

Proxmox : utiliser une carte graphique Nvidia avec un conteneur LXC

Lien : <https://shionn.github.io/draft/nvidia-proxmox-lxc-passthrough-ollama.html>

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
```

- Problèmes rencontrés avec l'installation de plusieurs pilotes Nvidia pour Debian 13.
- Installation manuelle de ces pilotes avec ce script :

```
#!/bin/bash

set -e # Stoppe le script en cas d'erreur

BASE_URL="https://developer.download.nvidia.com/compute/cuda/repos/debian13/x86_64"

# Liste des paquets à télécharger
packages=(
    "firmware-nvidia-gsp_590.48.01-1_amd64.deb"
    "libnvidia-gpucomp_590.48.01-1_amd64.deb"
    "libnvidia-ptxjitcompiler1_590.48.01-1_amd64.deb"
    "libnvidia-pkcs11-openssl3_590.48.01-1_amd64.deb"
    "libcuda1_590.48.01-1_amd64.deb"
    "libcudadebbugger1_590.48.01-1_amd64.deb"
    "libnvcuvid1_590.48.01-1_amd64.deb"
    "libnvidia-cfg1_590.48.01-1_amd64.deb"
    "libnvidia-encode1_590.48.01-1_amd64.deb"
    "nvidia-modprobe_590.48.01-1_amd64.deb"
    "nvidia-kernel-support_590.48.01-1_amd64.deb"
    "libnvidia-fbc1_590.48.01-1_amd64.deb"
    "libnvidia-ml1_590.48.01-1_amd64.deb"
    "libnvidia-nvvm4_590.48.01-1_amd64.deb"
    "libnvidia-nvvm704_590.48.01-1_amd64.deb"
    "libnvidia-opticalflow1_590.48.01-1_amd64.deb"
    "libnvidia-present_590.48.01-1_amd64.deb"
    "libnvidia-sandboxutils_590.48.01-1_amd64.deb"
    "libnvidia-tileiras_590.48.01-1_amd64.deb"
    "libnvoptix1_590.48.01-1_amd64.deb"
    "nvidia-opencl-icd_590.48.01-1_amd64.deb"
    "nvidia-persistenced_590.48.01-1_amd64.deb"
    "nvidia-driver-cuda_590.48.01-1_amd64.deb"
)

echo "=== Téléchargement et installation des paquets NVIDIA CUDA ==="

for pkg in "${packages[@]"; do
```

```

echo ""
echo "--- Téléchargement : $pkg ---"
wget -q "$BASE_URL/$pkg" -O "$pkg"

echo "Installation de $pkg..."
dpkg -i "$pkg"
done

echo ""
echo "=== Tous les paquets ont été installés avec succès ! ==="

echo "Correction des paquets manquants."
apt --fix-broken install
`

```

- 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/nvidiaactl
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-uvmm
crw-rw-rw- 1 root root 511, 1 Jan 14 11:56 /dev/nvidia-uvmm-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 que CUDA voit les deux cartes

```

# nvidia-smi -L
GPU 0: Tesla T4 (UUID: GPU-e5bc6842-5aa8-b29e-aa13-922b15c893f9)

```

GPU 1: Tesla T4 (UUID: GPU-6ac33a99-2cb8-eb7d-6097-f1c29e4d1e51)

- 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
- lancer un benchmark PCIe / mémoire

```
# nvidia-smi topo -m
      GPU0      GPU1      CPU Affinity      NUMA Affinity      GPU NUMA ID
GPU0      X        NODE      24-35,72-83        2                    N/A
GPU1      NODE      X        24-35,72-83        2                    N/A
```

```
Legend:
X      = Self
SYS    = Connection traversing PCIe as well as the SMP interconnect between NUMA nodes (e.g., QPI/UPI)
NODE   = Connection traversing PCIe as well as the interconnect between PCIe Host Bridges within a NUMA
node
PHB    = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
PXB    = Connection traversing multiple PCIe bridges (without traversing the PCIe Host Bridge)
PIX    = Connection traversing at most a single PCIe bridge
NV#    = Connection traversing a bonded set of # NVLinks
```

```
# nvidia-smi -i 0
Mon Mar 30 14:49:26 2026
+-----+
| NVIDIA-SMI 595.58.03              Driver Version: 595.58.03          CUDA Version: 13.2          |
+-----+-----+-----+
```

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	Off	00000000:86:00:0	Off	4%	Default	0
N/A	62C	P0	27W / 70W	0MiB / 15360MiB		N/A	

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	Usage
	ID	ID					
No running processes found							

- charger le GPU 0

```
# nvidia-smi --query-gpu=utilization.gpu --format=csv --loop=1 -i 0
```

- charger le GPU 1

```
# nvidia-smi --query-gpu=utilization.gpu --format=csv --loop=1 -i 1
```

Nvidia dans le Container LXC

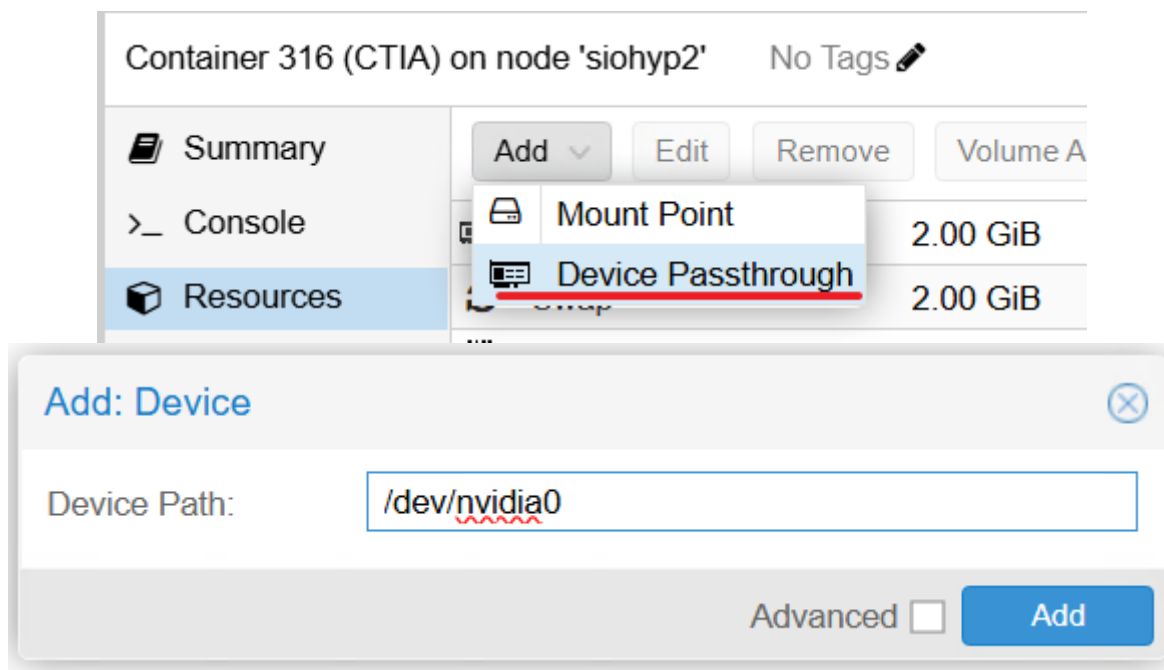
- mettre à jour le conteneur

```
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.

Le passthrough (ou PCI passthrough / USB passthrough / GPU passthrough) dans Proxmox permet de donner à un conteneur LXC, l'accès direct aux périphériques physiques (ici les GPU des cartes Nvidia) sans passer par la couche de virtualisation. Ces GPU de la Carte Nvidia sont retirés du contrôle de l'hôte Proxmox et attribués directement aux conteneurs LXC.

- 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-uvmm
Device (dev6)	/dev/nvidia-uvmm-tools

Ne plus installer le périphérique **/dev/nvidia-modset**

- Installez les drivers nvidia et la suite logicielle **cuda** dans le conteneur LXC (procédure semblable à celle de l'hôte 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 **nvidia-smi** pour confirmer que la carte est disponible et fonctionnelle sur votre container.

```
nvidia-smi
```

From:
/ - Les cours du BTS SIO

Permanent link:
</doku.php/reseau/cloud/proxmox/lxcnvidia?rev=1774876125>

Last update: **2026/03/30 15:08**

