1. Proxmox : utiliser une carte graphique Nvidia avec un conteneur LXC

Lien : https://shionn.github.io/draft/nvidia-proxmox-lxc-passthrought-ollama.html

1.1. Installation des pilotes sur le serveur Proxmox

- mise à jour du serveur

```
apt update && apt upgrade
```

- Installation des prérequis logiciels

```
apt install pve-nvidia-vgpu-helper nvtop pve-headers build-essential
```

- Pré configuration de Proxmox :

```
pve-nvidia-vgpu-helper setup
```

- Installation des paquets du driver nvidia.

```
wget https://developer.download.nvidia.com/compute/cuda/repos/debian13/x86_64/cuda-keyring_1.1-1_all.deb
apt install ./cuda-keyring_1.1-1_all.deb
apt update
apt upgrade
apt install nvidia-driver-cuda
```

- reboot du serveur
- il doit maintenant être possible d'utiliser l'outil nvidia-smi :

```
# nvidia-smi
Wed Jan 14 15:05:04 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 590.48.01              Driver Version: 590.48.01      CUDA Version: 13.1     |
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       On  |   00000000:86:00.0 Off |                    0 |
| N/A   38C    P8              8W /   70W |       0MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   1  Tesla T4                       On  |   00000000:AF:00.0 Off |                    0 |
| N/A   38C    P8              9W /   70W |       0MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

- Visualisation des périphériques Nvidia ajouté à l'hôte Proxmox :

```
# ls -l /dev/nvi*
crw-rw-rw- 1 root root 195,   0 Jan 12 23:18 /dev/nvidia0
crw-rw-rw- 1 root root 195,   1 Jan 12 23:18 /dev/nvidia1
crw-rw-rw- 1 root root 195, 255 Jan 12 23:18 /dev/nvidiactl
crw-rw-rw- 1 root root 195, 254 Jan 12 23:18 /dev/nvidia-modeset
crw-rw-rw- 1 root root 511,   0 Jan 14 11:56 /dev/nvidia-uvm
crw-rw-rw- 1 root root 511,   1 Jan 14 11:56 /dev/nvidia-uvm-tools

/dev/nvidia-caps:
```

```
total 0
cr-------- 1 root root 236, 1 Jan 14 11:56 nvidia-cap1
cr--r--r-- 1 root root 236, 2 Jan 14 11:56 nvidia-cap2
```

- Vérifier si Proxmox voit bien les deux GPU au niveau PCIe

```
# lspci | grep -i nvidia

AF:00.0 NVIDIA Corporation TU104GL [Tesla T4]
B0:00.0 NVIDIA Corporation TU104GL [Tesla T4]
```

- Vérifier si le driver charge bien les deux GPU : il ne doit y avoir aucune erreur

```
# dmesg | grep -i nvidia
Erreurs possibles :
GPU has fallen off the bus
PCIe error
failed to initialize gpu
RUNTIME_PM: error
Unknown chipset
NVRM: RmInitAdapter failed
```

- Vérifier si le module UVM détecte les deux GPU

```
# cat /proc/driver/nvidia/gpus/*/information
Il doit y avoir deux répertoires (0 et 1) :
# nvidia-smi -q | grep -i "Compute Mode"
    Compute Mode                                    : Default
root@siohyp2:~# cat /proc/driver/nvidia/gpus/*/information
Model:           Tesla T4
IRQ:             44
GPU UUID:        GPU-e5bc6842-5aa8-b29e-aa13-922b15c893f9
Video BIOS:      90.04.b4.00.04
Bus Type:        PCIe
DMA Size:        47 bits
DMA Mask:        0x7fffffffffff
Bus Location:    0000:86:00.0
Device Minor:    0
GPU Firmware:    590.48.01
GPU Excluded:    No
Model:           Tesla T4
IRQ:             46
GPU UUID:        GPU-6ac33a99-2cb8-eb7d-6097-f1c29e4d1e51
Video BIOS:      90.04.b4.00.04
Bus Type:        PCIe
DMA Size:        47 bits
DMA Mask:        0x7fffffffffff
Bus Location:    0000:af:00.0
Device Minor:    1
GPU Firmware:    590.48.01
GPU Excluded:    No
```

Il y a deux cartes avec des adresses PCI différentes :

- GPU 0 → 0000:86:00.0
- GPU 1 → 0000:af:00.0

1.1.1. Problème Debian 13 + NVIDIA 590 pour les cartes Testla T4

Debian 13 (Trixie) est en version Testing, et utilise :

- un kernel 6.7 / 6.8 / 6.17 selon build Proxmox
- libdrm, libglvnd, libnvidia-ml plus récentes que Debian 12
- NVIDIA driver branche 590.xx (pas encore stabilisée multi-GPU)

Sous Debian 13 + kernel récent, on observe :

- Le kernel NVIDIA voit les deux GPU ⇒ (/proc/driver/nvidia/gpus → OK)

- NVML (bibliothèque utilisée par nvidia-smi) ne recense qu'un seul GPU
  - ⇒ nvidia-smi n'affiche qu'un GPU
  - ⇒ topo -m n'affiche qu'un GPU
  - ⇒ nvidia-smi -i 1 = No devices found

1.1.2. SOLUTION OFFICIELLE POUR PROXMOX 9 + DEBIAN 13

- Installer le driver nvidia-driver-550

La branche 550 est compatible Debian 13 et ne souffre pas du bug NVML.

```
# apt remove --purge -y nvidia-driver nvidia-dkms-* nvidia-kernel-common-* nvidia-kernel-source-*
nvidia-utils-* libnvidia-ml-dev
# apt autoremove --purge -y
# reboot
```

- Télécharger le driver NVIDIA stable pour Tesla T4 (La version 550.Explicit est actuellement la plus stable, et compatible CUDA 12/13).

```
# wget https://us.download.nvidia.com/tesla/550.127.05/NVIDIA-Linux-x86_64-550.127.05.runchmod +x
NVIDIA-Linux-x86_64-550.127.05.run
# chmod +x NVIDIA-Linux-x86_64-550.127.05.run
```

- Lancer l'installation DKMS

```
# ./NVIDIA-Linux-x86_64-550.127.05.run --dkms -m=kernel

Répondre :

“yes” pour DKMS
“no” pour nouveau (il sera blacklisté automatiquement)
```

- Puis :

```
apt install nvidia-driver-550
reboot
```

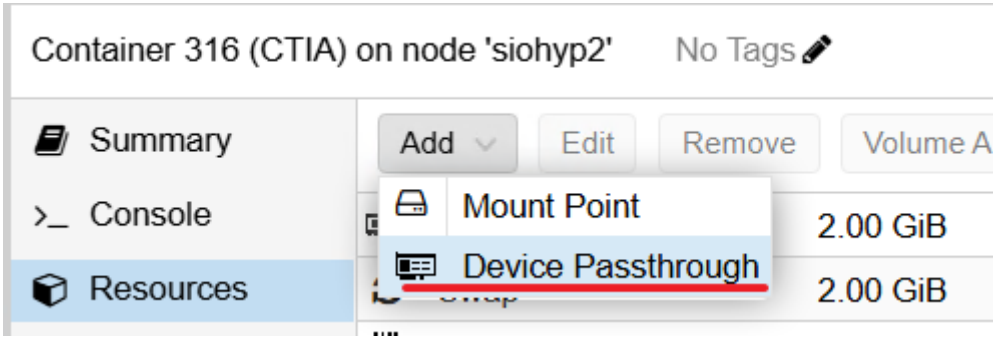
```
= driver stable, multi-GPU, compatible T4 / CUDA 12 / CUDA 13.
```

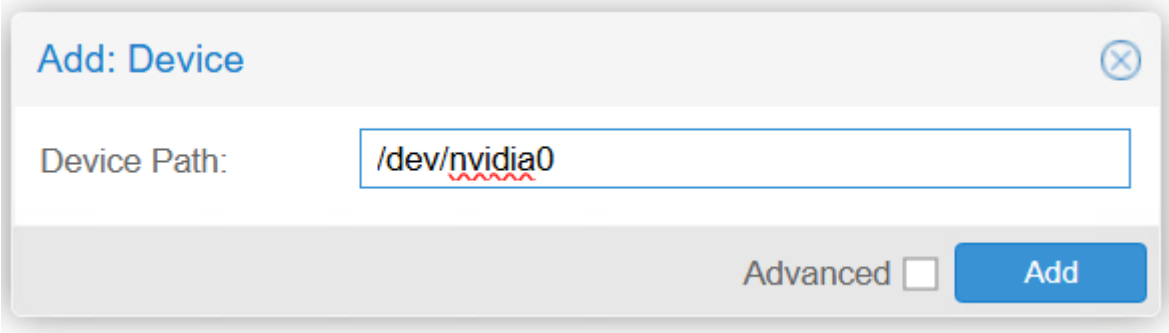
1.2. Nvidia dans le Container LXC

- mettre à jour le conteneur

```
apt update & apt upgrade
```

- Les conteneurs LXC n'ont pas besoin d'option particulière, ni besoin d'être privilégiés.
- Configurer Le passthrough (GPU passthrough) dans Proxmox pour les GPU des carte NVidia.
- ajoutez au conteneur LXC les périphériques passthrough





Device (dev0)	/dev/nvidia0
Device (dev1)	/dev/nvidiactl
Device (dev2)	/dev/nvidia-modeset
Device (dev3)	/dev/nvidia-caps/nvidia-cap1
Device (dev4)	/dev/nvidia-caps/nvidia-cap2
Device (dev5)	/dev/nvidia-uvm
Device (dev6)	/dev/nvidia-uvm-tools

- Installez les drivers nvidia et la suite logicielle **cuda** dans le conteneur LXC (procédure semblable à celle de l'hote Proxmox).

```
wget https://developer.download.nvidia.com/compute/cuda/repos/debian13/x86_64/cuda-keyring_1.1-1_all.deb
apt install ./cuda-keyring_1.1-1_all.deb
apt update
apt install cuda-toolkit
apt install nvidia-driver-cuda
```

- commande **nvdia-smi** pour confirmer que la carte est disponible et fonctionnelle sur votre container.

```
nvidia-smi
```

From:
https://siocours.lycees.nouvelle-aquitaine.pro/ - **Les cours du BTS SIO**

Permanent link:
https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/cloud/proxmox/lxcnvidia?rev=1774873511

Last update: **2026/03/30 12:25**