Emmanuel-Joyce Ntsia, Gauvain BOICHÉ, Joris Stavroulakis, Léo Jarry,

Virtualisation, Containérisation d'OS, Stockage évolutif & résilient

Pour démarrer l'évaluation on est parti sur une VM avec ces spécifications :

- Debian 13.1.0-amd64-netinst
- 8 Coeurs Logiques
- 8 Go RAM
- 20 Go d'espace disque (disque 1, le disque système)

À partir de là on a commencé par installer toutes les dépendances nécessaires

1.1 Installation des paquets pour libvirt / KVM / Qemu

On commence par installer et configurer tout ce qui sera nécessaire pour la virtualisation :

```
sudo apt install bridge-utils cpu-checker libvirt-clients libvirt-daemon-system virtinst qemu-kvm virt-manager -y
```

```
sudo usermod -aG kvm $USER
sudo usermod -aG libvirt $USER
```


1.2 Installation des paquets pour la containérisation LXC

Ensuite on installe tout ce qui sera nécessaire pour la conteneurisation :

```
sudo apt install lxc lxctl lxc-templates -y
```


1.3 Installation du paquet mdadm pour le RAID

Enfin on installe **mdadm** pour créer et gérer le RAID

```
sudo apt install mdadm -y
```


2.1 Ajout des 4 disques pour le RAID5 (3 disques actifs + un hotspare)

On a choisi un RAID de type 5 pour la tolérance de panne **et** les performances, on est parti sur des disques de 4Go, ce qui s'avèrera (voir la suite) ne pas être assez, donc il faudra agrandir par la suite :

2.2 Mise en place du RAID5

D'abord on repère les disques qu'on va mettre dans le RAID :

```
sudo lsblk
```

Ensuite on créé ce dernier en précisant le type et le nom :

```
sudo mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd
```


On ajoute le disque *sde* en tant que hotspare pour notre RAID :

```
sudo mdadm --add /dev/md1 /dev/sde
```


On vérifie que tout est en place :

```
sudo mdadm --detail /dev/md1
```


On voit que le RAID5 avec *hotspare* est bien en place, on peut passer à la partie LVM

2.3 Mise en place du LVM

Ici on commence par déclarer le pv qu'on va utiliser, notre RAID :

```
sudo pvcreate /dev/md1
```


Ensuite on créé le volume group nommé *eval* et les deux logical volume *vms* et *conteneurs* :

```
sudo vgcreate eval /dev/md1
```

```
sudo lvcreate -l 50%FREE -n vms eval
```

```
sudo lvcreate -l 100%FREE -n conteneurs eval
```


Ensuite on formate les volumes pour mettre le système de fichiers *ext4* :

```
sudo mkfs.ext4 /dev/mapper/eval-vms
```

```
sudo mkfs.ext4 /dev/mapper/eval-conteneurs
```


Puis on ajoute les labels :

```
sudo tune2fs -L vmsfs /dev/mapper/eval-vms
```

```
sudo tune2fs -L conteneursfs /dev/mapper/eval-conteneurs
```


On vérifie qu'ils sont bien mis :

2.4 Mise en place du montage au démarrage

On prépare les dossiers pour le montage de nos volumes :

```
sudo mkdir -p /mnt/vms /mnt/conteneurs
```


On modifie le fstab pour que le montage se fasse au démarrage du système :

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

```
  GNU nano 8.4                          /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=ba20becf-b825-48ed-9ca4-a81f4c622933 /              ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=b1d8d1e4-71bd-47db-9736-3e7d3133f4f4 none           swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto    0       0

LABEL=vmsfs /mnt/vms ext4 defaults 0 0
LABEL=conteneursfs /mnt/conteneurs ext4 defaults 0 0
```

2.5 Montage des volumes

Ici on le monte directement (plutôt que de devoir redémarrer), pour ça on oublie pas de relancer le service :

```
sudo systemctl daemon-reload
```

```
student@livecampus:~$ sudo systemctl daemon-reload
```

On monte les volumes sur les dossiers précédemment créés :

```
sudo mount -t ext4 /dev/mapper/eval-vms /mnt/vms
sudo mount -t ext4 /dev/mapper/eval-conteneurs /mnt/conteneurs
```

```
student@livecampus:~$ sudo mount -t ext4 /dev/mapper/eval-vms /mnt/vms/
student@livecampus:~$ sudo mount -t ext4 /dev/mapper/eval-conteneurs /mnt/conteneurs/
```

3.1 Installation des conteneurs LXC

Maintenant que tout est préparé on peut commencer à faire les conteneurs et les vms, on a décidé de commencer par les conteneurs

3.1.1 Le conteneur Debian

On installe le conteneur Debian :

```
sudo lxc-create -n debianLXC -t download -- -d debian -r bullseye
```

```
student@livecampus:~/iso$ sudo lxc-create -n debianLXC -t download -- -d debian -r bullseye
Downloading the image index

---
DIST            RELEASE     ARCH    VARIANT  BUILD
---
debian          bullseye    amd64   default  20251024_05:24
debian          bullseye    arm64   default  20251024_05:24
debian          bullseye    armhf   default  20251024_05:24
---

Architecture:
amd64

Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created a Debian bullseye amd64 (20251024_05:24) container.

To enable SSH, run: apt install openssh-server
No default root or user password are set by LXC.
student@livecampus:~/iso$ sudo lxc-ls
debianLXC
```

On crée le dossier qui servira de volume rattaché à ce conteneur :

```
sudo mkdir /mnt/conteneurs/debianlxc
```

```
student@livecampus:~/iso$ sudo mkdir /mnt/conteneurs/debianlxc
```

On va modifier le fichier de config pour le rattacher justement :

```
sudo nano /var/lib/lxc/debianLXC/config
```

```
  GNU nano 8.4                    /var/lib/lxc/debianLXC/config *
# Template used to create this container: /usr/share/lxc/templates/lxc-download
# Parameters passed to the template: -d debian -r bullseye
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)


# Distribution configuration
lxc.include = /usr/share/lxc/config/common.conf
lxc.arch = linux64

# Container specific configuration
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.rootfs.path = dir:/var/lib/lxc/debianLXC/rootfs
lxc.uts.name = debianLXC

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up

# Storage attachment
lxc.mount.entry = /mnt/conteneurs/debianlxc storage none bind,create=dir 0 0

student@livecampus:~/iso$ sudo lxc-start debianLXC
student@livecampus:~/iso$ sudo lxc-attach debianLXC
root@debianLXC:/# ls
bin   dev  home  lib64  mnt  proc  run   srv      sys  usr
boot  etc  lib   media  opt  root  sbin  storage  tmp  var
root@debianLXC:/# cd storage
root@debianLXC:/storage# touch test.debianlxc
root@debianLXC:/storage# exit
exit
student@livecampus:~/iso$ ls /mnt/conteneurs/debianlxc/
test.debianlxc
```

3.1.2 Le conteneur Ubuntu

On installe le conteneur Ubuntu :

```
sudo lxc-create -n ubuntuLXC -t download -- -d ubuntu -r noble
```

```
student@livecampus:~/iso$ sudo lxc-create -n ubuntuLXC -t download -- -d ubuntu -r noble
Downloading the image index

---
DIST            RELEASE     ARCH     VARIANT  BUILD
---
ubuntu          noble       amd64    default  20251022_21:14
ubuntu          noble       arm64    default  20251022_23:43
ubuntu          noble       armhf    default  20251023_21:04
ubuntu          noble       riscv64  default  20251022_21:31
---

Architecture:
amd64

Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created an Ubuntu noble amd64 (20251022_21:14) container.

To enable SSH, run: apt install openssh-server
No default root or user password are set by LXC.
```

On créé le dossier qui servira de volume rattaché à ce conteneur :

```
sudo mkdir /mnt/conteneurs/ubuntulxc
```

```
student@livecampus:~/iso$ sudo mkdir /mnt/conteneurs/ubuntulxc
```

On va modifier le fichier de config pour le rattacher justement :

```
sudo nano /var/lib/lxc/ubuntuLXC/config
```

```
  GNU nano 8.4                    /var/lib/lxc/ubuntuLXC/config
# Template used to create this container: /usr/share/lxc/templates/lxc-download
# Parameters passed to the template: -d ubuntu -r noble
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)


# Distribution configuration
lxc.include = /usr/share/lxc/config/common.conf
lxc.arch = linux64

# Container specific configuration
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.rootfs.path = dir:/var/lib/lxc/ubuntuLXC/rootfs
lxc.uts.name = ubuntuLXC

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up

# Storage attachment
lxc.mount.entry = /mnt/conteneurs/ubuntulxc storage none bind,create=dir 0 0
student@livecampus:~/iso$ sudo nano /var/lib/lxc/ubuntuLXC/config
student@livecampus:~/iso$ sudo nano /var/lib/lxc/ubuntuLXC/config
student@livecampus:~/iso$ sudo lxc-start ubuntuLXC && sudo lxc-attach ubuntuLXC
root@ubuntuLXC:/# ls
bin                dev   lib              media  proc  sbin               storage  usr
bin.usr-is-merged  etc   lib64            mnt    root  sbin.usr-is-merged  sys     var
boot               home  lib.usr-is-merged  opt    run   srv                tmp
root@ubuntuLXC:/# cd storage
root@ubuntuLXC:/storage# ls
root@ubuntuLXC:/storage# toush ubuntu.test_write
bash: toush: command not found
root@ubuntuLXC:/storage# touch ubuntu.test_write
root@ubuntuLXC:/storage# ls
ubuntu.test_write
root@ubuntuLXC:/storage# exit
exit
student@livecampus:~/iso$ ls /mnt/conteneurs/ubuntulxc/
ubuntu.test_write
```

3.1.3 Le conteneur Fedora

On installe le conteneur Fedora :

```
sudo lxc-create -n fedoraLXC -t download -- d fedora -r 42
```



```
student@livecampus:~/iso$ sudo lxc-create -n fedoraLXC -t download -- -d fedora -r adams
Downloading the image index

---
DIST            RELEASE    ARCH    VARIANT  BUILD
---
---

Architecture:
x

Downloading the image index
ERROR: Couldn't find a matching image
lxc-create: fedoraLXC: ../src/lxc/lxccontainer.c: create_run_template: 1601 Failed to create c
ontainer from template
lxc-create: fedoraLXC: ../src/lxc/tools/lxc_create.c: lxc_create_main: 318 Failed to create co
ntainer fedoraLXC
student@livecampus:~/iso$ sudo lxc-create -n fedoraLXC -t download -- -d fedora -r 42
Downloading the image index

---
DIST            RELEASE    ARCH    VARIANT  BUILD
---
fedora          42         amd64   default  20251023_21:25
fedora          42         arm64   default  20251023_23:39
---

Architecture:
amd64

Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created a Fedora 42 x86_64 (20251023_21:25) container.
student@livecampus:~/iso$ sudo lxc-ls
debianLXC fedoraLXC ubuntuLXC
```

On crée le dossier qui servira de volume rattaché à ce conteneur :

```
sudo mkdir /mnt/conteneurs/fedoralxc
```



```
student@livecampus:~/iso$ sudo mkdir /mnt/conteneurs/fedoralxc
```

On va modifier le fichier de config pour le rattacher justement :

```
sudo nano /var/lib/lxc/fedoraLXC/config
```

```
  GNU nano 8.4                      /var/lib/lxc/fedoraLXC/config *
# Template used to create this container: /usr/share/lxc/templates/lxc-download
# Parameters passed to the template: -d fedora -r 42
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)


# Distribution configuration
lxc.include = /usr/share/lxc/config/common.conf
lxc.arch = x86_64

# Container specific configuration
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.rootfs.path = dir:/var/lib/lxc/fedoraLXC/rootfs
lxc.uts.name = fedoraLXC

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up

# Storage attachment
lxc.mount.entry = /mnt/conteneurs/fedoralxc storage none bind,create=dir 0 0
student@livecampus:~/iso$ sudo nano /var/lib/lxc/fedoraLXC/config
student@livecampus:~/iso$ sudo lxc-start fedoraLXC && sudo lxc-attach fedoraLXC
[root@fedoraLXC /]# ls
afs  boot  etc   lib    media  opt   root  sbin  storage  tmp  var
bin  dev   home  lib64  mnt    proc  run   srv   sys      usr
[root@fedoraLXC /]# cd storage/
[root@fedoraLXC storage]# ls
[root@fedoraLXC storage]# touch fedora.non_cest_juste_de_la_redondance_a_ce_stade
[root@fedoraLXC storage]# exit
exit
student@livecampus:~/iso$ ls /mnt/conteneurs/fedoralxc/
fedora.non_cest_juste_de_la_redondance_a_ce_stade
```

3.2 Installation des VMS

Maintenant que les conteneurs sont installés et configurés on passe aux VMs

3.2.1 La VM Debian

On commence à créer un dossier dans lequel on rangera le disque de la VM (ainsi que son iso au passage) :

```
sudo mkdir debian
```

```
student@livecampus:/mnt/vms$ sudo mkdir debian
student@livecampus:/mnt/vms$ ls
debian  lost+found
student@livecampus:/mnt/vms$ cd debian/
student@livecampus:/mnt/vms/debian$ _
```

On télécharge le fichier iso :

```
sudo wget https://cdimage.debian.org/mirror/cdimage/archive/11.11.0/amd64/iso-cd/debian-11.11.0-
amd64-netinst.iso
```

```
student@livecampus:/mnt/vms/debian$ sudo wget https://cdimage.debian.org/mirror/cdimage/archiv
e/11.11.0/amd64/iso-cd/debian-11.11.0-amd64-netinst.iso
--2025-10-24 11:55:51--  https://cdimage.debian.org/mirror/cdimage/archive/11.11.0/amd64/iso-c
d/debian-11.11.0-amd64-netinst.iso
Resolving cdimage.debian.org (cdimage.debian.org) ... 194.71.11.165, 194.71.11.173, 2001:6b0:19
::173, ...
Connecting to cdimage.debian.org (cdimage.debian.org)|194.71.11.165|:443 ... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://saimei.ftp.acc.umu.se/mirror/cdimage/archive/11.11.0/amd64/iso-cd/debian-11.
11.0-amd64-netinst.iso [following]
--2025-10-24 11:55:51--  https://saimei.ftp.acc.umu.se/mirror/cdimage/archive/11.11.0/amd64/is
o-cd/debian-11.11.0-amd64-netinst.iso
Resolving saimei.ftp.acc.umu.se (saimei.ftp.acc.umu.se) ... 194.71.11.138, 2001:6b0:19::138
Connecting to saimei.ftp.acc.umu.se (saimei.ftp.acc.umu.se)|194.71.11.138|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 408944640 (390M) [application/x-iso9660-image]
Saving to: 'debian-11.11.0-amd64-netinst.iso'

debian-11.11.0-amd64-ne 100%[===================>] 390.00M  70.7MB/s    in 5.9s

2025-10-24 11:55:57 (65.6 MB/s) - 'debian-11.11.0-amd64-netinst.iso' saved [408944640/40894464
0]

student@livecampus:/mnt/vms/debian$ ls -glAiF
total 399560
130820 -rw-r--r-- 1 root          408944640 Aug 31  2024 debian-11.11.0-amd64-netinst.iso
130819 -rw-r--r-- 1 libvirt-qemu     196640 Oct 24 11:33 debian.qcow2
```

On crée le disque (qui s'avèrera trop petit, voir la suite) :

```
sudo qemu-img create -f qcow2 debian.qcow2 2G
```

On lance l'installation avec **virt-install** :

```
sudo virt-install \
--name debianVM \
--os-variant=debian11 \
--ram 2048 \
--vcpus 2 \
--location /mnt/vms/debian/debian-11.11.0-amd64-netinst.iso \
--disk debian.qcow2 \
--graphics none \
--console pty,target_type=serial \
--extra-args='console=ttyS0,115200n8 serial'
```
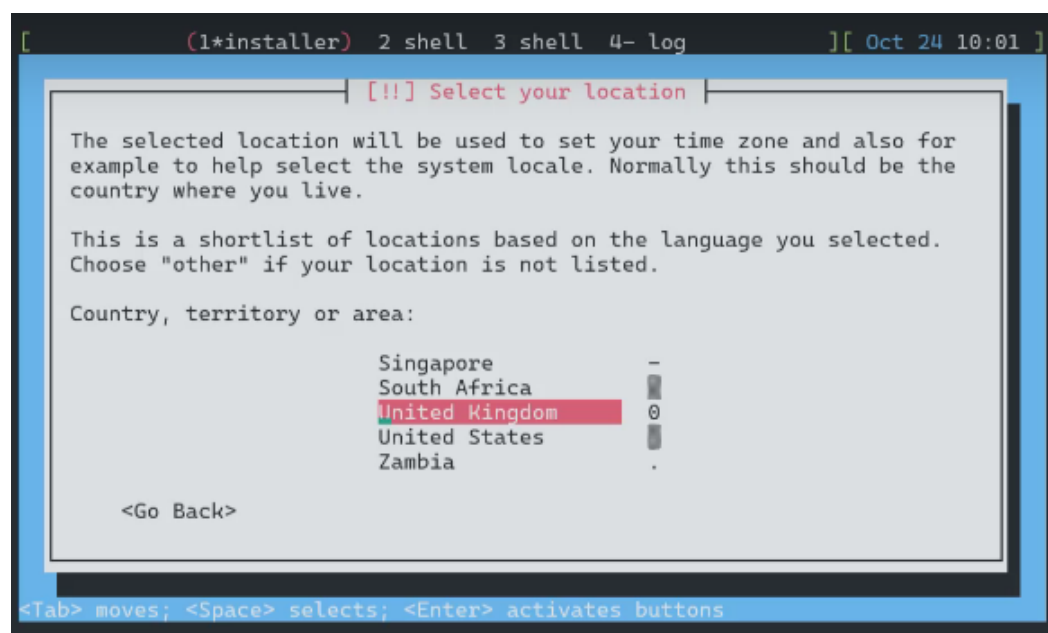

On a manqué de place pour l'installation, donc on rajoute deux disques de 4Go au raid et au LVM pour recommencer :

On ajoute nos deux disques au RAID (Après le reboot le md1 a été renommé par le système en md127) :

```
sudo mdadm --add /dev/md127 /dev/sdf /dev/sdg
```


On oublie pas cette commande, sinon nos disques seront juste en spare :

```
sudo mdadm --grow /dev/md127 --raid-devices=5
```

```
student@livecampus:~$ sudo mdadm --grow /dev/md127 --raid-devices=5
student@livecampus:~$ sudo mdadm --detail /dev/md127
/dev/md127:
           Version : 1.2
     Creation Time : Fri Oct 24 10:36:42 2025
        Raid Level : raid5
        Array Size : 8378368 (7.99 GiB 8.58 GB)
     Used Dev Size : 4189184 (4.00 GiB 4.29 GB)
      Raid Devices : 5
     Total Devices : 6
       Persistence : Superblock is persistent

     Intent Bitmap : Internal

       Update Time : Fri Oct 24 12:29:54 2025
             State : clean, reshaping
    Active Devices : 5
   Working Devices : 6
    Failed Devices : 0
     Spare Devices : 1

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : bitmap

    Reshape Status : 3% complete
     Delta Devices : 2, (3→5)

              Name : livecampus:1  (local to host livecampus)
              UUID : 3898e4e6:e3fb1ff5:0525bc7b:5617ad16
            Events : 46

    Number   Major   Minor   RaidDevice State
       0       8       16        0      active sync   /dev/sdb
       1       8       32        1      active sync   /dev/sdc
       3       8       48        2      active sync   /dev/sdd
       6       8       96        3      active sync   /dev/sdg
       5       8       80        4      active sync   /dev/sdf

       4       8       64        -      spare   /dev/sde
student@livecampus:~$
```

*Et ensuite il faut bien sûr étendre le LVM (**après avoir attendu que le reshape du RAID ait fini**)*

On commence par resize le pv :

```
sudo pvresize /dev/md127
```

```
student@livecampus:~$ sudo pvresize /dev/md127
  Physical volume "/dev/md127" changed
  1 physical volume(s) resized or updated / 0 physical volume(s) not resized
student@livecampus:~$ sudo vgdisplay
  --- Volume group ---
  VG Name               eval
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <15.98 GiB
  PE Size               4.00 MiB
  Total PE              4090
  Alloc PE / Size       2045 / <7.99 GiB
  Free  PE / Size       2045 / <7.99 GiB
  VG UUID               NSw6j0-ORcO-BqlC-GMst-2gzW-7ctF-fYUibw
```

On étend ensuite notre volume dédié au VM pour lui accorde 100% du nouvel espace libre :

```
sudo lvextend -l +100%FREE --resizefs /dev/mapper/eval-vms
```

```
student@livecampus:~$ sudo lvextend -l +100%FREE --resizefs /dev/mapper/eval-vms
  File system ext4 found on eval/vms mounted at /mnt/vms.
  Size of logical volume eval/vms changed from 3.99 GiB (1022 extents) to 11.98 GiB (3067 exte
nts).
  Extending file system ext4 to 11.98 GiB (12863930368 bytes) on eval/vms ...
resize2fs /dev/eval/vms
resize2fs 1.47.2 (1-Jan-2025)
Filesystem at /dev/eval/vms is mounted on /mnt/vms; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/eval/vms is now 3140608 (4k) blocks long.

resize2fs done
  Extended file system ext4 on eval/vms.
  Logical volume eval/vms successfully resized.
student@livecampus:~$ df -h
Filesystem                    Size  Used Avail Use% Mounted on
udev                          3.9G     0  3.9G   0% /dev
tmpfs                         795M  704K  794M   1% /run
/dev/sda1                      19G  4.6G   13G  27% /
tmpfs                         3.9G     0  3.9G   0% /dev/shm
tmpfs                         5.0M     0  5.0M   0% /run/lock
tmpfs                         1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs                         3.9G     0  3.9G   0% /tmp
/dev/mapper/eval-vms           12G  1.5G  9.8G  13% /mnt/vms
/dev/mapper/eval-conteneurs   3.9G  1.1M  3.7G   1% /mnt/conteneurs
tmpfs                         1.0M     0  1.0M   0% /run/credentials/getty@tty1.service
tmpfs                         795M   12K  795M   1% /run/user/1000
```

Ensuite on undefine le debianVM créé précédemment pour refaire l'installation à zéro :

(Cette étape est nécessaire pour pouvoir utiliser le disque avec le même nom qu'avant, sinon virt croit que le disque est déjà utilisé par une autre VM)

```
sudo virsh undefine debianVM
```

Puis on supprime le disque :

```
sudo rm debian.qcow2
```

Et on le recrée avec une plus grande taille :

```
sudo qemu-img create -f qcow2 debian.qcow2 5G
```

Et on refait l'installation avec la commande virt-install utilisée au-dessus, voici le résultat :

3.2.2 La VM Ubuntu

On commence à créer un dossier dans lequel on rangera le disque de la VM (ainsi que son iso au passage) :

```
sudo mkdir /mnt/vms/ubuntu
```


On télécharge le fichier iso :

```
sudo wget https://cdimage.ubuntu.com/ubuntu-legacy-server/releases/20.04/release/ubuntu-20.04.1-legacy-server-amd64.iso
```

```
student@livecampus:/mnt/vms/ubuntu$ sudo wget https://cdimage.ubuntu.com/ubuntu-legacy-server/
releases/20.04/release/ubuntu-20.04.1-legacy-server-amd64.iso
--2025-10-24 14:32:48--  https://cdimage.ubuntu.com/ubuntu-legacy-server/releases/20.04/releas
e/ubuntu-20.04.1-legacy-server-amd64.iso
Resolving cdimage.ubuntu.com (cdimage.ubuntu.com)... 91.189.91.124, 91.189.91.123, 185.125.190
.37, ...
Connecting to cdimage.ubuntu.com (cdimage.ubuntu.com)|91.189.91.124|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 855638016 (816M) [application/x-iso9660-image]
Saving to: 'ubuntu-20.04.1-legacy-server-amd64.iso'

ubuntu-20.04.1-legacy-s 100%[===================>]  816.00M  30.7MB/s    in 28s

2025-10-24 14:33:17 (29.4 MB/s) - 'ubuntu-20.04.1-legacy-server-amd64.iso' saved [855638016/85
5638016]
```

On créé notre disque de 6Go :

```
sudo qemu-img create -f qcow2 ubuntu.qcow2 6G
```

```
student@livecampus:/mnt/vms/ubuntu$ sudo qemu-img create -f qcow2 ubuntu.qcow2 6G
Formatting 'ubuntu.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib
size=6442450944 lazy_refcounts=off refcount_bits=16
student@livecampus:/mnt/vms/ubuntu$ ls
total 1452680
523266 -rw-r--r-- 1 root 1487339520 Mar 15  2023 ubuntu-20.04.6-live-server-amd64.iso
523267 -rw-r--r-- 1 root     196704 Oct 24 14:28 ubuntu.qcow2
student@livecampus:/mnt/vms/ubuntu$ df -h
Filesystem                    Size  Used Avail Use% Mounted on
udev                          3.9G     0  3.9G   0% /dev
tmpfs                         795M  708K  794M   1% /run
/dev/sda1                      19G  4.6G   13G  27% /
tmpfs                         3.9G     0  3.9G   0% /dev/shm
tmpfs                         5.0M     0  5.0M   0% /run/lock
tmpfs                         1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs                         3.9G     0  3.9G   0% /tmp
/dev/mapper/eval-vms           12G  4.3G  6.9G  39% /mnt/vms
/dev/mapper/eval-conteneurs   3.9G  1.1M  3.7G   1% /mnt/conteneurs
tmpfs                         1.0M     0  1.0M   0% /run/credentials/getty@tty1.service
tmpfs                         795M   16K  795M   1% /run/user/1000
```

On lance l'installation :

```
sudo virt-install \
--name ubuntuVM \
--os-variant=ubuntu20.04 \
--ram 4096 \
--vcpus 4 \
--location /mnt/vms/ubuntu/ubuntu-20.04.1-legacy-server-amd64.iso \
--disk ubuntu.qcow2 \
--graphics none \
--console pty,target_type=serial \
--extra-args='console=ttyS0,115200n8 serial'
```

```
student@livecampus:/mnt/vms/ubuntu$ sudo virt-install \
    --name ubuntuVM \
    --os-variant=ubuntu20.04 \
    --ram 4096 \
    --vcpus 4 \
    --location /mnt/vms/ubuntu/ubuntu-20.04.1-legacy-server-amd64.iso \
    --disk ubuntu.qcow2 \
    --graphics none \
    --console pty,target_type=serial \
    --extra-args='console=ttyS0,115200n8 serial'

Starting install...
Retrieving 'vmlinuz'                                                    |  11 MB  00:00:00
Retrieving 'initrd.gz'                                                  |  17 MB  00:00:00
```
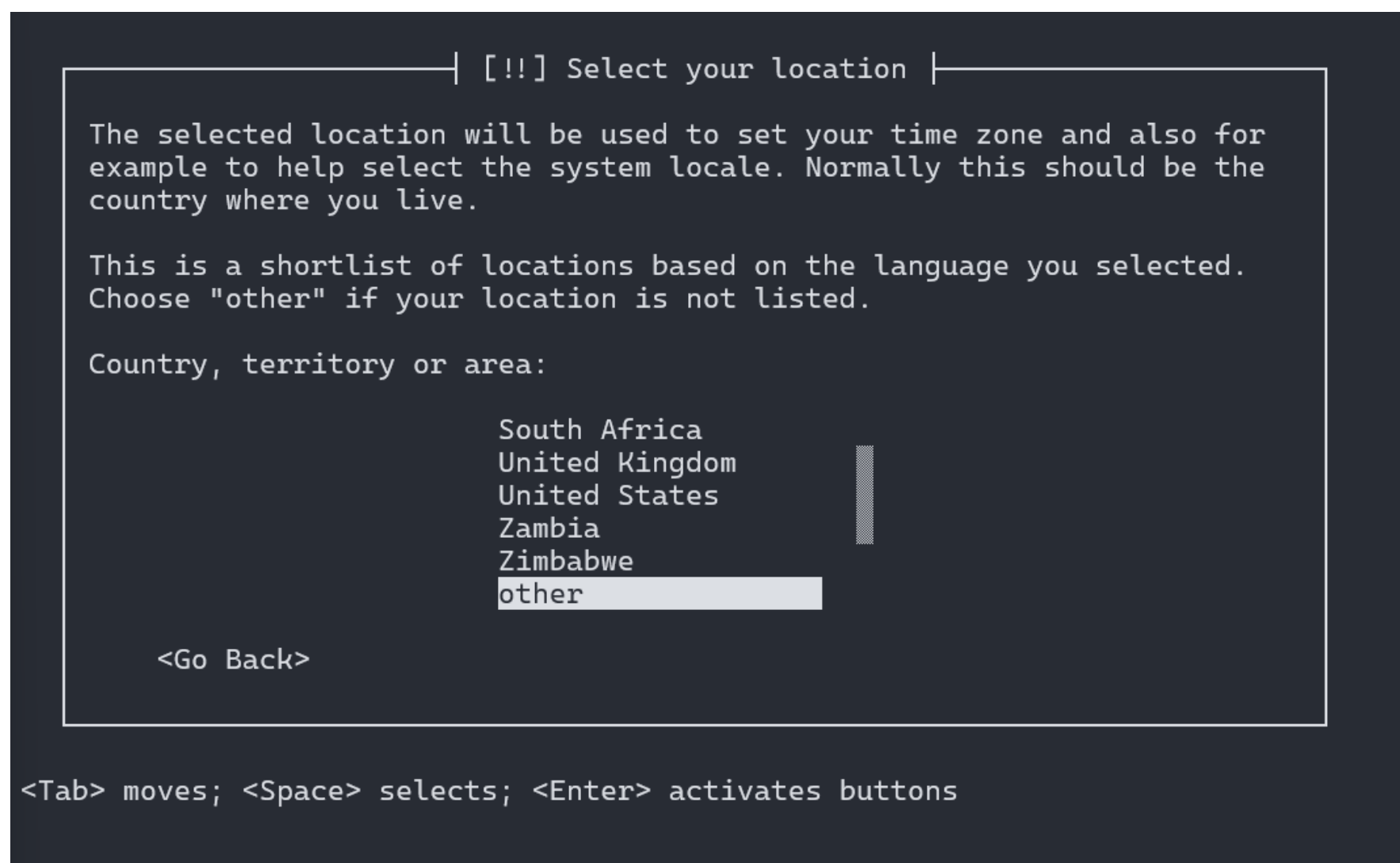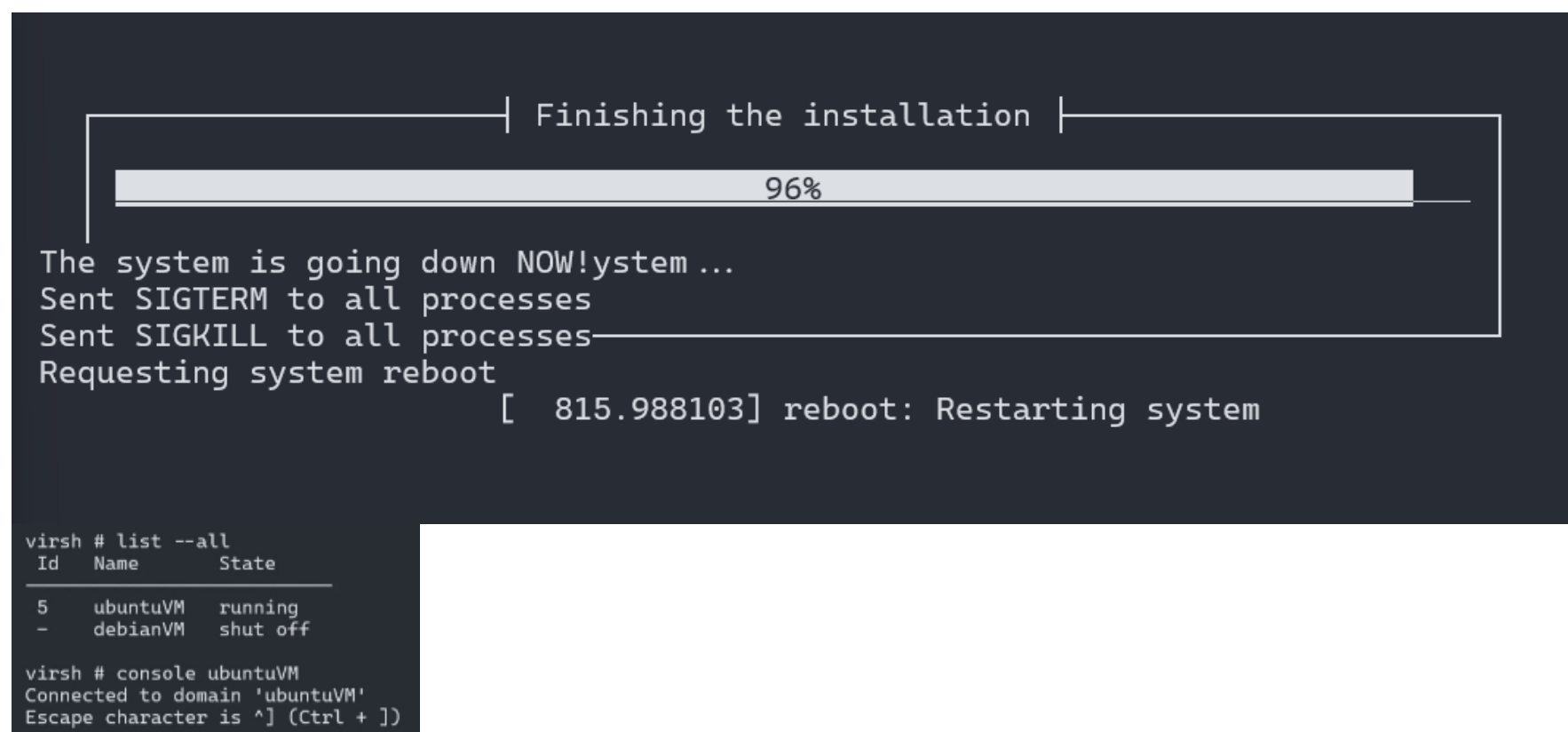
```
                         ┤ [!!] Select your location ├

    The selected location will be used to set your time zone and also for
    example to help select the system locale. Normally this should be the
    country where you live.

    This is a shortlist of locations based on the language you selected.
    Choose "other" if your location is not listed.

    Country, territory or area:

                          South Africa
                          United Kingdom
                          United States
                          Zambia
                          Zimbabwe
                          other

          <Go Back>



 <Tab> moves; <Space> selects; <Enter> activates buttons
```

L'installation du ubuntu va jusqu'au bout, mais après on a un écran "noir" comme mentionné par le formateur, donc impossible d'aller plus loin :

```
                         ┤ Finishing the installation ├
    ┌──────────────────────────────────────────────────────────────┐
    │                              96%                             │
    └──────────────────────────────────────────────────────────────┘
    The system is going down NOW!ystem ...
    Sent SIGTERM to all processes
    Sent SIGKILL to all processes
    Requesting system reboot
                        [  815.988103] reboot: Restarting system
```

```
virsh # list --all
 Id   Name       State
─────────────────────────
 5    ubuntuVM   running
 -    debianVM   shut off

virsh # console ubuntuVM
Connected to domain 'ubuntuVM'
Escape character is ^] (Ctrl + ])
```

Mais on a réussi à y accéder grâce à SSH et au final tout fonctionne :

```
sudo nmap 192.168.122.0/24 -sP # pour trouver l'ip de la VM ubuntu
```

```
student@livecampus:/mnt/vms/ubuntu$ ssh ubu@192.168.122.107
The authenticity of host '192.168.122.107 (192.168.122.107)' can't be established.
ED25519 key fingerprint is SHA256:ikhK+oJvuzNy9utLNs3qqmDFdSUGUzHj2Kq81tQ0J80.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.122.107' (ED25519) to the list of known hosts.
ubu@192.168.122.107's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubu@ubuntu:~$
```