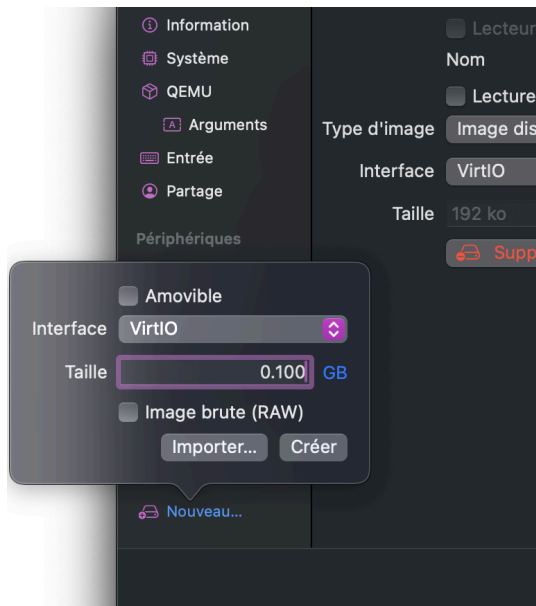


Ajouter encore 4 nouveaux disques SATA de 100Mo dans la machine en prenant soin de rajouter l'option "Branchable à chaud / Hotswap" pour chacun dans Virtualbox (Si vous utiliser autre chose que Virtualbox et que l'option Hotswap n'est pas dispo, continuez sans)



Pour commencer, j'ai éteint ma machine virtuelle (VM) et accéder 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 des 4 disques à 100 Mo pour correspondre à la consigne. Ensuite, j'ai **créé** les disques et **relancé** ma VM afin de pouvoir le **configurer** et **l'utiliser** dans le système.

Créer un RAID 5 entre les 3 premiers nouveaux disques

Voilà les **nouveau** disques (que j'ai du changer plus tard pour delete un des disques car sur UTM VirtuO ne laisser pas passer la commande, j'ai donc dû changer pour SCSI)

```
vdb    252:16    0      1G    0 disk
├─vdb1 252:17    0      50M    0 part /mnt/monext4
└─vdb2 252:18    0      50M    0 part [SWAP]
vdc    252:32    0      1G    0 disk
vdd    252:48    0      1G    0 disk
vde    252:64    0      1G    0 disk
vdf    252:80    0      1G    0 disk
arthur@ubuntu:~$
```

j'ai fait la **commande** :

```
sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3
/dev/vdc /dev/vdd /dev/vde
```

```

arthur@ubuntu:~$ sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/vdc /
dev/vdd /dev/vde

[sudo] Mot de passe de arthur :
Désolé, essayez de nouveau.
[sudo] Mot de passe de arthur :
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.

```

qui est utilisée pour créer un **ensemble** RAID 5 avec trois **disques** (vde, vdd, vdc). En exécutant cette commande avec des privilèges administratifs, **mdadm** crée un nouveau dispositif RAID nommé **/dev/md0**.

Le niveau 5 permet de **combiner redondance** et **performances** en répartissant les données et les informations de parité sur les trois disques, assurant ainsi une tolérance aux pannes

J'ai fait la commande **cat /proc/mdstat**

pour vérifier si mes disques sont comme je le souhaite

```

arthur@ubuntu:~$ cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid1
0]
md0 : active raid5 vde[3] vdd[1] vdc[0]
      2093056 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>

```

Ajouter le 4ème comme spare

La **commande** `sudo mdadm --manage /dev/md0 --add /dev/sdd` est utilisée pour **ajouter** le disque `/dev/sdd` comme disque de **réserve** (spare) à l'ensemble RAID 5, `/dev/md0`. En exécutant cette commande avec des privilèges **administratifs**, `mdadm` permet au disque d'être intégré dans la configuration RAID, prêt à prendre le **relais** automatiquement en cas de défaillance de l'un des disques actifs. Cela améliore la **tolérance** aux pannes de l'ensemble RAID, garantissant une **continuité** de service en cas de problème avec un disque.

```
arthur@ubuntu:~$ sudo mdadm --manage /dev/md0 --add /dev/vdf
mdadm: added /dev/vdf
```

Formater le volume RAID en XFS et nommer le FS "RAID5"

La commande `sudo mkfs.xfs -L RAID5 /dev/md0` formate le volume RAID `/dev/md0` avec le système de fichiers XFS et lui attribue le label "RAID5". En utilisant `mkfs.xfs`, on prépare le volume pour le stockage de données, offrant ainsi de bonnes performances et une gestion efficace des grands volumes.

```
arthur@ubuntu:~$ sudo mkfs.xfs -L RAID5 /dev/md0
log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/md0          isize=512    agcount=8, agsize=65408 blks
                        =               sectsz=512   attr=2, projid32bit=1
                        =               crc=1        finobt=1, sparse=1, rmapbt=0
                        =               reflink=1     bigtime=1 inobtcount=1 nrext6
4=0
data      =                bsize=4096   blocks=523264, imaxpct=25
                        =                sunit=128   swidth=256 blks
naming    =version 2        bsize=4096   ascii-ci=0, ftype=1
log       =internal log     bsize=4096   blocks=16384, version=2
                        =                sectsz=512   sunit=8 blks, lazy-count=1
realtime  =aucun            extsz=4096   blocks=0, rtextents=0
arthur@ubuntu:~$ sudo blkid /dev/md0
/dev/md0: LABEL="RAID5" UUID="044fc6bf-07be-4ea5-8814-2efa5504e0db" BLOCK_S
IZE="512" TYPE="xfs"
arthur@ubuntu:~$
```

Monter RAID5 dans `/mnt/raid5` et automatiser ce montage au démarrage via le fichier `/etc/fstab`

La **commande** `sudo mount /dev/md0 /mnt/raid5` monte le volume RAID `/dev/md0` dans le **répertoire** `/mnt/raid5`, rendant le système de fichiers accessible.

```
arthur@ubuntu:~$ sudo mkdir -p /mnt/raid5
arthur@ubuntu:~$ sudo mount /dev/md0 /mnt/raid5
```

Pour **automatiser** ce montage au démarrage, il faut ajouter la ligne suivante dans le **fichier** `/etc/fstab`:

```
/dev/md0 /mnt/raid5 xfs defaults 0 0
```

Cela permet à Linux de monter **automatiquement** le volume RAID à chaque **démarrage**, garantissant ainsi que le système de fichiers est toujours disponible sans **intervention manuelle**.

```
# 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-7T1VEovFa5SZQBduvYgRdu8qX7txFVm4PyMVvYfZ9E5qXj8QUCFpJds18espy9
# /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
/dev/swap.img none swap sw 0 0
/dev/vdb2 none swap sw 0 0
/dev/vdb1 /mnt/monext4 ext4 defaults 0 2
/dev/md0 /mnt/raid5 xfs defaults 0 0
share /media/HostShared 9p trans=virtio,version=9p2000.L,rw,_netdev,nofail 0
```

Créer un fichier `date.txt` contenant la date dans `/mnt/raid5`

La **commande** `date | sudo tee /mnt/raid5/date.txt` crée un fichier nommé `date.txt` dans le **répertoire** `/mnt/raid5`, contenant la date actuelle. En utilisant `date`, on obtient la **date système**, et `tee` permet d'écrire cette sortie dans le **fichier** tout en ayant les **privilèges** nécessaires grâce à `sudo`, ce qui est nécessaire pour écrire dans un répertoire monté en tant que système de fichiers.

```
arthur@ubuntu:~$ date | sudo tee /mnt/raid5/date.txt > /dev/null
arthur@ubuntu:~$
```

Montrer le résultat final à votre examinateur (capture des stockages Virtualbox et des commandes `df -h`, `lsblk --fs -e7`)

la **commande** `df -h`

affiche l'utilisation du disque de manière **lisible** pour l'homme (en Go, Mo, etc.), montrant les **points de montage** et **l'espace utilisé**. On y voit bien `/dev/md0` monté dans `raid5`

```
arthur@ubuntu:~$ df -h
Sys. de fichiers          Taille Utilisé Dispo Uti% Monté sur
tmpfs                    391M    2,0M  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/vdb1                 43M   152K   40M   1% /mnt/monext4
/dev/vda2                 2,0G   285M   1,6G  16% /boot
/dev/vda1                 1,1G    6,4M   1,1G   1% /boot/efi
tmpfs                    391M   132K   391M   1% /run/user/1000
/dev/md0                  2,0G    47M   1,9G   3% /mnt/raid5
arthur@ubuntu:~$
```

la **commande** `lsblk --fs -e7` liste **tous** les **périphériques** de blocs, leurs systèmes de fichiers, ainsi que des informations supplémentaires, en excluant les **périphériques** dont l'ID de type est 7. Ici on voit bien mes différents **périphériques** monté dans `raid5` ce qui signifie que le volume raid est **configuré** et **monté** correctement.

```
ext4 1.0 MONEXT4
2b5bb1e9-ff1d-4e5e-82e1-ad9bf7390aa3 39,2M 0%
/mnt/monext4
└─vdb2
  swap 1 3044129b-9cf9-4da4-a011-6505ee1d65ea
[SWAP]
vdc linux_ 1.2 ubuntu:0
c0f3692a-92ba-e41d-db77-a22bddf55a16
└─md0
  xfs RAID5 044fc6bf-07be-4ea5-8814-2efa5504e0db 1,9G 2%
/mnt/raid5
vdd linux_ 1.2 ubuntu:0
c0f3692a-92ba-e41d-db77-a22bddf55a16
└─md0
  xfs RAID5 044fc6bf-07be-4ea5-8814-2efa5504e0db 1,9G 2%
/mnt/raid5
vde linux_ 1.2 ubuntu:0
c0f3692a-92ba-e41d-db77-a22bddf55a16
└─md0
  xfs RAID5 044fc6bf-07be-4ea5-8814-2efa5504e0db 1,9G 2%
/mnt/raid5
vdf linux_ 1.2 ubuntu:0
c0f3692a-92ba-e41d-db77-a22bddf55a16
└─md0
  xfs RAID5 044fc6bf-07be-4ea5-8814-2efa5504e0db 1,9G 2%
/mnt/raid5
arthur@ubuntu:~$
```

Avec la machine en fonctionnement, se mettre en visualisation live des logs système

j'utilise la commande : `sudo journalctl -f`

Retirer un des disques actif du RAID5 à chaud

- via les paramètres Virtualbox, ou
- via la commande suivante si vous n'avez pas trouvé le Hotswap précédemment (adaptez sdX à votre besoin) : `echo 1 | sudo tee /sys/block/sdX/device/delete`

La **commande** `echo 1 | sudo tee /sys/block/sda/device/delete` permet de retirer un **disque actif** d'un ensemble RAID 5 à chaud.

- **echo 1** génère la valeur **1** pour indiquer la **suppression**.
- Le **|** redirige cette sortie vers la commande suivante.
- `sudo tee /sys/block/sda/device/delete` écrit cette valeur dans le fichier **delete** du disque spécifié (**/dev/sda**), où **sda** est le disque que je souhaite supprimer

```
arthur@ubuntu:~$ echo 1 | sudo tee /sys/block/sda/device/delete
[sudo] Mot de passe de arthur :
1
arthur@ubuntu:~$
```

Cette commande permet de **retirer** le disque **sans** redémarrer le système, maintenant ainsi la **disponibilité** du service. Le RAID fonctionnera en mode dégradé jusqu'à ce qu'un disque de remplacement soit ajouté.

Constater ce qu'il se passe

- dans les logs système (visualiser en live),
- pour notre RAID avec mdadm (utiliser watch en parallèle des logs),
- pour notre FS qui doit rester fonctionnel...

Ici, on voit mes **logs** après l'exécution de la commande `echo 1 | sudo tee /sys/block/sda/device/delete` on voit bien que tout est comme on l'attend et que le disque est bel est bien **supprimé**

```
COMMAND=/usr/bin/tee /sys/block/sda/device/delete
oct. 25 15:36:28 ubuntu sudo[3488]: pam_unix(sudo:session): session opened for user root(u
id=0) by (uid=1000)
oct. 25 15:36:28 ubuntu sudo[3488]: pam_unix(sudo:session): session closed for user root
oct. 25 15:36:28 ubuntu kernel: sd 0:0:0:0: [sda] Synchronizing SCSI cache
oct. 25 15:36:33 ubuntu kernel: md/raid:md0: Disk failure on sda, disabling device.
oct. 25 15:36:33 ubuntu kernel: md/raid:md0: Operation continuing on 2 devices.
oct. 25 15:36:33 ubuntu kernel: md: recovery of RAID array md0
oct. 25 15:36:33 ubuntu udisksd[930]: Unable to resolve /sys/devices/virtual/block/md0/md/
dev-sda/block symlink
oct. 25 15:36:33 ubuntu udisksd[930]: Unable to resolve /sys/devices/virtual/block/md0/md/
dev-sda/block symlink
oct. 25 15:36:34 ubuntu mdadm[3511]: sh: 1: /usr/sbin/sendmail: not found
oct. 25 15:36:38 ubuntu kernel: md: md0: recovery done.
oct. 25 15:36:41 ubuntu spice-vdagentd[1974]: invalid message size for VDAgentMonitorsConf
in
```

Montrer le résultat intermédiaire à votre examinateur (capture mdadm pendant la reconstruction)

je n'ai pas le screen mais il y avait écrit **building** a côté de mon disque cela confirme que le système est en train de reconstruire le RAID

Après reconstruction du RAID, rebrancher votre disque (voir annexe si vous n'avez pas le Hotswap Virtualbox)

```
arthur@ubuntu:~$ ls -l /sys/class/block/s??
lrwxrwxrwx 1 root root 0 oct. 25 15:27 /sys/class/block/sdb -> ../../devices/pci0000:00/0000:00:08.0/host0/target0:0:1:0:0:1:0/block/sdb
lrwxrwxrwx 1 root root 0 oct. 25 15:27 /sys/class/block/sdc -> ../../devices/pci0000:00/0000:00:08.0/host0/target0:0:2:0:0:2:0/block/sdc
lrwxrwxrwx 1 root root 0 oct. 25 15:27 /sys/class/block/sdd -> ../../devices/pci0000:00/0000:00:08.0/host0/target0:0:3:0:0:3:0/block/sdd
lrwxrwxrwx 1 root root 0 oct. 25 15:27 /sys/class/block/sr0 -> ../../devices/pci0000:00/0000:00:04.0/usb1/1-4/1-4.1/1-4.1:1.0/host1/target1:0:0:1:0:0:0/block/sr0
arthur@ubuntu:~$ echo "- - -" | sudo tee /sys/class/scsi_host/host0/scan
[sudo] Mot de passe de arthur :
- - -
arthur@ubuntu:~$ dmesg
dmesg: échec de lecture du tampon de noyau: Opération non permise
arthur@ubuntu:~$ sudo dmesg
[ 0.000000] Booting Linux on physical CPU 0x0000000000 [0xc10f0000]
```

Vérifier la présence des fichiers

On voit bien que mon disque sda est **représent** parmi les autres périphériques

```
loop14      7:14    0   33,7M    1 loop  /snap/snapd/21761
loop15      7:15    0   35,2M    1 loop  /snap/snapd/20674
sda          8:0      0    1G    0 disk
sdb          8:16    0    1G    0 disk
└md0        9:0      0    2G    0 raid5  /mnt/raid5
sdc          8:32    0    1G    0 disk
└md0        9:0      0    2G    0 raid5  /mnt/raid5
sdd          8:48    0    1G    0 disk
└md0        9:0      0    2G    0 raid5  /mnt/raid5
sr0         11:0    1  1024M    0 rom
vda         252:0    0   80G    0 disk
├vda1       252:1    0    1G    0 part  /boot/efi
├vda2       252:2    0    2G    0 part  /boot
└vda3       252:3    0   76,9G   0 part
  └ubuntu--vg-ubuntu--lv
    253:0    0   38,5G    0 lvm    /var/snap/firefox/common/host-hunspell
```

Que faire de ce disque ? Mettre votre solution en oeuvre

Pour la suite, il est **important** de **vérifier** l'état du RAID avec `cat /proc/mdstat` pour s'assurer que le disque est reconnu et en cours de reconstruction. Enfin, il est recommandé de faire une capture d'écran de l'état du RAID pour montrer que le disque a été intégré avec succès et que le système fonctionne normalement.

Transformer le disque spare en disque actif

j'ai utiliser cette **commande** :

```
arthur@ubuntu:~$ sudo mdadm --grow /dev/md0 --raid-devices=
```

et Tadam ! on voit bien que `/dev/sda` est actif

```
Update Time : Fri Oct 25 17:10:18 2024
State : clean
Active Devices : 4
Working Devices : 4
Failed Devices : 0
Spare Devices : 0

Layout : left-symmetric
Chunk Size : 512K

Consistency Policy : resync

Name : ubuntu:0 (local to host ubuntu)
UUID : e5bb2a88:47e8b093:303804ed:be506ee5
Events : 64

Number Major Minor RaidDevice State
  4      8     48        0  active sync  /dev/sdd
  1      8     16        1  active sync  /dev/sdb
  3      8     32        2  active sync  /dev/sdc
  5      8      0        3  active sync  /dev/sda
arthur@ubuntu:~$
```

Agrandir le FS XFS en conséquence

ensuite j'ai fait la **commande** `sudo xfs growfs /mnt/raid5/`

La **commande** permet **d'agrandir** un système de fichiers XFS monté sur le point de montage `/mnt/raid5/`. En utilisant `sudo`, tu exécutes l'opération avec les privilèges administratifs. Cela permet au système de fichiers d'utiliser tout l'espace supplémentaire alloué au volume RAID sans avoir besoin de le démonter, ce qui est pratique pour gérer des volumes de données en temps réel.

```
arthur@ubuntu:~$ sudo xfs growfs /mnt/raid5/
sudo: xfs : commande introuvable
arthur@ubuntu:~$ sudo xfs_growfs /mnt/raid5/
meta-data=/dev/md0          isize=512    agcount=8, agsize=65408 blks
                =          sectsz=512   attr=2, projid32bit=1
                =          crc=1        finobt=1, sparse=1, rmapbt=0
                =          reflink=1    bigtime=1 inobtcount=1 nnext64=
0
data      =                  bsize=4096   blocks=523264, imaxpct=25
                =          sunit=128    swidth=256 blks
naming    =version 2        bsize=4096   ascii-ci=0, ftype=1
log       =internal log    bsize=4096   blocks=16384, version=2
                =          sectsz=512   sunit=8 blks, lazy-count=1
realtime  =aucun           extsz=4096   blocks=0, rtextents=0
blocs de données modifiés de 523264 à 784896
```