

Proxmox - Hardware durchreichen in VM (KVM)

Um echte Hardware in die Virtuelle Maschine durchzureichen (KVM)

- Einschalten von vt-d für Intel CPUs (iommu)
- VGA Grafikkarte in VM (KVM) durchreichen

Einschalten von vt-d für Intel CPUs (iommu)

Immou aktivieren für Intel CPUs. Hier wird unterschiedn ob es ein UEFI System oder legacy System ist mit oder ohne zfs.

Vorraussetzung ist das im BIOS alles mit vt-d eingeschlatet ist und somit das Mainboard und CPU das auch unterstützen.

Überprüfe welches System vorliegt

!!!!!!VGA PASSTHROUGH funktioniert nur wenn das Host Betriebssystem im EFI Boot installiert wurde!!!!!!

```
ls /sys/firmware/efi
```

Wenn kein Ergebnis vor liegt ist es BIOS legacy boot.

Unter Legacy Boot (also nicht EFI) in der Datei oder EFI ohne ZFS!

```
nano /etc/default/grub
```

Dort

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
```

zu

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on"
```

ändern und

```
update-grub
```

ausführen.

Unter EFI Boot mit ZFS in der Datei

```
nano /etc/kernel/cmdline
```

Dort

```
root=ZFS=rpool/ROOT/pve-1 boot=zfs
```

zu

```
root=ZFS=rpool/ROOT/pve-1 boot=zfs intel_iommu=on
```

ändern und

```
pve-efiboot-tool refresh
```

ausführen.

Module beim start laden. Der Modules Datei anfügen

```
nano /etc/modules
```

```
vfiio  
vfiio_iommu_type1  
vfiio_pci  
vfiio_virqfd
```

Wenn Grafikkarten durchgereicht werden sollen müssen noch die Treiber geblacklistet werden. ansonsten kann dieser Teil übersprungen werden.

```
echo "blacklist radeon" >> /etc/modprobe.d/blacklist.conf  
echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf  
echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf  
update-initramfs -u
```

Hier gehts weiter. Wenn der Grafikkarten teil übersprungen wurde.
Nun die Maschine neustarten. Ob iommu aktiv ist kann man mittels.

```
dmesg | grep IOMMU
```

feststellen.

Ausgabe:

```
[ 0.028060] DMAR: IOMMU enabled  
[ 0.090496] DMAR-IR: IOAPIC id 2 under DRHD base 0xfed90000 IOMMU 0
```

Sollte nichts ausgegeben werden, im bios nochmals schauen ob vt-d aktiv ist.

Nun `iommu_unsafe_interrupts` setzen

```
echo "options vfio_iommu_type1 allow_unsafe_interrupts=1" > /etc/modprobe.d/iommu_unsafe_interrupts.conf  
update-initramfs -u
```

Für ein Funktionieren `pci passthrough` müssen alle Geräte die in eine VM durchgereicht werden sollen in einer eigenen Gruppe befinden. Sprich alleine sein. Zur Not PCI Karte umstecken.
Um überhaupt separate IOMMU Gruppen zu haben, muss das feature ACS (Access Control Services, im BIOS aktiv sein.

```
find /sys/kernel/iommu_groups/ -type l
```

Ausgabe: In unserm Beispiel ist es die 01:00, diese steckt aber noch mit 00:01.0 (00:01.0 PCI bridge: Intel Corporation 6th-10th Gen Core Processor PCIe Controller (x16) (rev 05) in einer Gruppe.

Wenn PCI Karte umbauen nicht geht, so wie bei mir.

Dann dem `group loader` noch

```
pcie_acs_override=downstream
```

Unter legacy

```
nano /etc/default/grub
```

```
update grub
```

Unter EFI

```
nano /etc/kernel/cmdline
```

```
pve-efiboot-tool refresh
```

und danach egal welches system

```
update-initramfs -u
```

mit geben.

Ausgabe ohne PCI KArte umsetzten oder grub Befehl Downstream

```
/sys/kernel/iommu_groups/7/devices/0000:00:1c.7  
/sys/kernel/iommu_groups/5/devices/0000:00:1c.0  
/sys/kernel/iommu_groups/3/devices/0000:00:16.0  
/sys/kernel/iommu_groups/11/devices/0000:05:00.0  
/sys/kernel/iommu_groups/1/devices/0000:00:01.0  
/sys/kernel/iommu_groups/1/devices/0000:01:00.0  
/sys/kernel/iommu_groups/1/devices/0000:01:00.1  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.2  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.0  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.3  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.4  
/sys/kernel/iommu_groups/6/devices/0000:00:1c.5  
/sys/kernel/iommu_groups/4/devices/0000:00:17.0  
/sys/kernel/iommu_groups/2/devices/0000:00:14.2  
/sys/kernel/iommu_groups/2/devices/0000:00:14.0  
/sys/kernel/iommu_groups/10/devices/0000:03:00.0  
/sys/kernel/iommu_groups/0/devices/0000:00:00.0  
/sys/kernel/iommu_groups/9/devices/0000:02:00.0
```

Ausgabe mit grub parameter downstream. Es hat geklappt

```
/sys/kernel/iommu_groups/7/devices/0000:00:1c.7  
/sys/kernel/iommu_groups/5/devices/0000:00:1c.0  
/sys/kernel/iommu_groups/3/devices/0000:00:16.0  
/sys/kernel/iommu_groups/11/devices/0000:03:00.0  
/sys/kernel/iommu_groups/1/devices/0000:00:01.0  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.2  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.0  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.3  
/sys/kernel/iommu_groups/8/devices/0000:00:1f.4  
/sys/kernel/iommu_groups/6/devices/0000:00:1c.5
```

```
/sys/kernel/iommu_groups/4/devices/0000:00:17.0
/sys/kernel/iommu_groups/12/devices/0000:05:00.0
/sys/kernel/iommu_groups/2/devices/0000:00:14.2
/sys/kernel/iommu_groups/2/devices/0000:00:14.0
/sys/kernel/iommu_groups/10/devices/0000:02:00.0
/sys/kernel/iommu_groups/0/devices/0000:00:00.0
/sys/kernel/iommu_groups/9/devices/0000:01:00.0
/sys/kernel/iommu_groups/9/devices/0000:01:00.1
```

Die Grafikkarte ist jetzt unter der Gruppe 9 kann bei euch anders sein . Zu der Grafikkarte gehören zwei Einträge.

```
/sys/kernel/iommu_groups/9/devices/0000:01:00.0
/sys/kernel/iommu_groups/9/devices/0000:01:00.1
```

ff

VGA Grafikkarte in VM (KVM) durchreichen

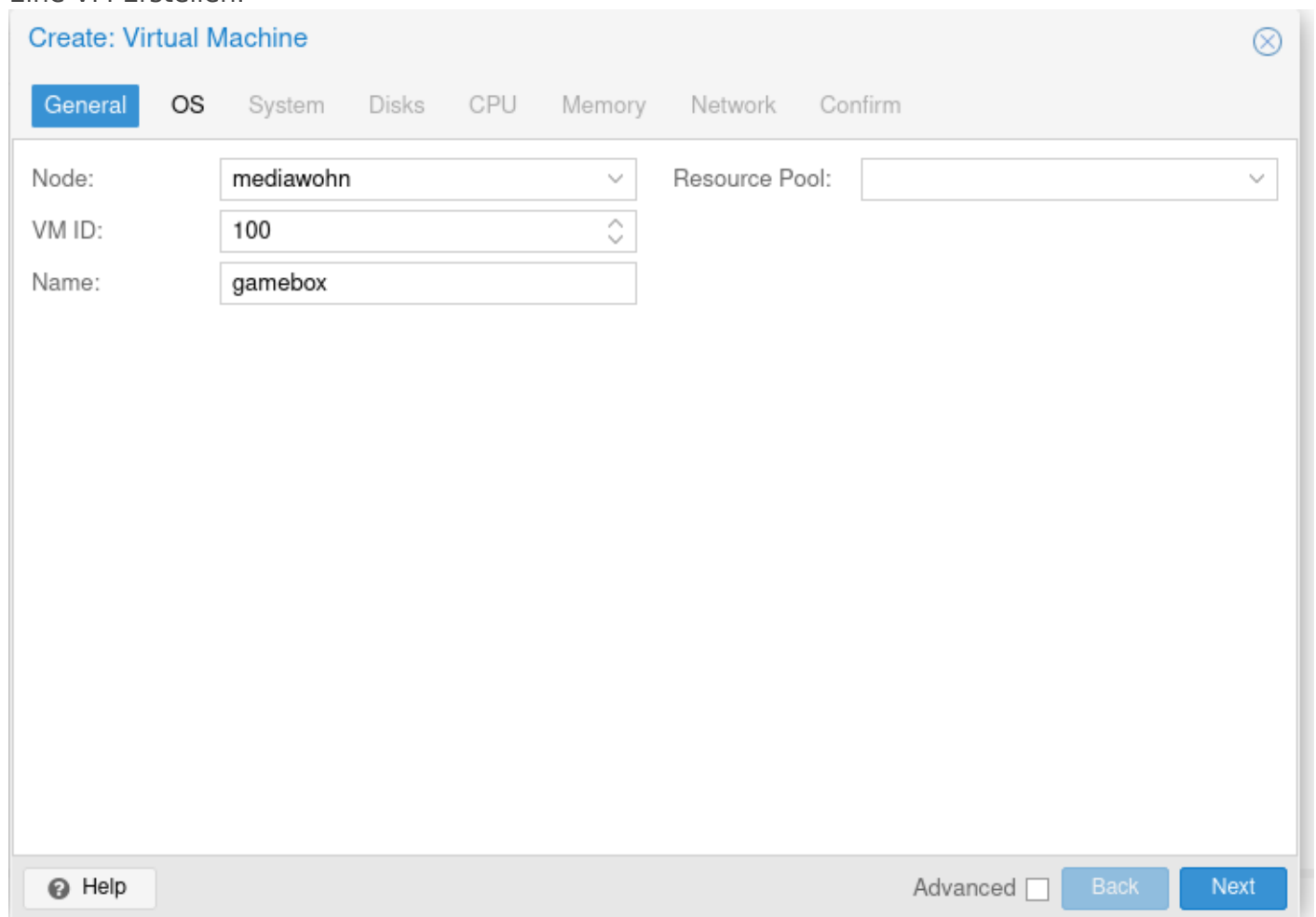
Vorraussetzung wie im vorherigen Seite. Das auch die Grafikkarten geblacklistet wurden.

!!!!VGA PASSTHROUGH funktioniert nur wenn das Host Betriebssystem im EFI Boot installiert wurde!!!!

!!!!Wenn eine Onboard Grafikkarte verfügbar ist, diese als Primär im BIOS einstellen!!!!

Siehe: [Einschalten von vt-d für Intel CPUs \(iommu\)](#)

Eine VM Erstellen:



The screenshot shows the 'Create: Virtual Machine' wizard in Proxmox VE. The 'General' tab is selected, and the following fields are visible:

- Node:** mediawohn (dropdown menu)
- Resource Pool:** (empty dropdown menu)
- VM ID:** 100 (spin box)
- Name:** gamebox (text input field)

At the bottom of the wizard, there is a 'Help' button, an 'Advanced' checkbox (unchecked), and 'Back' and 'Next' buttons.

Nun ISO und OS auswählen, hier eine Windows 10 Machine mit ISO. Wenns ein Linux ist,linux auswählen.

Hier sind die Einstellungen egal.

Create: Virtual Machine ✕

General **OS** System Disks CPU Memory Network Confirm

Use CD/DVD disc image file (iso)

Storage: ▼

ISO image: ▼

Use physical CD/DVD Drive

Do not use any media

Guest OS:

Type: ▼

Version: ▼

Advanced

Hier:

- Machine q35 auswählen.
- BIOS : OVMF (UEFI) auswählen
- Add EFI Disk : Haken rein
- Storage dazu auswählen
- Pre-Enroll-Keys : Haken rein
- Qemu Agent : Haken rein
- Add TPM : Haken rein
- Storage dazu auswählen

Create: Virtual Machine ⊗

General OS **System** Disks CPU Memory Network Confirm

Graphic card:	<input type="text" value="Default"/>	SCSI Controller:	<input type="text" value="VirtIO SCSI"/>
Machine:	<input type="text" value="q35"/>	Qemu Agent:	<input checked="" type="checkbox"/>
Firmware			
BIOS:	<input type="text" value="OVMF (UEFI)"/>	Add TPM:	<input checked="" type="checkbox"/>
Add EFI Disk:	<input checked="" type="checkbox"/>	TPM Storage:	<input type="text" value="local-zfs"/>
EFI Storage:	<input type="text" value="local-zfs"/>	Version:	<input type="text" value="v2.0"/>
Format:	<input type="text" value="Raw disk image (raw)"/>		
Pre-Enroll keys:	<input checked="" type="checkbox"/>		

ⓘ Help Advanced Back Next

Nun wieder wie gehabt eine Festplatte zuweisen. Ich nehme hier virtio und 128 GB.
Denkt bei Windows dran, im Anschluss ein zweites CD/DVD Laufwerk mit den Virtio Treibern hinzuzufügen.

Download hier : <https://github.com/virtio-win/virtio-win-pkg-scripts/blob/master/README.md>
Die latest Stable und die guest tools sind im Anhang dieses Artikels.

Nun die CPU settings

Create: Virtual Machine ⊗

General OS System Disks **CPU** Memory Network Confirm

Sockets:	<input type="text" value="1"/>	Type:	<input type="text" value="Default (kvm64)"/>
Cores:	<input type="text" value="4"/>	Total cores:	4

Memory Settings

Create: Virtual Machine ⓧ

General OS System Disks CPU **Memory** Network Confirm

Memory (MiB):

? Help Advanced Back Next

Network Settings.

Hier auf VirtIO stellen

Das schöne ist das Win 110/1 die Treiber nicht kennt und somit ein lokales Konto verwendet/erstellt werden kann,

Create: Virtual Machine ⊗

General OS System Disks CPU Memory **Network** Confirm

No network device

Bridge: Model:

VLAN Tag: MAC address:

Firewall:

? Help Advanced Back Next

Nun auf finisch, haken bei Start after creation raus.

Create: Virtual Machine ⊗

General OS System Disks CPU Memory Network **Confirm**

Key ↑	Value
agent	1
bios	ovmf
cores	4
efidisk0	local-zfs:1,efitype=4m,pre-enrolled-keys=1
ide2	local:iso/Win10_21H2_German_x64.iso,media=cdrom
machine	q35
memory	4096
name	gamebox
net0	virtio,bridge=vibr0,firewall=1
nodename	mediawohn
numa	0
ostype	win10

Nun noch ein zweites CD/DVD Laufwerk erstellen für die VirtIO treiber. Aber nur nötig bei Windows VMS. sonst kann dieser Teil übersprungen werden.

Nun die Grafikkarte hinzufügen.

Dazu auf Add Hardware PCI, dann die Grafikkarte auswählen. mit der 0 bei der ID am ende.

ID ↑	IOMM...	Vendor	Device	Medi...
0000:00:14.0	-	Intel Corporation	100 Series/C230 Series Chipset Family USB 3.0 xHCI Controller	No
0000:00:14.2	-	Intel Corporation	100 Series/C230 Series Chipset Family Thermal Subsystem	No
0000:00:16.0	-	Intel Corporation	100 Series/C230 Series Chipset Family MEI Controller #1	No
0000:00:17.0	-	Intel Corporation	Q170/Q150/B150/H170/H110/Z170/CM236 Chipset SATA Controll...	No
0000:00:1f.3	-	Intel Corporation	100 Series/C230 Series Chipset Family HD Audio Controller	No
0000:00:1f.4	-	Intel Corporation	100 Series/C230 Series Chipset Family SMBus	No
0000:01:00.0	-	NVIDIA Corporation	GM107 [GeForce GTX 750 Ti]	No
0000:01:00.1	-	NVIDIA Corporation	GM107 High Definition Audio Controller [GeForce 940MX]	No
0000:02:00.0	-	Spin Master Ltd.	PCIe Video Bridge	No
0000:05:00.0	-	Realtek Semicond	RTL 8111/8168/8411 PCI Express Gigabit Ethernet Controller	No

Hier Haken bei :
All Funcktions rein
Primary GPU rein
Rom-BAR rein
PCI-Express rein

Device: 0000:01:00.0 MDev Type:

All Functions: Primary GPU:

ROM-Bar: PCI-Express:

Vendor ID: From Device Sub-Vendor ID: From Device

Device ID: From Device Sub-Device ID: From Device

Help Advanced Add

Damit die Graffikarte nicht geladen wird, müssen wir die IDs black listen.

Dazu erstmal die IDs bekommen.

Die bekommen wir mittels `lspci -n -s 01:00`

01:00 ist die erste PCI ID usnerer Graffikarte (Siehe Screenshot Grafikkarten auswahl in derVM

```
lspci -n -s 01:00
```

Ausgabe:

```
01:00.0 0300: 10de:1380 (rev a2)
```

```
01:00.1 0403: 10de:0fbc (rev a1)
```

Nun die Beiden IDS blacklisten. Diese IDS sind von der GPU und der Soundkarte auf der GPU (HDMI hat Sound)

```
echo "options vfio-pci ids=10de:1380,10de:0fbc" > /etc/modprobe.d/vfio.conf
```

Den Grub anhängen

```
initcall_blacklist=sysfb_init vfio_pci.ids=10de:1380,10de:0fbc iommu=pt  
vfio_iommu_type1.allow_unsafe_interrupts=1"
```

In der GRUB conf hinzufügen

Für legacy

```
nano /etc/default/grub  
update grub
```

Unter EFI mit ZFS

Achtung in der cmdline kein gänsefüßchen am ende!!!!

```
nano /etc/kernel/cmdline  
pve-efiboot-tool refresh
```

hinzufügen

dann initrafs aktualisieren

```
update-initramfs -u
```

Für Soundkartenasugabe kratzen verzerrung oder gar keinen Ton mehr

```
echo "options snd-hda-intel enable_msi=1" >> /etc/modprobe.d/snd-hda-intel.conf
```

Nun reboot, des Proxmox hosts.

```
reboot
```

Wenn alles geklappt hat sollte der Loginscreen nicht mehr kommen, sondern mitten im Boottext stecken bleiben, da die Grafikkarte hier abgewürgt wurde.

Nun nur noch die conf file der vm folgende Zeile am ende hinzufügen.

```
nano /etc/pve/qemu-server/<vmid>.conf  
nano /etc/pve/qemu-server/100.conf
```

```
...  
args: -cpu 'host,+kvm_pv_unhalt,+kvm_pv_eoi,hv_vendor_id=1234567890a4,kvm=off' -machine  
'type=q35,kernel_irqchip=on'
```

Speichern.

Nun die Vm starten, voila oder aber nein Kein Bild.
Dann Fehlersuche hier

Fehlersuche

Wenn das Bild auf no signal bleibt.

In der /var/log/syslog schauen

```
tail -f /var/log/syslog
```

Hier der LOG Fehler. Bedeutet nicht im EFI Boot modus installiert der Host

```
Oct 21 16:15:57 mediawohn kernel: [ 32.695520] vfio-pci 0000:01:00.0: No more image in the PCI ROM
Oct 21 16:15:58 mediawohn pvedaemon[1616]: <root@pam> end task
UPID:mediawohn:000007B2:00000C05:6352A99B:qmstart:100:root@pam: OK
Oct 21 16:15:59 mediawohn kernel: [ 34.964385] vfio-pci 0000:01:00.0: No more image in the PCI ROM
Oct 21 16:15:59 mediawohn kernel: [ 34.964435] vfio-pci 0000:01:00.0: No more image in the PCI ROM
```

weiterer Fehler:

Ab Kernel 5.15.+ muss in der grub conf

```
video=efib:off
```

durch

```
initcall_blacklist=sysfb_init
```

ersetzt werden

```
- Invalid PCI ROM header signature
- kernel: vfio-pci 0000:01:00.0: BAR 1: can't reserve [mem 0x4020000000-0x402ffffff 64bit pref]
```

Jetzt bleibt nur noch Invalid PCI ROM header signature und cannot read device übrig

```
vfio-pci: Cannot read device rom at 0000:01:00.0
```

```
Oct 21 20:27:02 pve kernel: [ 298.009369] vfio-pci 0000:01:00.0: vfio_ecap_init: hiding ecap 0x1e@0x258
Oct 21 20:27:02 pve kernel: [ 298.009382] vfio-pci 0000:01:00.0: vfio_ecap_init: hiding ecap 0x19@0x900
Oct 21 20:27:02 pve kernel: [ 298.010765] vfio-pci 0000:01:00.0: Invalid PCI ROM header signature: expecting
0xaa55, got 0xffff
```

Bevor es weiter geht testen ob die Grafikkarte UEFI OMFV fähig ist

```
apt install git gcc make
git clone https://github.com/awilliam/rom-parser
cd rom-parser
make
```

Nun das rom aus der Grafikkarte holen ud testen

```
cd /sys/bus/pci/devices/0000:01:00.0/
echo 1 > rom
cat rom > /root/image.rom
echo 0 > rom
cd ~
cd rom-parser
./rom-parser /root/image.rom
```

Ausgabe :

Hier ein Fehler, am ende.

Dies bedeutet das Auf das Rom nicht zugegriffen werden kann. In meinem Beispiel hab Ich legacy Boot.

Denn am einfachsten gets wenn das Hostsystem gleich als UEFI installiert ist.

Die gegenprobe mach ich noch. Also für die Grafikkarte ein VBIOS holen

```
alid ROM signature found @0h, PCIR offset 190h
[]PCIR: type 0 (x86 PC-AT), vendor: 10de, device: 1380, class: 030000
[]PCIR: revision 0, vendor revision: 1
Error, ran off the end
```

VBIOS holen:

unter <https://www.techpowerup.com/vgabios/> sein passendes rom suchen.

Dann dort mit rechtecklick den Downlink undklicken und adresse kopiren meins liegt unter

<https://www.techpowerup.com/vgabios/175522/Asus.GTX750Ti.2048.141104.rom>

Nun per wget das image ablegen.

Wenn das passende rom nicht dabei selbst dumpen mit nvflash

<https://www.techpowerup.com/download/nvidia-nvflash/>

Usage:

```
chmox +x nvflash
./nvflash --save <filename>
```

ansonsten mit wget das richtige file holen, wenn bei techpowerup verfügbar.

Beispiel für meine File

```
wget -O /usr/share/kvm/gtx750ti2048m.bin
https://www.techpowerup.com/vgabios/175522/Asus.GTX750Ti.2048.141104.rom
```

ROM File ablegen unter
/usr/share/kvm/meinromfile.bin

Nun die vm conf anpassen und die romfile reinpacken ohne pfad nur den dateinamen

```
nano /etc/pve/qemu-server/1<VMID>.conf  
nano /etc/pve/qemu-server/100.conf
```

Dort an die vga Zeile anfügen

```
,romfile=vbios.bin
```

Beispiel:

```
,romfile=gtx750ti2048m.bin
```

```
hostpci0: 0000:01:00,pcie=1,x-vga=1,romfile=gtx750ti2048m.bin
```

Nun an die Syslog hängen

```
tail -f /var/log/syslog
```

ff