

HW-Acceleration mit NVIDIA GPU in LXC-Container

Vorbereitung Proxmox

Proxmox auf die neueste Version aktualisieren:

```
apt update && apt dist-upgrade -y
```

Nach dem Upgrade neu starten:

```
reboot
```

Installation von git, gcc, make und den Header-Dateien:

```
apt install git gcc make pve-headers-$(uname -r)
```

Aktuellen NVIDIA-Treiber installieren:

Hierfür die folgende Liste als Referenz verwenden: [GitHub - keylase/nvidia-patch](https://github.com/keylase/nvidia-patch)

```
mkdir /opt/nvidia
```

```
cd /opt/nvidia
```

```
wget https://download.nvidia.com/XFree86/Linux-x86_64/<version>/NVIDIA-Linux-x86_64-<version>.run
```

```
chmod +x NVIDIA-Linux-x86_64-<version>.run
```

```
./NVIDIA-Linux-x86_64-<version>.run --no-questions --ui=none --disable-nouveau
```

Der Treiber erstellt die Datei /etc/modprobe.d/nvidia-installer-disable-nouveau.conf und deaktiviert den Nouveau-Treiber. Das bitte mit der erstellten .conf-Datei überprüfen:

```
more /etc/modprobe.d/nvidia-installer-disable-nouveau.conf
```

```
# generated by nvidia-installer
blacklist nouveau
options nouveau modeset=0
```

Proxmox zum Deaktivieren der Nouveau-Treiber neu starten:

```
reboot
```

Führe das NVIDIA-Installationsprogramm erneut aus, um die Treiberinstallation abzuschließen:

```
/opt/nvidia/NVIDIA-Linux-x86_64-<version>.run --no-questions --ui=none
```

Kontrolliere, ob der Befehl nvidia-smi richtig ausgeführt wird:

```
nvidia-smi
```

```
+-----+
| NVIDIA-SMI 525.60.11    Driver Version: 525.60.11    CUDA Version: 12.0    |
+-----+-----+-----+
| 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  Quadro P400      On      | 00000000:05:00:0 Off |          N/A |
| 34%   32C   P8      N/A /  N/A |    1MiB /  2048MiB |      0%      Default |
|                               |          N/A         |                      |
+-----+-----+-----+

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

Erstellen / Aktualisieren der Datei modules.conf für den Bootvorgang:

```
nano /etc/modules-load.d/modules.conf
```

```
# /etc/modules: kernel modules to load at boot time.

#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

nvidia
nvidia_uvm
```

Generiere das Initramfs Image mit der neuen modules.conf:

```
update-initramfs -u
```

Erstelle Regeln zum Laden der Treiber beim Booten sowohl für NVIDIA als auch für nvidia_uvm:

```
nano /etc/udev/rules.d/70-nvidia.rules
```

```
# /etc/udev/rules.d/70-nvidia.rules
# Create /nvidia0, /dev/nvidia1 ... and /nvidiactl when nvidia module is loaded
KERNEL=="nvidia", RUN+="/bin/bash -c '/usr/bin/nvidia-smi -L && /bin/chmod 666 /dev/nvidia*'"
#
# Create the CUDA node when nvidia_uvm CUDA module is loaded
KERNEL=="nvidia_uvm", RUN+="/bin/bash -c '/usr/bin/nvidia-modprobe -c0 -u && /bin/chmod 0666 /dev/nvidia-
uvm*'"
```

Installiere **GitHub - NVIDIA/nvidia-persistenced:**

```
git clone https://github.com/NVIDIA/nvidia-persistenced.git
```

```
cd nvidia-persistenced/init
```

```
./install.sh
```

Checking for common requirements...

sed found in PATH? Yes

useradd found in PATH? Yes

userdel found in PATH? Yes

id found in PATH? Yes

Common installation/uninstallation supported

Creating sample System V script... done.

Creating sample systemd service file... done.

Creating sample Upstart service file... done.

Checking for systemd requirements...

/usr/lib/systemd/system directory exists? No

/etc/systemd/system directory exists? Yes

systemctl found in PATH? Yes

systemd installation/uninstallation supported

Installation parameters:

User : nvidia-persistenced

Group : nvidia-persistenced

systemd service installation path : /etc/systemd/system

Adding user 'nvidia-persistenced' to group 'nvidia-persistenced'... done.

Installing sample systemd service nvidia-persistenced.service... done.

Enabling nvidia-persistenced.service... done.

Starting nvidia-persistenced.service... done.

systemd service successfully installed.

Prüfen, ob der Dienst läuft und aktiviert ist:

```
systemctl status nvidia-persistenced
```

```
# generated by nvidia-installer
blacklist nouveau
options nouveau modeset=0
root@pve:~# nano /etc/modules-load.d/modules.conf
root@pve:~# nano /etc/udev/rules.d/70-nvidia.rules
root@pve:~# systemctl status nvidia-persistenced
● nvidia-persistenced.service - NVIDIA Persistence Daemon
   Loaded: loaded (/lib/systemd/system/nvidia-persistenced.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-01-05 11:03:33 CET; 1h 47min ago
     Process: 1000 ExecStart=/usr/bin/nvidia-persistenced --user nvidia-persistenced (code=exited,
status=0/SUCCESS)
    Main PID: 1005 (nvidia-persiste)
      Tasks: 1 (limit: 38358)
     Memory: 776.0K
        CPU: 292ms
    CGroup: /system.slice/nvidia-persistenced.service
            └─1005 /usr/bin/nvidia-persistenced --user nvidia-persistenced

Jan 05 11:03:33 pve systemd[1]: Starting NVIDIA Persistence Daemon...
Jan 05 11:03:33 pve nvidia-persistenced[1005]: Started (1005)
Jan 05 11:03:33 pve systemd[1]: Started NVIDIA Persistence Daemon.
```

Neustart und Überprüfung, ob alle NVIDIA-Geräte richtig angezeigt werden:

```
reboot
```

```
ls -l /dev/nv*
```

```
crw-rw-rw- 1 root root 195,  0 Jan  5 11:03 /dev/nvidia0
crw-rw-rw- 1 root root 195, 255 Jan  5 11:03 /dev/nvidiaactl
crw-rw-rw- 1 root root 195, 254 Jan  5 11:03 /dev/nvidia-modeset
crw-rw-rw- 1 root root 507,  0 Jan  5 11:03 /dev/nvidia-uvdm
crw-rw-rw- 1 root root 507,  1 Jan  5 11:03 /dev/nvidia-uvdm-tools
```

Patchen des NVIDIA Treibers, um die Beschränkung der maximalen Sitzungen zu entfernen:

```
cd /opt/nvidia
```

```
git clone https://github.com/keylase/nvidia-patch.git
```

```
cd nvidia-patch
```

```
./patch.sh
```

Detected nvidia driver version: <version>

Attention! Backup not found. Copy current libnvcuvid.so to backup.

751706615c652c4725d48c2e0aaf53be1d9553d5 /opt/nvidia/libnvidia-encode-backup/libnvcuvid.so.<version>

ee47ac207a3555adccad593dbcda47d8c93091c0 /usr/lib/x86_64-linux-gnu/libnvcuvid.so.<version>

Patched!

Vorbereitung LXC-Container

Erstelle in Proxmox einen neuen LXC oder bearbeite die Container-Config (in meinem Fall Container 105) und füge die LXC-Gruppe und Mounts am Ende der .conf-Datei hinzu:

```
nano /etc/pve/lxc/105.conf
```

```
lxc.cgroup2.devices.allow: c 195:* rwm
```

```
lxc.cgroup2.devices.allow: c 507:* rwm
```

```
lxc.mount.entry: /dev/nvidia0 dev/nvidia0 none bind,optional,create=file
```

```
lxc.mount.entry: /dev/nvidiactl dev/nvidiactl none bind,optional,create=file
```

```
lxc.mount.entry: /dev/nvidia-uvdm dev/nvidia-uvdm none bind,optional,create=file
```

```
lxc.mount.entry: /dev/nvidia-modeset dev/nvidia-modeset none bind,optional,create=file
```

```
lxc.mount.entry: /dev/nvidia-uvdm-tools dev/nvidia-uvdm-tools none bind,optional,create=file
```

Die beiden obigen Gruppen (195 und 507) stammen aus der Ausgabe von `ls -l /dev/nv*`

In manchen Fällen ändert sich die Gruppe für -uvdm und -uvdm-tools beim Neustart. Wenn du feststellst, dass das passiert, füge alle Gruppen, die du siehst, zur allow-Liste hinzu, um zu verhindern, dass du die .conf-Datei ständig ändern musst.

Zum Beispiel:

```
lxc.cgroup.devices.allow: c 195:* rwm
```

```
lxc.cgroup.devices.allow: c 235:* rwm
```

```
lxc.cgroup.devices.allow: c 511:* rwm
```

Starte den LXC-Container und lade den gleichen NVIDIA-Treiber in den Container herunter:

Führe die Installation aus. Das Installationsprogramm fragt, ob libglvnd installiert werden soll, weil die Installation unvollständig ist. Hier dann bitte: "Don't install libglvnd". Alles andere wird mit "yes" beantwortet und Warnungen ignoriert.

```
mkdir /opt/nvidia
```

```
cd /opt/nvidia
```

```
wget https://download.nvidia.com/XFree86/Linux-x86_64/<version>/NVIDIA-Linux-x86_64-<version>.run
```

```
chmod +x NVIDIA-Linux-x86_64-<version>.run
```

```
./NVIDIA-Linux-x86_64-<version>.run --no-kernel-module
```

Führe nvidia-smi aus, um die Installation zu überprüfen. Prüfe auch, ob die richtigen NVIDIA-Geräte vorhanden sind:

```
nvidia-smi
```

```
+-----+
| NVIDIA-SMI 525.60.11    Driver Version: 525.60.11    CUDA Version: 12.0    |
+-----+-----+-----+
| 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  Quadro P400      Off | 00000000:05:00.0 Off |          N/A |
| 34%   32C   P8   N/A /  N/A |   1MiB /  2048MiB |   0%      Default |
|                               |          N/A |
+-----+-----+-----+

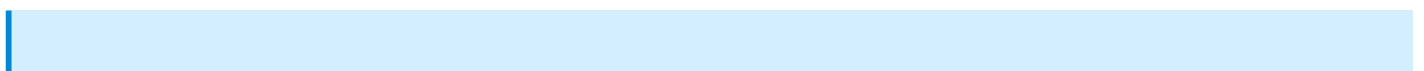
+-----+
| Processes:                                |
| GPU  GI  CI       PID   Type   Process name          GPU Memory |
|   ID ID              Usage   |
+=====+
=====|
| No running processes found              |
+-----+
```

```
ls -l /dev/nv*
```

```
crw-rw-rw- 1 root root 195, 254 Jan  5 10:03 /dev/nvidia-modeset
crw-rw-rw- 1 root root 507,  0 Jan  5 10:03 /dev/nvidia-uvm
crw-rw-rw- 1 root root 507,  1 Jan  5 10:03 /dev/nvidia-uvm-tools
crw-rw-rw- 1 root root 195,  0 Jan  5 10:03 /dev/nvidia0
crw-rw-rw- 1 root root 195, 255 Jan  5 10:03 /dev/nvidiactl
```

Anpassung und Überprüfung in Jellyfin

Logge dich in Jellyfin ein und aktiviere die Hardwarebeschleunigung



Um zu überprüfen, ob der Medienserver einen Thread zur Hardwarebeschleunigung startet, muss man nvidia-smi auf dem Host aufrufen, nicht aus dem LXC-Container heraus.

nvidia-smi

```
+-----+
| NVIDIA-SMI 525.60.11    Driver Version: 525.60.11    CUDA Version: 12.0    |
+-----+-----+-----+
| 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  Quadro P400      On   | 00000000:05:00.0 Off |          N/A |
| 34%   36C   P0     N/A /  N/A |  760MiB /  2048MiB |   86%    Default |
|                               |          N/A |
+-----+-----+-----+

+-----+
| Processes:                                |
| GPU  GI  CI       PID   Type   Process name          GPU Memory |
|  ID   ID                                   Usage     |
+=====+=====+=====+
=====|
|  0   N/A  N/A    49900    C   ...ib/jellyfin-ffmpeg/ffmpeg    756MiB |
+-----+-----+-----+
```