# 1.3.5 Hardware Passthrough

This chapter will discuss on how to pass through different devices to your virtual machines. That is you will give your VM direct access to the hardware instead of via a virtualization layer. This is extremely hand for passing through HDD for a TrueNas VM for instance and a GPU for a media or machine learning VM.

Now, hardware passthrough really means that only one VM can use the piece of hardware you are passing through. Yes, there are occasions where this is not the case but these are most of the time expensive pieces of technology such as GPU's that can virtually split up its VRAM to multiple containers. A solution however is to use LXC containers instead of VM's or have all your applications running inside the same VM. We will discuss both in this chapter


- 1. HDD/SSD VM Pass through
- 2. PCIe NVME Pass through
- 3. GPU Passthrough
- 4. NIC Pass Through

# 1. HDD/SSD VM Pass through

Pass-through disks in virtual machines (VMs) cannot be shared between VMs or used by the host system at the same time because they require exclusive access. This means if a disk is pass-through to one VM, it cannot be pass-through to another VM simultaneously. For security reasons this is great ! since, you will know if your host provider

The first passthrough option is the SATA passthrough option, where the sata ports are boldered straight on the motherboard. For instance the Supermicro X10SRL-F has 10 sata ports which would mean 10 HDD/SSD's could be passthrough. Now, to do this we need to open a fresh Shell inside our Proxmox server. This can either be done via SSH or via the GUI. Once, there we will

1. Display all drives with their corresponding serial numbers

```
lsblk -o +MODEL,SERIAL
```

2. Identify the VM ID's you want to connect the drives to e.g sda, sdb,..., etcetera
3. Next we will move to the by-id folder to list the full names of the drives you identified. For this we use

```
ls /dev/disk/by-id/
```

4. Once you identified the full names of the drives e.g "ata-KINGSTON_SA400S37120G_50026B7782550FFA" you can then connect them to the VM's you created by using the following commands for each drive:

```
qm set VM-ID -scsi1 /dev/disk/by-id/serialnumberofdrive_1
:
qm set VM-ID -scsin /dev/disk/by-id/serialnumberofdrive_n
```

5. As example:

```
qm set 1001 -scsi1 /dev/disk/by-id/ata-KINGSTON_SA400S37120G_50026B7782550FFA
```

6. Which should put out the message

```
update VM 1001: -scsi1 /dev/disk/by-id/ata-KINGSTON_SA400S37120G_50026B7782550FFA
```

7. Then go to the hardware section of your Truenas Proxmox VM and you will see that your drives are added to your VM as harddrives. Below an 8TB HDD has been passthrough to VM 1001.

Screenshot 2024-11-25 at 16.37.07.png

# 02. PCIe NVME Passthrough

Source: https://www.youtube.com/watch?v=-qm-m4q8hIU

init=/bin/bash mount -n -o remount,rw / mount -o remount,rw /

grub-mkpasswd-pbkdf2 Enter password: reenter password:

nano /etc/grub.d/00_header

grub PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.CB31EA9617F1C79171F60A120BE732F47F754E9F36C679B05371EE4CBCD9BEED9706A8DA0704B1E8A7C67F39E3AD3E05D6E80E20C9A4627F38A5C99A99134AF6.0721F4428CCF0354A53C6112DB38901C50395A7E87B8BA4509FBB560895065A30FAD19C8DC0AC29C1534D010EAA75D938D1D14F329BED8C9C1B5F09937C01F03

nano /etc/grub.d/40_custom

Passthrough video https://www.youtube.com/watch?v=McdVvFOhHP0

# 03. PCIE SATA HDD/SSD Passthrough

Source: https://www.youtube.com/watch?v=-qm-m4q8hIU

If you happen to have a PCIE SATA CARD then again like with the NVME drives you need to passthrough the PCIe cards to your VM.

grub.pbkdf2.sha512.10000.D0F8975DB0B74F6710290A9BF144086B4704D513AE2529F1E00DA05EA375BC52B3ACFC8D6641B3C6FCA74B927CC6FD97892B50827F34A761317DEC7BCAB127E7.1182DB95B13F915CFB9A82EAAA9598C3ADBB6D009EA0761D457428E057829F5790585B40073D8FC02B5903EF4855A51B8AFAC755E32B5EBCFB1D74FF90365321

GRUB Password Protection

nano /etc/grub.d/00_header

192.168.1.51
grub.pbkdf2.sha512.10000.2227CCCA08A1C17494648C345FDACE61849D693E4B6F83B1A8A6600FA69287F131999FDB075A5B57390C9836E55166E749F4B3BC4E5518DBC2586F3DCD1882D0.7812FEBC4890B483BF8D12332CC92396E3EAB9D324CA2E7CBD50FB15759209336BA061E4F2ABC81347A4818487CFAC92CA46017C311D0EDF4996FF9D88BD4EB3

set superusers="student" passwd_pbkdf2

update-grub

Now, if you do not password protect your GRUB

grub.pbkdf2.sha512.10000.0C40B0FFA2985B390DE69CE90601232DECE18DABB897FD3FD860D6073B4ADABB2EB71BBEEF7E3D5946F32DC8D6A571DEEFEA6DF323C7B2DC88F89AA43057290F.4329D094ED9B2E160390FD12FCA34D6DF2F03999638CF603DCD829687537BDD4FC452E8485138E2AEC4BB9DF9EAB75C9F677E2D4A884EEFE70725ADD5C5FE8F6

```
grub-mkpasswd-pbkdf2
```

Enter password: reenter password:

```
cat <<EOF
set superusers="student"
password_pbkdf2 student grub.pbkdf2.sha512.........
```

# TrueNAS

password_pbkdf2 student
grub.pbkdf2.sha512.10000.D3C2F7AC998E7FC41584CBA9CA0099FB9230C4D51275F2AB657110C7DEDF973AF7F566396B8C95AD8C6BA853B8767022934809616923CAB381F38758C6ACA0D2.E85E2D867DE0AB0414AF3A550D333AF2775C55622F18CB9D489090220E3103110FD76D7F2A874FBF75093485F47F9E4A763DFE349A51859DCDCE5B23C713B246

# TrueNAS no modifications

Configuring GRUB to Prevent Unauthorized Modification Edit the /boot/grub2/grub.cfg file as root. Add the following line to the file: set superusers="username" (replace “username” with the desired username). Add the following line to the file: password username password (replace “username”

and “password” with the desired credentials).

password_pbkdf2 student
grub.pbkdf2.sha512.10000.CA227ECF97D179F0AAC709A0CCB9C5F2238FF017B2BC00445DE0668
02BD6885B271A21DD00692C851A358148E47DF95FB73E197FC2E93F05DCF4EDE18EEA8FB5.1D5
FCFB7D7375AD0C68095836A46CD04BE7AC0E90CAAD9D21B6DE1D8FDD4161C9C34DE3C62C5EF
72CBA64A48E122A0D946744C6B1A3ED045858222A6A524F879

cat <<EOF set superusers="student" password_pbkdf2 student
grub.pbkdf2.sha512.10000.CE885BE10C3EFF3B0B597A163863593E8103C0FEB1F73BABAF044152
4EBA22F301B22679B7A24819F82665DD82B4AE1F68E01726068B77097F7A045D92CD9D4B.54A9
F2079EFCAC8EB0DECDF76AA298BCCB6639B7F9C2EB2B088AB0BF328AB89409112542F33D74894
F194938CEE625314C7A1B4A5C7055F5304E9B1308A5485E

qm set 1002 -scsi1 /dev/disk/by-id/ata-ST18000NM000J-2TV103_ZR502M9E

qm set 1002 -scsi2 /dev/disk/by-id/ata-ST18000NM000J-2TV103_ZR502F8H

# 2. PCIe NVME Pass through

# 3. GPU Passthrough

Source: https://techfuelhq.com/tutorials/gpu-passthrough-proxmox-vm-2026/

Great source for GPU Pass Through in a proxmox VM!

Source for which apps use GPU: https://www.xda-developers.com/docker-containers-that-use-your-gpu/

In this section we will explain how you can pass through a GPU device to a virtual machine. Most GPU's can at least be passed through to one VM, however there are high end GPU's available that can be split up in multiple virtual GPU's. However, we will focus in this section on passing through one GPU to one VM. I would recommend to pass through the GPU to your Docker VM since you can then leverage the GPU for multiple applications. For instance for Jellyfin and Frigate transcoding tasks.

The thing with GPU pass through is that it is different for each brand and it is not always that simple.

We will first discuss the configurations that need to happen for every GPU pass through. At this moment most GPU's are either from Intel or from Invidia. Hence, we will focus on both of these right now.

# 4. NIC Pass Through