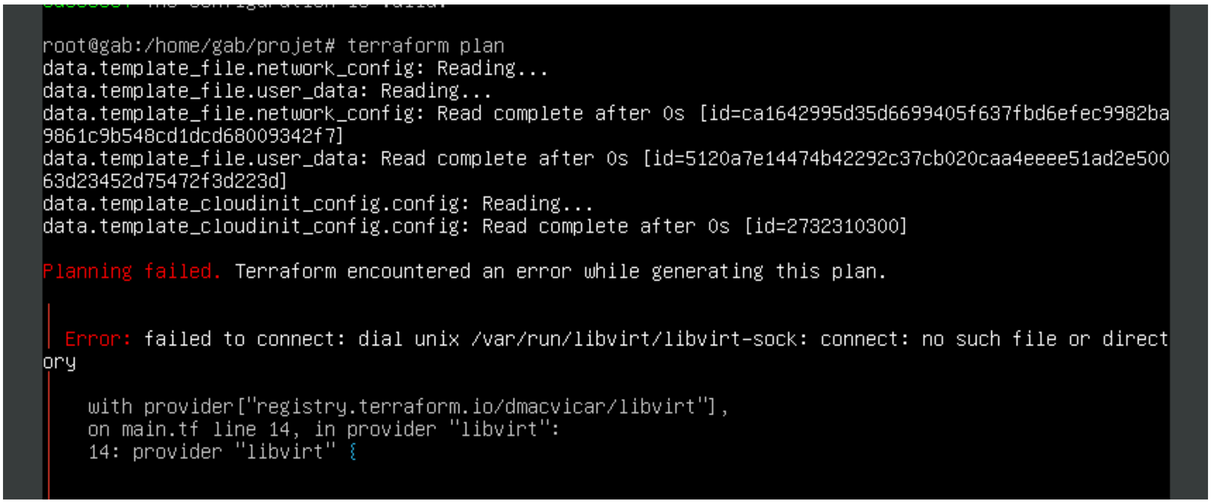
Voici un rapport des problèmes que nous avons rencontrés et de nos avancées. Il contient les commandes que nous avons utilisées pour résoudre chaque problèmes.

<u>Partie de Gabriel : tout sur le host</u>

J'ai utilisé un ubuntu-server avec virtualBox pour faire le projet.

- **<u>Première étape :</u>** recopier les fichiers main.tf, network_config_dhcp.cfg, cloud_init.cfg sur lesquels Alphonse avait déjà travaillé et les adapter à ma configuration.

ssh-keygen id25519 et la recopier dans .ssh/authorized_keys

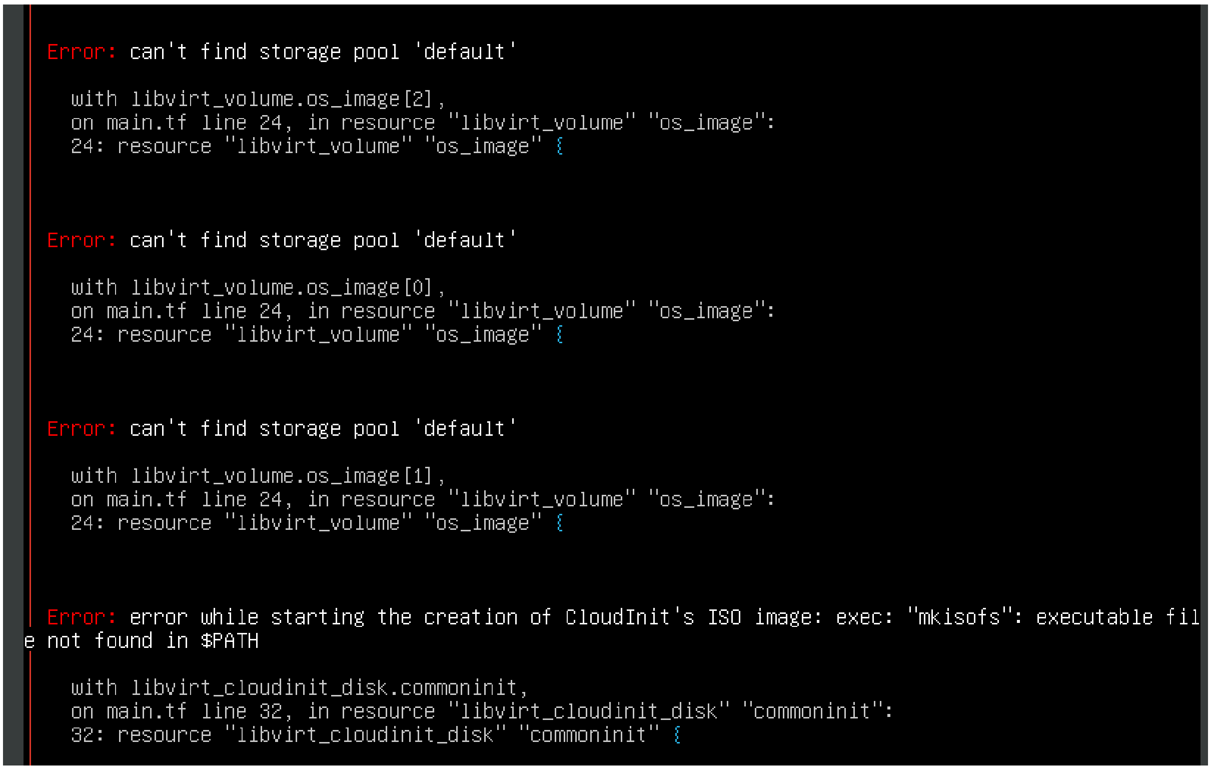


```
Success! The configuration is valid.

root@gab:/home/gab/projet# terraform plan
data.template_file.network_config: Reading...
data.template_file.user_data: Reading...
data.template_file.network_config: Read complete after 0s [id=ca1642995d35d6699405f637fbd6efec9982ba
9861c9b548cd1dcd68009342f7]
data.template_file.user_data: Read complete after 0s [id=5120a7e14474b42292c37cb020caa4eeee51ad2e500
63d23452d75472f3d223d]
data.template_cloudinit_config.config: Reading...
data.template_cloudinit_config.config: Read complete after 0s [id=2732310300]

Planning failed. Terraform encountered an error while generating this plan.

│
│ Error: failed to connect: dial unix /var/run/libvirt/libvirt-sock: connect: no such file or direct
ory
│
│   with provider["registry.terraform.io/dmacvicar/libvirt"],
│   on main.tf line 14, in provider "libvirt":
│   14: provider "libvirt" {
│
```

- *Problème : Ensure that libvirt is intalled and start the Libvirt service:*

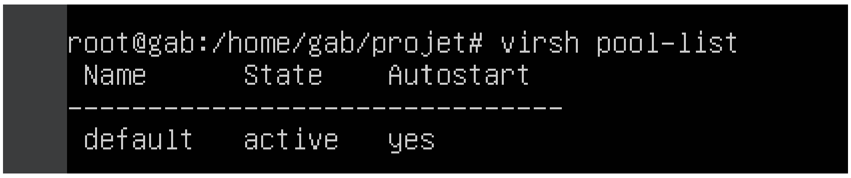
sudo apt update
sudo apt install libvirt-daemon-system libvirt-clients
sudo systemctl start libvirtd
sudo systemctl enable libvirtd
sudo usermod -aG libvirt $(whoami)   // add my user to the libvirt group

- *Problème : can't find storage pool 'default'*



```
Error: can't find storage pool 'default'

  with libvirt_volume.os_image[2],
  on main.tf line 24, in resource "libvirt_volume" "os_image":
  24: resource "libvirt_volume" "os_image" {


Error: can't find storage pool 'default'

  with libvirt_volume.os_image[0],
  on main.tf line 24, in resource "libvirt_volume" "os_image":
  24: resource "libvirt_volume" "os_image" {


Error: can't find storage pool 'default'

  with libvirt_volume.os_image[1],
  on main.tf line 24, in resource "libvirt_volume" "os_image":
  24: resource "libvirt_volume" "os_image" {


Error: error while starting the creation of CloudInit's ISO image: exec: "mkisofs": executable fil
e not found in $PATH

  with libvirt_cloudinit_disk.commoninit,
  on main.tf line 32, in resource "libvirt_cloudinit_disk" "commoninit":
  32: resource "libvirt_cloudinit_disk" "commoninit" {
```

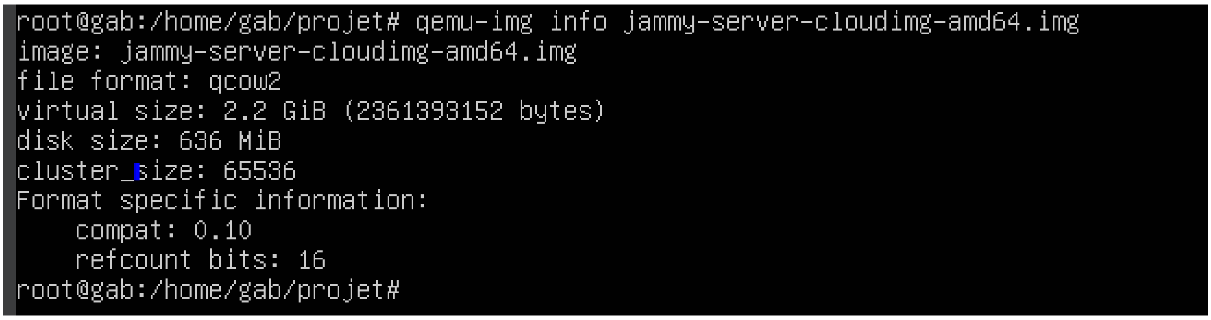
```
virsh pool-list --all
virsh pool-define-as --name default --type dir --target /var/lib/libvirt/images
virsh pool-build default
virsh pool-start default
virsh pool-autostart default
virsh pool-list                                   // OK storage pool default is available and active
```



```
root@gab:/home/gab/projet# virsh pool-list
 Name      State    Autostart
-------------------------------
 default   active   yes
```

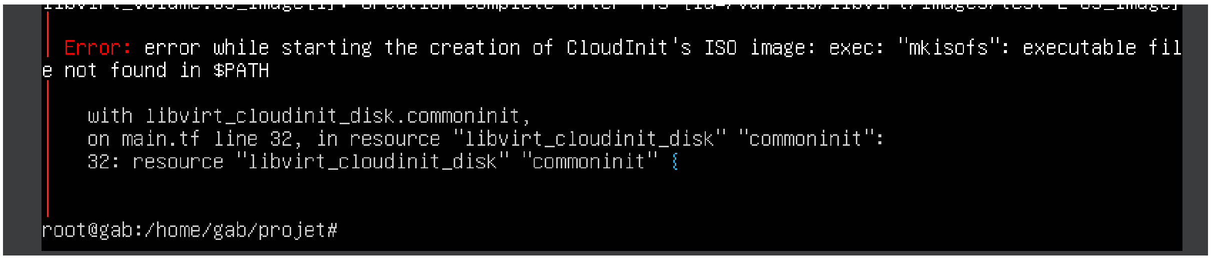
- *Problème : error while deterlining image type for jammy-server_cloudimg-amd64.img: error while openin jammy-server-cloudimg-amd64.img: no such file in directory*

```
wget https://cloud-images.ubuntu.com/jammy/current/jammy-server-cloudimg-amd64.img
chmod 644 jammy-server-cloudimg-amd64.img
sudo apt install qemu-utils
qemu-img info jammy-server-cloudimg-amd64.img
```



```
root@gab:/home/gab/projet# qemu-img info jammy-server-cloudimg-amd64.img
image: jammy-server-cloudimg-amd64.img
file format: qcow2
virtual size: 2.2 GiB (2361393152 bytes)
disk size: 636 MiB
cluster_size: 65536
Format specific information:
    compat: 0.10
    refcount bits: 16
root@gab:/home/gab/projet#
```

- *Problème : error while starting the creation of cloudInit's ISO image: exec: "mkisofs": executable file not found in $PATH*



```
│ Error: error while starting the creation of CloudInit's ISO image: exec: "mkisofs": executable fil
e not found in $PATH
│
│   with libvirt_cloudinit_disk.commoninit,
│   on main.tf line 32, in resource "libvirt_cloudinit_disk" "commoninit":
│   32: resource "libvirt_cloudinit_disk" "commoninit" {
│
root@gab:/home/gab/projet#
```

sudo apt update
sudo apt install genisoimage
which mkisofs

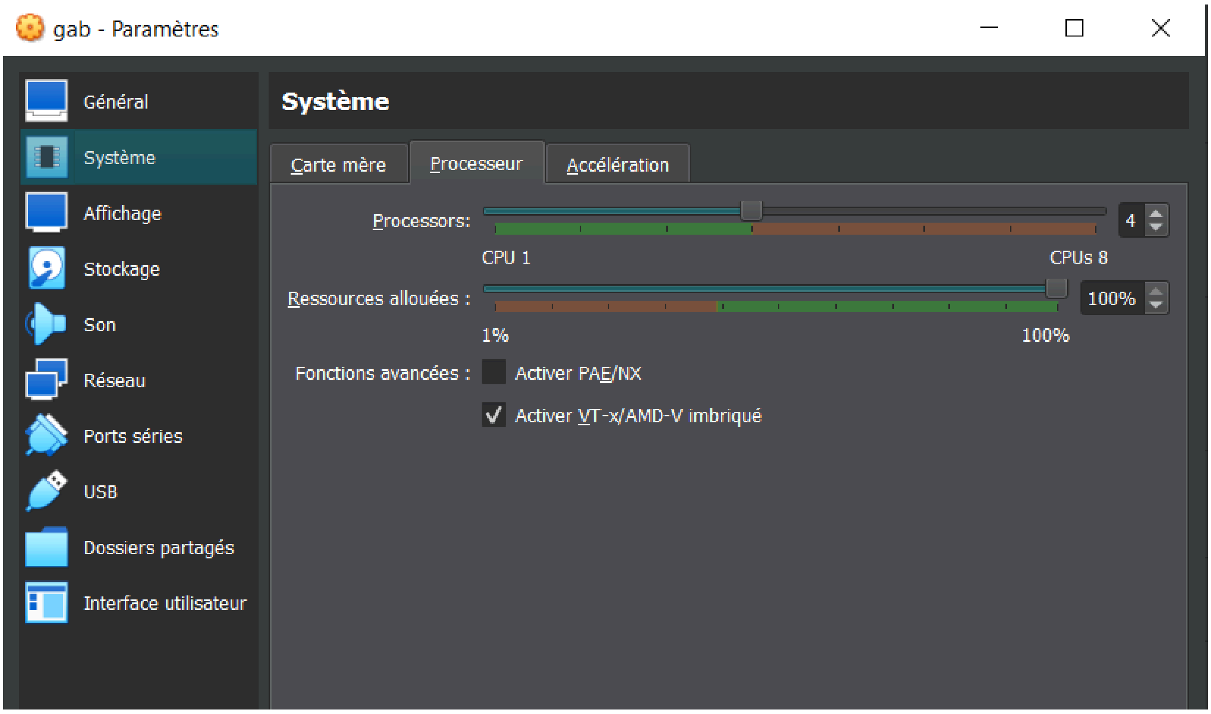
- *Problème : error defining libvirt domain : procedure interrupted while awaiting response line 62 : resource "libvirt_domain" "domain-ubuntu" {*

sudo journalctl -xeu libvirtd
virsh -c qemu:///system list --all
virsh define /path/to/test-domain.xml
virsh pool-list --all
virsh pool-info default
virsh pool-start default
df -h /var/lib/libvirt/images
virsh net-list --all
virsh net-define /etc/libvirt/qemu/networks/default.xml
virsh net-autostart default
virsh net-start default
sudo systemctl restart libvirtd

- *Problème : libvirtd[814]: unable to open /dev/kvm : no such file or directory*

egrep -c '(vmx|svm)' /proc/cpuinfo
sudo apt update
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager
sudo modprobe kvm_intel

- *Problème : activer VT-x/AMD-V imbriqué dans virtualBox*



Processors: 4

Ressources allouées : 100%

Fonctions avancées : ☐ Activer PAE/NX

☑ Activer VT-x/AMD-V imbriqué

dans invite de commande Windows en mode admin :

```
bcdedit /set hypervisorlaunchtype off
reboot
cd 'C:\Program Files\Oracle\VirtualBox\'
.\VBoxManage.exe modifyvm "gab" --nested-hw-virt on
```

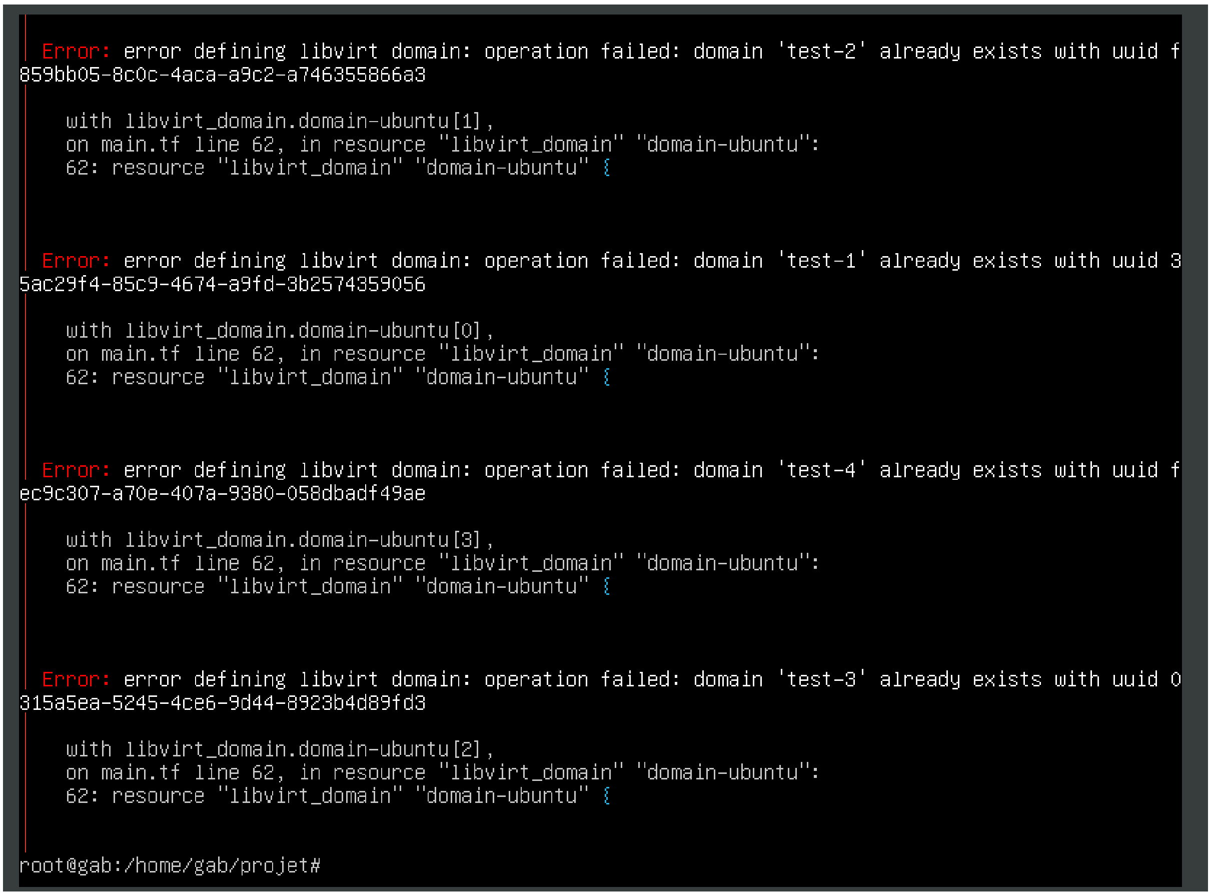
- *Problème : unit libvirtd.service could not be found*

```
sudo apt update
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager
systemctl list-units --type=service | grep libvirt
sudo systemctl start libvirtd
sudo systemctl enable libvirtd
systemctl list-units --type=service | grep libvirt
        libvirtd.service              loaded active running Virtualization daemon
        libvirt-guests.service        loaded active running Suspend/Resume guests
```

- *Problème : could not open /var/lib/libvirt/images/test--os_image' : permission denied*

```
journalctl -xeu libvirtd
ls -ld /var/lib/libvirt/images
sudo chown -R libvirt-qemu:kvm /var/lib/libvirt/images
sudo chmod 755 /var/lib/libvirt/images
sudo chown libvirt-qemu:kvm /var/lib/libvirt/images/*
sestatus
sudo setenforce 0
```

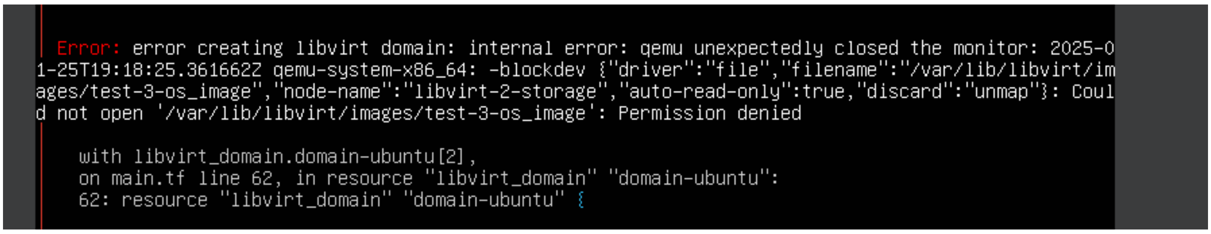
- *Problème : error defining libvirt domain : ‘test-*’ already exists*



```
│ Error: error defining libvirt domain: operation failed: domain 'test-2' already exists with uuid f
859bb05-8c0c-4aca-a9c2-a746355866a3
│
│   with libvirt_domain.domain-ubuntu[1],
│   on main.tf line 62, in resource "libvirt_domain" "domain-ubuntu":
│   62: resource "libvirt_domain" "domain-ubuntu" {
│
│

│ Error: error defining libvirt domain: operation failed: domain 'test-1' already exists with uuid 3
5ac29f4-85c9-4674-a9fd-3b2574359056
│
│   with libvirt_domain.domain-ubuntu[0],
│   on main.tf line 62, in resource "libvirt_domain" "domain-ubuntu":
│   62: resource "libvirt_domain" "domain-ubuntu" {
│
│

│ Error: error defining libvirt domain: operation failed: domain 'test-4' already exists with uuid f
ec9c307-a70e-407a-9380-058dbadf49ae
│
│   with libvirt_domain.domain-ubuntu[3],
│   on main.tf line 62, in resource "libvirt_domain" "domain-ubuntu":
│   62: resource "libvirt_domain" "domain-ubuntu" {
│
│

│ Error: error defining libvirt domain: operation failed: domain 'test-3' already exists with uuid 0
315a5ea-5245-4ce6-9d44-8923b4d89fd3
│
│   with libvirt_domain.domain-ubuntu[2],
│   on main.tf line 62, in resource "libvirt_domain" "domain-ubuntu":
│   62: resource "libvirt_domain" "domain-ubuntu" {
│
│
root@gab:/home/gab/projet#
```

virsh list --all
virsh undefine test-1
virsh undefine test-2
virsh undefine test-3
virsh undefine test-4
virsh vol-delete --pool default test-1
virsh vol-delete --pool default test-2
virsh vol-delete --pool default test-3
virsh vol-delete --pool default test-4
rm /var/lib/libvirt/images/*
terraform init
terraform plan
terraform apply

- *Problème : permission denied*



```
│ Error: error creating libvirt domain: internal error: qemu unexpectedly closed the monitor: 2025-0
1-25T19:18:25.361662Z qemu-system-x86_64: -blockdev {"driver":"file","filename":"/var/lib/libvirt/im
ages/test-3-os_image","node-name":"libvirt-2-storage","auto-read-only":true,"discard":"unmap"}: Coul
d not open '/var/lib/libvirt/images/test-3-os_image': Permission denied
│
│   with libvirt_domain.domain-ubuntu[2],
│   on main.tf line 62, in resource "libvirt_domain" "domain-ubuntu":
│   62: resource "libvirt_domain" "domain-ubuntu" {
```

ps aux | grep qemu
sudo chown -R libvirt-qemu:kvm /var/lib/libvirt/images
sudo usermod -aG libvirt $(whoami)

```
sudo usermod -aG kvm $(whoami)
newgrp libvirt
sudo aa-status
sudo systemctl stop apparmor
sudo setenforce 0
```

Modifications dans main.tf :

```
resource "null_resource" "fix_permissions" {
  provisioner "local-exec" {
    command = "sudo chown -R libvirt-qemu:kvm /var/lib/libvirt/images/* && sudo chmod -R 770 /var/lib/libvirt/images/*"
  }
  triggers = {
    always_run = timestamp()
  }
}


resource "libvirt_domain" "domain-ubuntu" {
  count = 4
  depends_on = [null_resource.fix_permissions]
  name   = "${var.hostname}-${count.index + 1}"
  memory = var.memoryMB
  vcpu   = var.cpu
…
…
  provisioner "local-exec" {
    command = "sudo chown -R libvirt-qemu:kvm /var/lib/libvirt/images/* && sudo chmod -R 770 /var/lib/libvirt/images/*"
  }
}
resource "libvirt_volume" "os_image" {
  count   = 4
  name    = "${var.hostname}-${count.index + 1}-os_image"
  pool    = "default"
  source  = "jammy-server-cloudimg-amd64.img"
  format  = "qcow2"

  provisioner "local-exec" {
    command = "sudo chown libvirt-qemu:kvm /var/lib/libvirt/images/${var.hostname}-${count.index + 1}-os_image && sudo chmod 770 /var/lib/libvirt/images/${var.hostname}-${count.index + 1}-os_image"
  }
}
```

```
virsh pool-dumpxml default                                        // vérifications
```

le pool default donne :

```
        <permissions>
         <mode>0770</mode>
         <owner>64055</owner>                              // libvirt-qemu
         <group>108</group>                                // kvm
        </permissions>
getent passwd 64055
getent group 108
```

Toutes ces modifications ont permis de bien modifier les permissions des fichiers dans /var/lib/libvirt/images mais j'ai toujours l'erreur *Permission denied* . Il faut donc chercher à qui donner les permissions.

Après de nombreuses recherches,j'ai vu qu'il fallait modifier le ficher /etc/libvirt/qemu.conf et mettre dedans :

```
user = "libvirt-qemu"
group = "kvm"
```

pour aligner la configuration avec les permissions que j'ai données avec les provisioners. Ça ne marchait toujours pas donc j'ai testé en changeant tout en mode root : les provisioners et user et group dans qemu.conf et CA MARCHE, plus d'erreurs après terraform apply !!!!

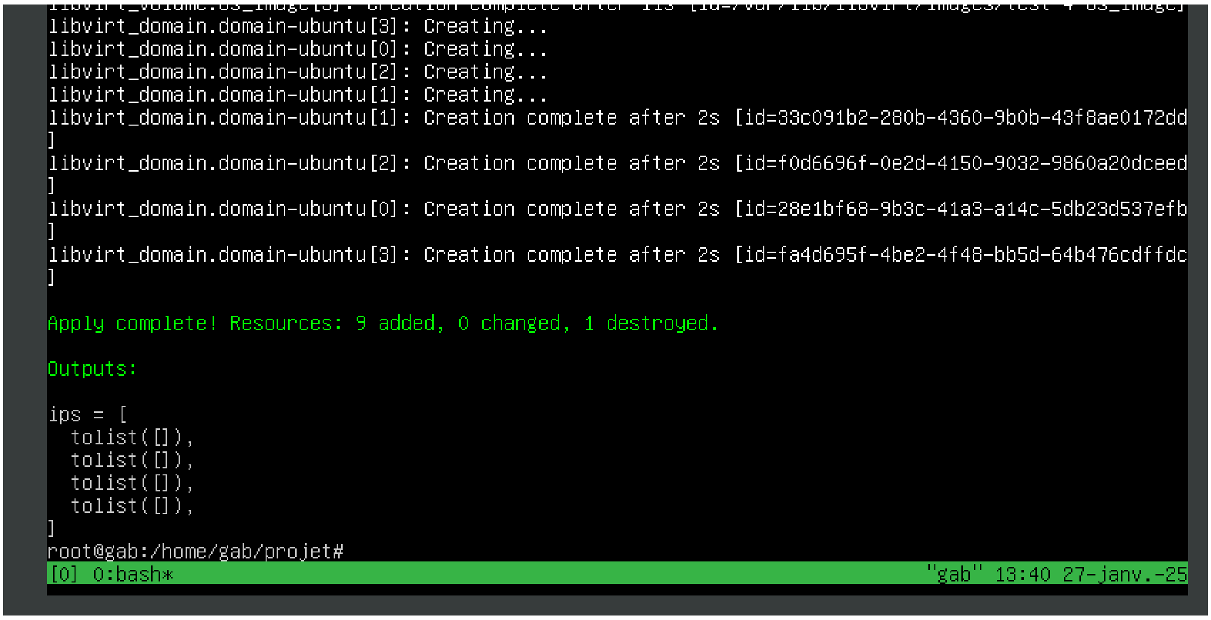
- *Problème : réactiver AppArmor mais changer une règle*

```
sudo nano /etc/apparmor.d/usr.sbin.libvirtd
```

puis écrire : `/var/lib/libvirt/images/** rwk,` dans le fichier

```
sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.libvirtd
```

- *Problème : terraform apply lance bien 4 VMs mais elles n'ont pas d'adresse IP*

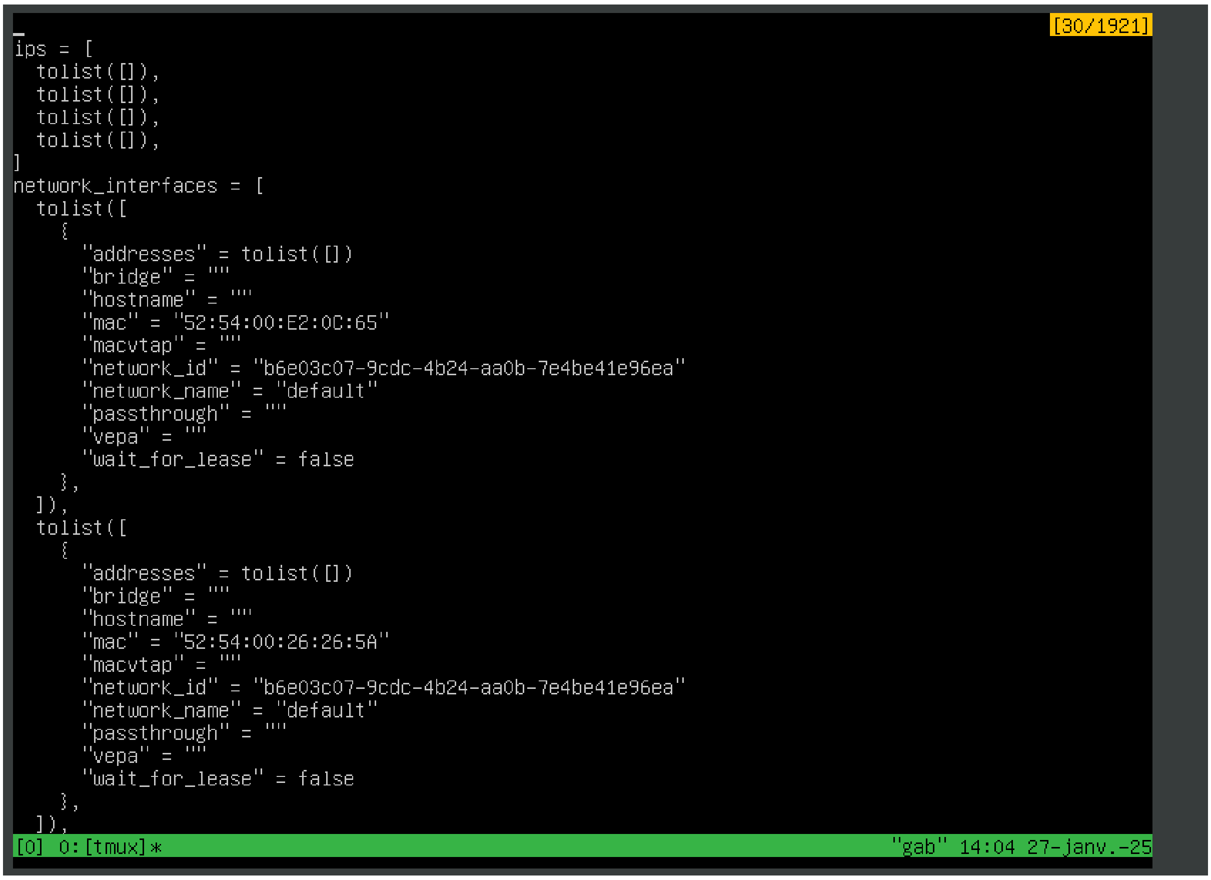


```
libvirt_volume.os_image[3]: Creation complete after 11s [id=/var/lib/libvirt/images/test-4-os_image]
libvirt_domain.domain-ubuntu[3]: Creating...
libvirt_domain.domain-ubuntu[0]: Creating...
libvirt_domain.domain-ubuntu[2]: Creating...
libvirt_domain.domain-ubuntu[1]: Creating...
libvirt_domain.domain-ubuntu[1]: Creation complete after 2s [id=33c091b2-280b-4360-9b0b-43f8ae0172dd
]
libvirt_domain.domain-ubuntu[2]: Creation complete after 2s [id=f0d6696f-0e2d-4150-9032-9860a20dceed
]
libvirt_domain.domain-ubuntu[0]: Creation complete after 2s [id=28e1bf68-9b3c-41a3-a14c-5db23d537efb
]
libvirt_domain.domain-ubuntu[3]: Creation complete after 2s [id=fa4d695f-4be2-4f48-bb5d-64b476cdffdc
]

Apply complete! Resources: 9 added, 0 changed, 1 destroyed.

Outputs:

ips = [
  tolist([]),
  tolist([]),
  tolist([]),
  tolist([]),
]
root@gab:/home/gab/projet#
[0] 0:bash*                                                              "gab" 13:40 27-janv.-25
```

```
virsh net-list --all
         Name      State    Autostart    Persistent
        --------------------------------------------
         default   active   yes          yes
virsh net-dumpxml default
        <ip address='192.168.122.1' netmask='255.255.255.0'>
          <dhcp>
            <range start='192.168.122.2' end='192.168.122.254'/>
          </dhcp>
        </ip>
```

Ajout pour le debug dans main.tf :

```
output "network_interfaces" {
  value = libvirt_domain.domain-ubuntu[*].network_interface
}
```



```
                                                                                                   [30/1921]
ips = [
  tolist([]),
  tolist([]),
  tolist([]),
  tolist([]),
]
network_interfaces = [
  tolist([
    {
      "addresses" = tolist([])
      "bridge" = ""
      "hostname" = ""
      "mac" = "52:54:00:E2:0C:65"
      "macvtap" = ""
      "network_id" = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea"
      "network_name" = "default"
      "passthrough" = ""
      "vepa" = ""
      "wait_for_lease" = false
    },
  ]),
  tolist([
    {
      "addresses" = tolist([])
      "bridge" = ""
      "hostname" = ""
      "mac" = "52:54:00:26:26:5A"
      "macvtap" = ""
      "network_id" = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea"
      "network_name" = "default"
      "passthrough" = ""
      "vepa" = ""
      "wait_for_lease" = false
    },
  ]),
[0] 0:[tmux]*                                                                      "gab" 14:04 27-janv.-25
```

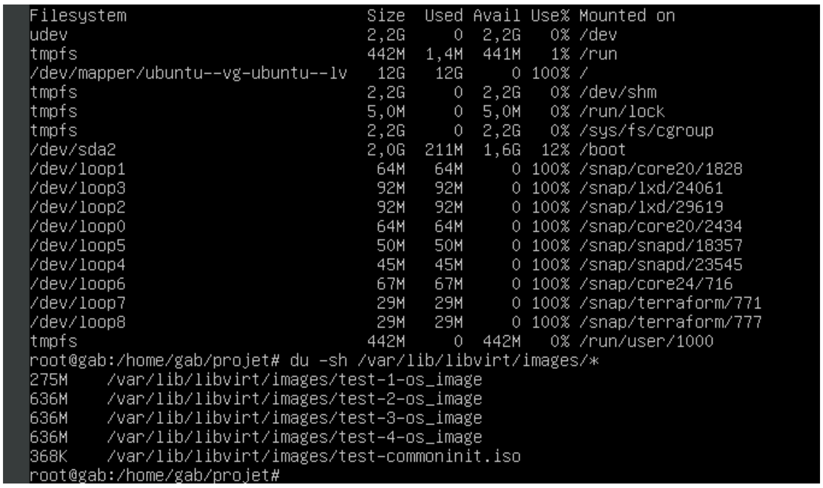
- *Problème : SSD full sur ma machine physique*

“The I/O cache encountered an error while updating data in medium &quot;ahci-0-0&quot; (rc=VERR_DISK_FULL). Make sure there is enough free space on the disk and that the disk is working properly. Operation can be resumed afterwards. Error ID:BLKCACHE_IOERR Severity:Erreur non Fatale”

J’ai libéré de l’espace sur mon SSD et la VM s’est remis en marche

- *Problème : no space left on device*

Plus de place sur ma VM.
df -h



```
Filesystem                          Size  Used Avail Use% Mounted on
udev                                2,2G     0  2,2G   0% /dev
tmpfs                               442M  1,4M  441M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv    12G   12G     0 100% /
tmpfs                               2,2G     0  2,2G   0% /dev/shm
tmpfs                               5,0M     0  5,0M   0% /run/lock
tmpfs                               2,2G     0  2,2G   0% /sys/fs/cgroup
/dev/sda2                           2,0G  211M  1,6G  12% /boot
/dev/loop1                           64M   64M     0 100% /snap/core20/1828
/dev/loop3                           92M   92M     0 100% /snap/lxd/24061
/dev/loop2                           92M   92M     0 100% /snap/lxd/29619
/dev/loop0                           64M   64M     0 100% /snap/core20/2434
/dev/loop5                           50M   50M     0 100% /snap/snapd/18357
/dev/loop4                           45M   45M     0 100% /snap/snapd/23545
/dev/loop6                           67M   67M     0 100% /snap/core24/716
/dev/loop7                           29M   29M     0 100% /snap/terraform/771
/dev/loop8                           29M   29M     0 100% /snap/terraform/777
tmpfs                               442M     0  442M   0% /run/user/1000
root@gab:/home/gab/projet# du -sh /var/lib/libvirt/images/*
275M    /var/lib/libvirt/images/test-1-os_image
636M    /var/lib/libvirt/images/test-2-os_image
636M    /var/lib/libvirt/images/test-3-os_image
636M    /var/lib/libvirt/images/test-4-os_image
368K    /var/lib/libvirt/images/test-commoninit.iso
root@gab:/home/gab/projet#
```

```
vgs
lvextend -L+10G /dev/mapper/ubuntu--vg-ubuntu--lv
resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

- *Problème final : Toujours pas d'adresse IP sur les VMs*

```
virsh net-dhcp-leases default                                   // marche pas
```

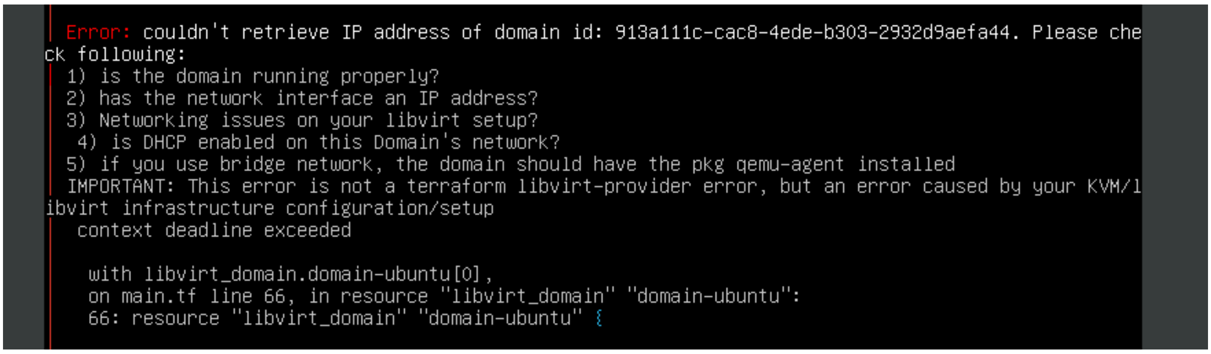
Créer un réseau bridge :

```
ip a | grep br
```

```
nano bridge-network.xml
        <network>
           <name>virbr0-network</name>
             <forward mode="bridge"/>
           <bridge name="virbr0"/>
        </network>
virsh net-define bridge-network.xml
virsh net-start br0-network
virsh net-autostart br0-network
```

Modifier dans main.tf :

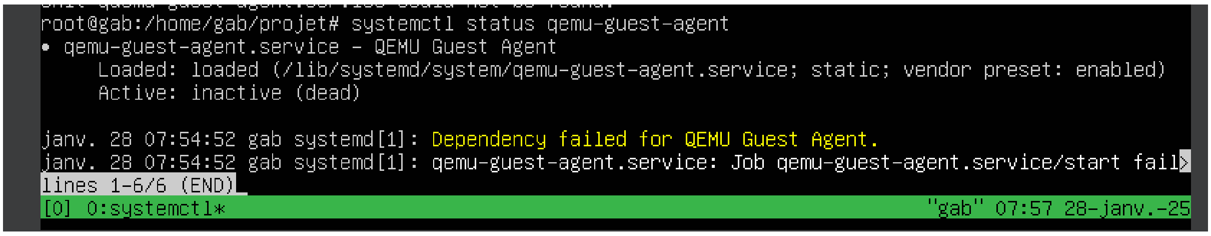
```
network_interface {
  bridge = "virbr0"
```



```
 Error: couldn't retrieve IP address of domain id: 913a111c-cac8-4ede-b303-2932d9aefa44. Please che
ck following:
 1) is the domain running properly?
 2) has the network interface an IP address?
 3) Networking issues on your libvirt setup?
  4) is DHCP enabled on this Domain's network?
 5) if you use bridge network, the domain should have the pkg qemu-agent installed
 IMPORTANT: This error is not a terraform libvirt-provider error, but an error caused by your KVM/l
ibvirt infrastructure configuration/setup
  context deadline exceeded

   with libvirt_domain.domain-ubuntu[0],
   on main.tf line 66, in resource "libvirt_domain" "domain-ubuntu":
   66: resource "libvirt_domain" "domain-ubuntu" {
```

```
dpkg -l | grep qemu-guest-agent
systemctl status qemu-guest-agent
```

a dependency job for qemu-guest-agent.service failed



```
root@gab:/home/gab/projet# systemctl status qemu-guest-agent
● qemu-guest-agent.service - QEMU Guest Agent
     Loaded: loaded (/lib/systemd/system/qemu-guest-agent.service; static; vendor preset: enabled)
     Active: inactive (dead)

janv. 28 07:54:52 gab systemd[1]: Dependency failed for QEMU Guest Agent.
janv. 28 07:54:52 gab systemd[1]: qemu-guest-agent.service: Job qemu-guest-agent.service/start fail>
lines 1-6/6 (END)
[0] 0:systemctl*                                                              "gab" 07:57 28-janv.-25
```

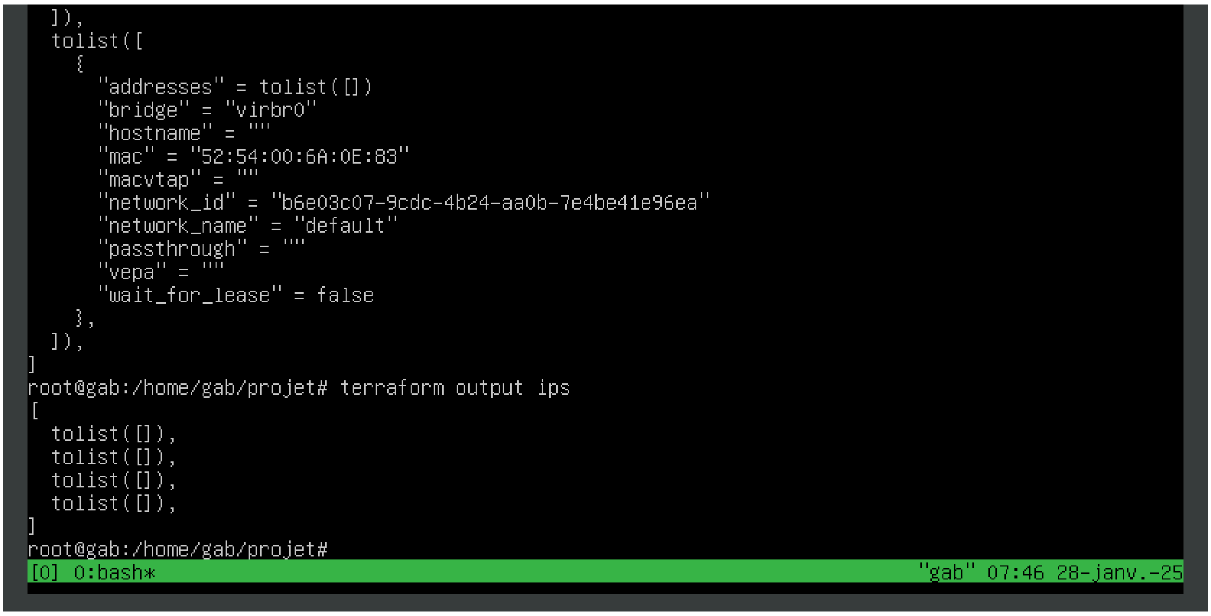
```
systemctl list-dependencies --failed
sudo apt --fix-broken install
sudo systemctl daemon-reexec
sudo systemctl restart qemu-guest-agent
sudo apt reinstall libvirt-daemon libvirt-daemon-system
sudo systemctl restart libvirtd
sudo systemctl restart qemu-guest-agent
```

Dans main.tf, ajouter à resource “libvirt_domain” :

```
qemu_agent = true
```

```
sudo systemctl enable --now qemu-guest-agent
systemctl list-dependencies --failed                          // AppArmor
sudo aa-complain /etc/apparmor.d/usr.sbin.libvirtd
sudo aa-complain /etc/apparmor.d/usr.sbin.qemu-system-x86_64
sudo systemctl restart libvirtd qemu-guest-agent
sudo nano /etc/apparmor.d/local/usr.sbin.libvirtd
        écrire /usr/sbin/qemu-system-x86_64 rix,
sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.libvirtd
sudo systemctl restart qemu-guest-agent
```

Résultat : Il y a un pont virbr0 bien configuré mais toujours pas d’adresses IP

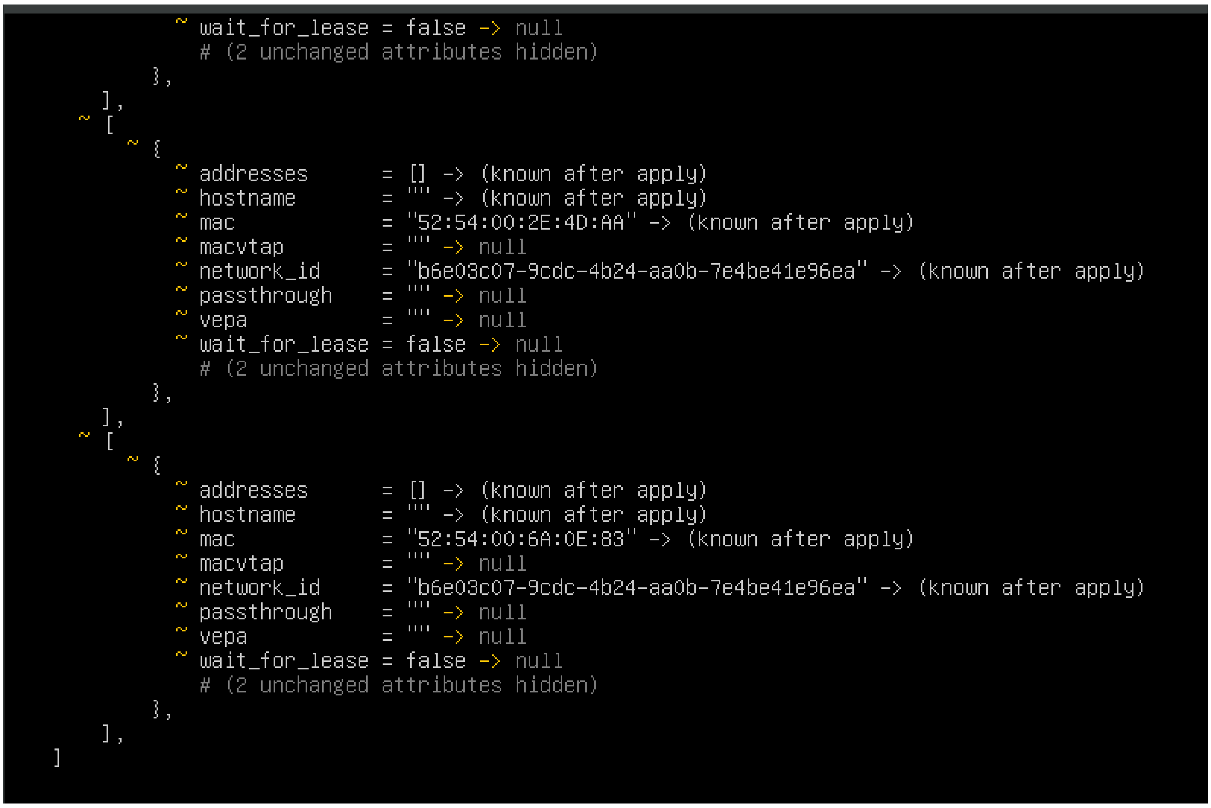


```
  ]),
  tolist([
    {
      "addresses" = tolist([])
      "bridge" = "virbr0"
      "hostname" = ""
      "mac" = "52:54:00:6A:0E:83"
      "macvtap" = ""
      "network_id" = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea"
      "network_name" = "default"
      "passthrough" = ""
      "vepa" = ""
      "wait_for_lease" = false
    },
  ]),
]
root@gab:/home/gab/projet# terraform output ips
[
  tolist([]),
  tolist([]),
  tolist([]),
  tolist([]),
]
root@gab:/home/gab/projet#
[0] 0:bash*                                                          "gab" 07:46 28-janv.-25
```

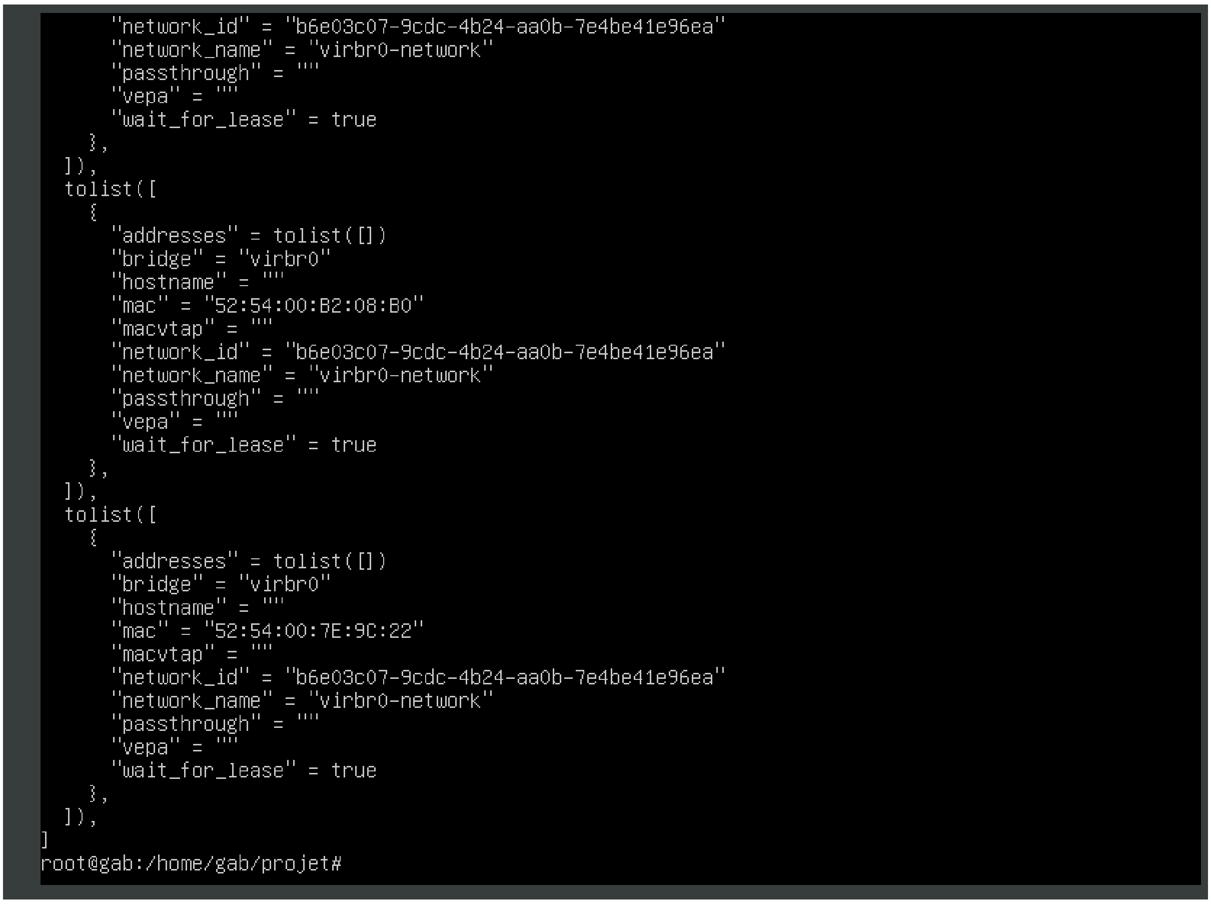
Résultat final :

```
terraform plan
```

Il y a bien la mention known fter apply pour les adress IP et autres paramètres réseau.



```
              ~ wait_for_lease = false -> null
                # (2 unchanged attributes hidden)
            },
        ],
      ~ [
          ~ {
              ~ addresses      = [] -> (known after apply)
              ~ hostname       = "" -> (known after apply)
              ~ mac            = "52:54:00:2E:4D:AA" -> (known after apply)
              ~ macvtap        = "" -> null
              ~ network_id     = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea" -> (known after apply)
              ~ passthrough    = "" -> null
              ~ vepa           = "" -> null
              ~ wait_for_lease = false -> null
                # (2 unchanged attributes hidden)
            },
        ],
      ~ [
          ~ {
              ~ addresses      = [] -> (known after apply)
              ~ hostname       = "" -> (known after apply)
              ~ mac            = "52:54:00:6A:0E:83" -> (known after apply)
              ~ macvtap        = "" -> null
              ~ network_id     = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea" -> (known after apply)
              ~ passthrough    = "" -> null
              ~ vepa           = "" -> null
              ~ wait_for_lease = false -> null
                # (2 unchanged attributes hidden)
            },
        ],
    ]
```

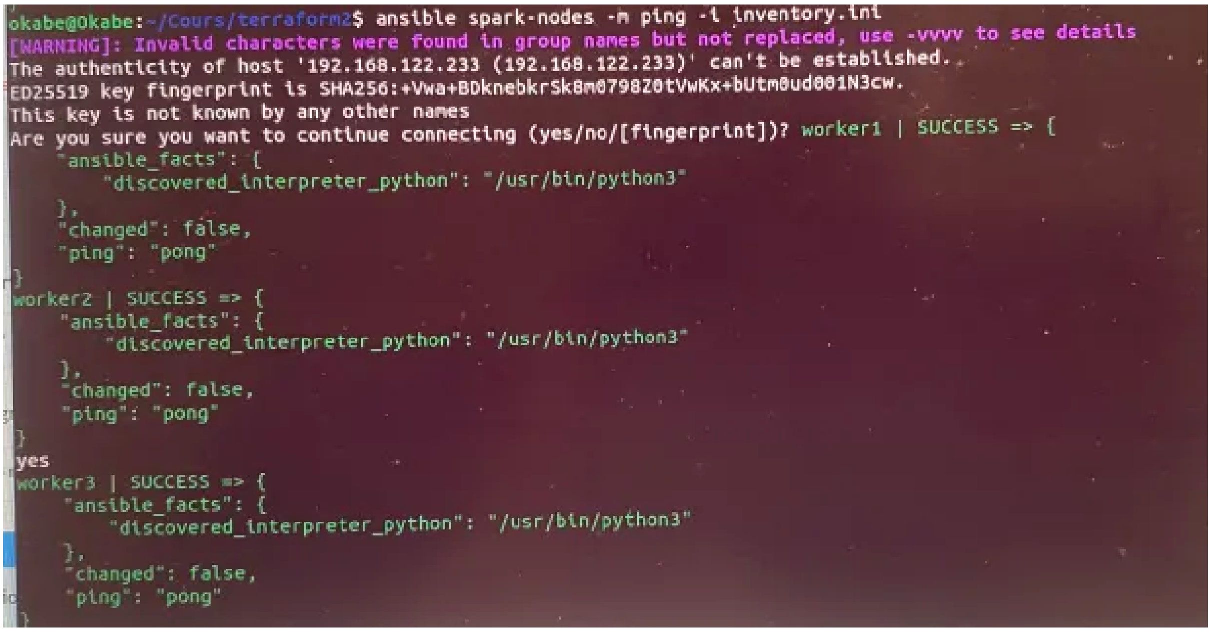


```
      "network_id" = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea"
      "network_name" = "virbr0-network"
      "passthrough" = ""
      "vepa" = ""
      "wait_for_lease" = true
    },
  ]),
  tolist([
    {
      "addresses" = tolist([])
      "bridge" = "virbr0"
      "hostname" = ""
      "mac" = "52:54:00:B2:08:B0"
      "macvtap" = ""
      "network_id" = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea"
      "network_name" = "virbr0-network"
      "passthrough" = ""
      "vepa" = ""
      "wait_for_lease" = true
    },
  ]),
  tolist([
    {
      "addresses" = tolist([])
      "bridge" = "virbr0"
      "hostname" = ""
      "mac" = "52:54:00:7E:9C:22"
      "macvtap" = ""
      "network_id" = "b6e03c07-9cdc-4b24-aa0b-7e4be41e96ea"
      "network_name" = "virbr0-network"
      "passthrough" = ""
      "vepa" = ""
      "wait_for_lease" = true
    },
  ]),
]
root@gab:/home/gab/projet#
```

<u>Partie de Alphonse:</u>

Pour ma part j’avais un dual boot donc je me suis évité une partie des problèmes qu’à pu rencontrer Gabriel, et j’ai pu écrire et modifier les fichiers main.tf et les fichiers .cfg en m’aidant de la doc terraform et kvm.

Après avoir résolu le problème des adresses ip données aux vms j’ai défini 1 master node et des worker nodes et ai lancé un playbook ansible pour vois si le master pouvait ping les 3 workers :



```
okabe@Okabe:~/Cours/terraform2$ ansible spark-nodes -m ping -i inventory.ini
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
The authenticity of host '192.168.122.233 (192.168.122.233)' can't be established.
ED25519 key fingerprint is SHA256:+Vwa+BDknebkrSk8m0798Z0tVwKx+bUtm0ud001N3cw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? worker1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
worker2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
worker3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

ça à marché mais lorsque j’ai ensuite lancé le playbook qui installe java puis spark. Je me suis aperçu que l’image que j’avais utilisé pour mes vms n’était pas asser grande et il n’y avait pas la place d’installer java. J’ai donc modifié l’image avec kvm mais suite a cette manip, je n’ai jamais réussi à réattribuer des ips cohérentes à mes vms et donc à faire en sorte qu’elles puissent communiquer entre elles. Je n’ai donc pas pu configurer correctement spark, même une fois installé sur une des vms.

Je n’ai pas compris comment résoudre à 100% le problème des ips, en relançant le réseau en faisant :

virsh net-destroy default
virsh net-start default

le problème ne se réglait pas, en vidant la config dhcp du réseau avant de relancer le réseau:

virsh net-edit default

le problème ne se réglait pas non plus. J’ai aussi essayé de configurer des ips statiques mais comme il fallait relancer terraform pour les attribuer aux vms a leur création, je ne pouvais pas (les @ mac des vms changent). J’ai aussi essayer de juste relancer dhcp et pas le réseau en entier, de restart qemu-agent sur les vms, faire du NAT au lieu du bridge …

Je n’ai au final jamais réussi à résoudre ce problème.