en deux partitions GPT de 50Mo

J'ai utilisé la commande `sudo fdisk /dev/vdb` pour **accéder** au **disque**.

Ensuite, j'ai choisi de **créer** une nouvelle **table** de **partition** GPT en tapant `g`.

Pour chaque **partition**, j'ai utilisé la commande `n`, en fixant la **taille** à 50 Mo. Après avoir créé les deux **partitions**, j'ai **sauvegardé** les modifications en tapant `w`.

Le disque est maintenant **partitionné** en deux, avec les **partitions** `/dev/vdb1` et `/dev/vdb2`.

```
arthur@ubuntu:~$ sudo fdisk /dev/vdb

Bienvenue dans fdisk (util-linux 2.38.1).
Les modifications resteront en mémoire jusqu'à écriture.
Soyez prudent avant d'utiliser la commande d'écriture.

Le périphérique ne contient pas de table de partitions reconnue.
Created a new DOS (MBR) disklabel with disk identifier 0x7110f61f.

Commande (m pour l'aide) : g
Une nouvelle étiquette de disque GPT a été créée (GUID : B7FE0EA7-C5B0-F04D-8
862-1D56B365262F).

Commande (m pour l'aide) : n
Numéro de partition (1-128, 1 par défaut) : 1
Premier secteur (2048-2097118, 2048 par défaut) :
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097118, 2095103 par défaut) : +50M

Une nouvelle partition 1 de type « Linux filesystem » et de taille 50 MiB a été créée.

Commande (m pour l'aide) : n
Numéro de partition (2-128, 2 par défaut) : 2
Premier secteur (104448-2097118, 104448 par défaut) :
Last sector, +/-sectors or +/-size{K,M,G,T,P} (104448-2097118, 2095103 par défaut) : +50M

Une nouvelle partition 2 de type « Linux filesystem » et de taille 50 MiB a été créée.

Commande (m pour l'aide) : w
La table de partitions a été altérée.
Appel d'ioctl() pour relire la table de partitions.
Synchronisation des disques.
```

Formater la première partition au format **EXT4** avec le label "MONEXT4"

J'ai **formaté** la première **partition** en utilisant la commande :

```
sudo mkfs.ext4 -L MONEXT4 /dev/vdb1.
```

Cela a créé un **système de fichiers** au format **EXT4** sur la **partition** `/dev/vdb1` et lui a attribué le label "MONEXT4". Ce format est adapté pour les systèmes Linux et permet une **gestion efficace** des fichiers.

```
arthur@ubuntu:~$ sudo mkfs.ext4 -L MONEXT4 /dev/vdb1
mke2fs 1.47.0 (5-Feb-2023)
Rejet des blocs de périphérique : complété
En train de créer un système de fichiers avec 12800 4k blocs et 12800 i-noeuds.

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (1024 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété
```

Monter la partition dans `/mnt/monext4`

Pour rendre la **partition** accessible, j'ai monté la **partition** `/dev/vdb1` dans le **répertoire** `/mnt/monext4` avec la commande suivante :

```
sudo mount /dev/vdb1 /mnt/monext4
```

Cela permet d'accéder aux **fichiers** et **dossiers** présents sur cette **partition** à partir du chemin `/mnt/monext4`.

```
arthur@ubuntu:~$ sudo mkdir -p /mnt/monext4
arthur@ubuntu:~$ sudo mount /dev/vdb1 /mnt/monext4
```

Montrer le résultat intermédiaire (par exemple, capture des commandes `fdisk -l [new_disk]`, `df -h`, `lsblk --fs -e7`)

Pour vérifier la **configuration** de la **partition** et s'assurer qu'elle a été correctement montée, j'ai utilisé les commandes suivantes :

1. `fdisk -l /dev/vdb` :

Cette commande **liste** les informations sur le **disque** et ses **partitions**. Elle permet de **vérifier** que les deux **partitions** de 50 Mo ont bien été créées sur le disque `/dev/vdb`.

```
arthur@ubuntu:~$ sudo fdisk -l /dev/vdb
Disque /dev/vdb : 1 GiB, 1073741824 octets, 2097152 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : gpt
Identifiant de disque : B7FE0EA7-C5B0-F04D-8862-1D56B365262F

Périphérique Début   Fin   Secteurs Taille Type
/dev/vdb1      2048 104447   102400    50M Système de fichiers Linux
/dev/vdb2    104448 206847   102400    50M Système de fichiers Linux
```

2. `df -h` :

Cette commande affiche l'**espace disque** utilisé et disponible de manière **lisible** pour l'humain.

Elle permet de voir si la **partition** `/dev/vdb1` est **montée** et **accessible** à l'emplacement `/mnt/monext4`.

```
arthur@ubuntu:~$ df -h
Sys. de fichiers      Taille Utilisé Dispo Uti% Monté sur
tmpfs                  391M   1,9M  389M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 38G    21G   16G  57% /
tmpfs                  2,0G     0   2,0G   0% /dev/shm
tmpfs                  5,0M    8,0K   5,0M   1% /run/lock
/dev/vda2              2,0G   285M   1,6G  16% /boot
/dev/vda1              1,1G    6,4M   1,1G   1% /boot/efi
tmpfs                  391M   112K   391M   1% /run/user/1000
/dev/vdb1              43M   152K    40M   1% /mnt/monext4
```

3. `lsblk --fs -e7`:

Cette commande montre la **structure des disques** et **partitions** sous forme **d'arborescence**, avec les types de systèmes de fichiers associés.

Elle permet de **vérifier** que la partition `/dev/vdb1` est bien **formatée** en EXT4 et que le point de montage est correct.

```
arthur@ubuntu:~$ lsblk --fs -e7
NAME FSTYPE FSVER LABEL UUID                                FSAVAIL FSUSE%
MOUNTPOINTS
sr0
vda
├─vda1
│   └─vfat FAT32      A440-8AF5                      1G      1%
│   /boot/efi
├─vda2
│   └─ext4  1.0        7c9b207d-a77a-422e-9853-e1e5902cfdbe 1,5G    15%
│   /boot
└─vda3
    └─LVM2_m LVM2      8bCqVS-2TJJ-4CGg-WyN1-E558-T0Nz-mFHBr
        └─ubuntu--vg-ubuntu--lv
            └─ext4  1.0        21e38355-448f-40d6-b508-131840d8520c 15,4G   54%
            /var/snap/firefox/common/host-hunspell
/
vdb
├─vdb1
│   └─ext4  1.0        MONEXT4
│           2b5bb1e9-ff1d-4e5e-82e1-ad9bf7390aa3 39,2M   0%
│   /mnt/monext4
└─vdb2
```

Ces commandes montrent le **bon état** de la **configuration** du **disque** et de ses **partitions** avant de passer aux étapes suivantes.

Faire de la seconde partition une partition de swap et l'ajouter au noyau

J'ai utilisé la commande suivante pour **formater** la seconde **partition**, `/dev/vdb2`, en espace de **swap** :

```
sudo mkswap /dev/vdb2
```

Cela configure la **partition** pour qu'elle puisse être utilisée comme **mémoire swap**, ce qui permet **d'étendre** la mémoire RAM en utilisant de l'**espace disque**.

Pour activer le **swap** et le rendre disponible immédiatement, j'ai exécuté :

```
sudo swapon /dev/vdb2
```

Cette commande ajoute la **partition** de **swap** au noyau, la rendant ainsi **utilisable** par le système pour **gérer** la **mémoire**.

```
arthur@ubuntu:~$ sudo mkswap /dev/vdb2
Configure l'espace d'échange (swap) en version 1, taille = 50 MiB (52424704 octets)
pas d'étiquette, UUID=3044129b-9cf9-4da4-a011-6505ee1d65ea
arthur@ubuntu:~$ sudo swapon /dev/vdb2
arthur@ubuntu:~$ sudo swapon --show
NAME      TYPE      SIZE USED  PRIO
/swap.img file      3,8G   0B   -2
/dev/vdb2 partition 50M    0B   -3
```

Montrer le résultat final à votre examinateur (par exemple, capture des commandes `fdisk -l` `[new_disk]`, `swapon`, `lsblk --fs -e7`)

Pour démontrer que les **partitions** ont été correctement **créées** et **configurées**, j'ai exécuté plusieurs commandes et capturé leurs résultats :

Pour **vérifier** les **partitions**, j'ai utilisé la commande suivante pour afficher la **liste** des **disques** et **partitions** :

```
sudo fdisk -l /dev/vdb
```

Cela a permis de **confirmer** que le **nouveau** disque de 100 Mo est bien **partitionné** en deux **partitions** de 50 Mo chacune.

```
arthur@ubuntu:~$ sudo fdisk -l /dev/vdb
[sudo] Mot de passe de arthur :
Disque /dev/vdb : 1 GiB, 1073741824 octets, 2097152 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : gpt
Identifiant de disque : B7FE0EA7-C5B0-F04D-8862-1D56B365262F

Périphérique Début Fin Secteurs Taille Type
/dev/vdb1 2048 104447 102400 50M Système de fichiers Linux
/dev/vdb2 104448 206847 102400 50M Système de fichiers Linux
```

Pour vérifier **l'état** du **swap**, j'ai exécuté :

```
sudo swapon --show
```

Cela a affiché les **partitions** de swap **activées**, confirmant que `/dev/vdb2` est bien en **cours d'utilisation** comme swap.

```
arthur@ubuntu:~$ swapon --show
NAME      TYPE      SIZE USED PRIO
/swap.img file      3,8G  0B  -2
/dev/vdb2 partition 50M   0B  -3
```

Pour **afficher** les **systèmes de fichiers** et leur **état**, j'ai utilisé :

```
lsblk --fs -e7
```

Cette commande a montré **toutes** les **partitions** avec leurs types de **systèmes de fichiers**, y compris la **partition** EXT4 et la **partition** swap.

```
arthur@ubuntu:~$ lsblk --fs -e7
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
sr0
vda
├─vda1
│   vfat    FAT32      A440-8AF5                      1G       1%  /boot/efi
├─vda2
│   ext4    1.0        7c9b207d-a77a-422e-9853-e1e5902cfdbe 1,5G     15%  /boot
├─vda3
│   LVM2_m  LVM2          8bCqVS-2TJJ-4CGg-WyN1-E558-T0Nz-mFHBrG
│   └─ubuntu--vg-ubuntu--lv
│       ext4    1.0        21e38355-448f-40d6-b508-131840d8520c 15,4G    54%  /var/snap/firefox/common/host-hunspell
vdb
├─vdb1
│   ext4    1.0        MONEXT4
│                   2b5bb1e9-ff1d-4e5e-82e1-ad9bf7390aa3
└─vdb2
    swap    1          3044129b-9cf9-4da4-a011-6505ee1d65ea
WAP]
```

Ces résultats permettent de **prouver** que les **configurations** ont été effectuées **correctement** et que les **partitions** fonctionnent comme prévu.

Automatiser ces montages au démarrage via le fichier `/etc/fstab`

Pour garantir que les **partitions** sont montées **automatiquement** au **démarrage**, j'ai modifié le fichier `/etc/fstab`. Voici les étapes suivies :

J'ai utilisé l'éditeur de texte `nano` pour ouvrir le **fichier** :

```
sudo nano /etc/fstab
```

Ensuite, j'ajoute les **entrées** pour les **partitions**, j'ai ajouté les lignes suivantes au **fichier** pour définir le **montage automatique** :

```
/dev/vdb2 none swap sw 0 0
```

```
/dev/vdb1 /mnt/monext4 ext4 defaults 0 2
```

La **première** ligne spécifie que `/dev/vdb2` doit être utilisée comme **partition** swap.

La **seconde** ligne indique que la **partition** `/dev/vdb1` (notre partition EXT4) doit être montée dans `/mnt/monext4` avec le **système** de fichiers `ext4`.

```
GNU nano 7.2 /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).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-7T1VEovFa5SZQBduvYgRdu8qX7txFVm4PyMVvYfZ9E5qXj8QUCFpJds18espy9j / ext4 defaults 0 1
# /boot was on /dev/vda2 during curtin installation
/dev/disk/by-uuid/7c9b207d-a77a-422e-9853-e1e5902cfdbe /boot ext4 defaults 0 1
# /boot/efi was on /dev/vda1 during curtin installation
/dev/disk/by-uuid/A440-8AF5 /boot/efi vfat defaults 0 1
/swap.img none swap sw 0 0
/dev/vdb2 none swap sw 0 0
/dev/vdb1 /mnt/monext4 ext4 defaults 0 2
share /media/HostShared 9p trans=virtio,version=9p2000.L,rw,_netdev,nofail 0 0
```

Puis j'ai enregistré et quitter l'éditeur, j'ai ensuite vérifié la syntaxe du fichier avec la commande suivante pour m'assurer qu'il n'y a pas d'erreurs :

```
sudo mount -a
```

```
arthur@ubuntu:~$ sudo mount -a
arthur@ubuntu:~$
```

Montrer le résultat final à votre examinateur (logs système qui montrent ces deux actions au boot comme par exemple "journalctl -b | grep Mounted")

Pour vérifier que les montages de mes **partitions** ont été effectués correctement au **démarrage** après avoir fait un reboot de ma vm, j'ai consulté les **logs** système à l'aide de la commande suivante :

```
journalctl -b | grep -E "mnt-monext4|Activated swap"
```

Cette commande permet de **filtrer** les **entrées** du **journal système** pour n'afficher que celles liées au **montage** de la **partition** EXT4 et à l'**activation** de la **partition** swap.

```
arthur@ubuntu:~$ journalctl -b | grep -E "mnt-monext4|Activated swap"
oct. 18 19:03:45 ubuntu systemd[1]: Activated swap swap.img.swap - /swap.img.
oct. 18 19:03:45 ubuntu systemd[1]: Activated swap dev-vdb2.swap - /dev/vdb2.
oct. 18 19:03:46 ubuntu systemd[1]: Mounting mnt-monext4.mount - /mnt/monext4...
oct. 18 19:03:46 ubuntu systemd[1]: Mounted mnt-monext4.mount - /mnt/monext4.
```

Cela indique que la **partition** EXT4 a été montée avec **succès** dans **/mnt/monext4** et que la **partition** swap a été activée **correctement** au **démarrage** du système.

En guise de conclusion, vous indiquerez le nombre maximum d'inodes que peut stocker votre FS EXT4 ainsi que la manière dont vous l'avez obtenu (capture)

Pour conclure ce TP, j'ai déterminé le nombre **maximum d'inodes** que peut **stocker** mon **système de fichiers** EXT4. Les **inodes** sont des structures de données qui contiennent des informations sur les **fichiers**, et chaque **fichier** ou **répertoire** a besoin d'un **inode** pour être **stocké** sur le disque.

Pour obtenir cette information, j'ai utilisé la commande suivante :

```
sudo tune2fs -l /dev/vdb1 | grep "Inode count"
```

Cette commande affiche le **nombre** total d'**inodes** disponibles pour le **système de fichiers** spécifié (dans ce cas, **/dev/vdb1**, qui correspond à la première **partition** formatée en EXT4).

```
arthur@ubuntu:~$ sudo tune2fs -l /dev/vdb1 | grep 'Inode count'
Inode count:          12800
arthur@ubuntu:~$
```

Ainsi, mon **système de fichiers** EXT4 peut stocker un **maximum** de **12 800 inodes**. Cela signifie que je peux avoir jusqu'à 12 800 **fichiers** et **répertoires** dans cette **partition**, en supposant que chaque inode est utilisé pour un **fichier** ou un **répertoire**.