Einrichten VM

Installation Ubuntu 24.04.3 LTS (Minimal Server)

1. ISO herunterladen

Offizielle Quelle:

https://ubuntu.com/download/server

- Variante: **Ubuntu Server 24.04.3 LTS**
- Image: `ubuntu-24.04.3-live-server-amd64.iso`

---

2. VM in Proxmox anlegen

1. In der Proxmox Web-GUI: **Create VM**
   - Name: `KI-VM`
   - ISO: `ubuntu-24.04.3-live-server-amd64.iso` (zuvor hochgeladen in Proxmox Storage)
   - System:
     - BIOS: **OVMF (UEFI)**
     - Machine: **q35**
     - Graphic Card: **none** (GPU wird durchgereicht)
   - Disks:
     - Speicher auf **Crucial NVMe 4 TB** oder System-SSD je nach Planung
     - Größe: mind. 250 GB (je nach Modellgrößen)
     - Cache: **Write back (unsafe)** (für Performance)
   - CPU:
     - 6 Cores (i5-13400 hat 10 → 6 reichen für KI-Workloads, Rest für Host)
     - Typ: **host**
   - RAM:
     - 32 GB (von 128 GB gesamt)

    - Ballooning: deaktivieren (konstante Zuweisung)
- Netzwerk: **VirtIO (paravirtualized)**
- Hardware: **GPU & Audio-Device** via PCI Passthrough hinzufügen (wie vorher eingerichtet).

---

3. Ubuntu Installation

1. Boot von ISO → Auswahl: **Install Ubuntu Server**
2. Sprache: `Deutsch` (oder Englisch, je nach Vorliebe)
3. Tastatur: `Deutsch`
4. Netzwerk:
    - `ens18` (VirtIO) automatisch via DHCP oder manuell konfigurieren
    - Empfehlung: **statische IP** für die VM (z. B. 192.168.33.200)
5. Storage:
    - Geführte Installation auf die virtuelle Disk
    - Partitionierung: Standard (LVM möglich, aber nicht zwingend)
6. Benutzer anlegen:
    - Username: `kiadmin`
    - Passwort: sicher setzen
    - SSH-Server installieren: **Ja**
7. Snap-Pakete: alle abwählen (nicht benötigt)
8. Installation starten → Neustart nach Abschluss

---

4. Erste Anpassungen nach der Installation

Nach Login per SSH oder Console:

a) System aktualisieren

```
sudo apt update && sudo apt upgrade -y
sudo reboot
```

b) Nützliche Basis-Tools installieren

```
sudo apt install -y htop ncdu git curl wget unzip zip tar net-tools iftop lsb-release pciutils
```

c) SSH konfigurieren

```
sudo nano /etc/ssh/sshd_config
```

Empfohlen:

- `PermitRootLogin no`
- `PasswordAuthentication no` (falls SSH-Key genutzt wird)

Dann:

```
sudo systemctl restart ssh
```

d) Zeitsynchronisation prüfen

```
timedatectl set-timezone Europe/Berlin
timedatectl status
```

---

5. GPU-Treiber vorbereiten

Da die GPU durchgereicht wird, braucht die VM die NVIDIA-Treiber:

a) Repository aktivieren

```
sudo apt install -y software-properties-common
sudo add-apt-repository ppa:graphics-drivers/ppa -y
sudo apt update
```

b) NVIDIA-Treiber installieren

```
sudo apt install -y nvidia-driver-550 nvidia-utils-550
```

c) Neustart & Test

`nvidia-smi`
→ sollte GPU-Daten (Modell RTX 5060 Ti, Treiber, CUDA-Version) anzeigen.

---

6. Optional: Basis-Optimierungen für KI-Workloads

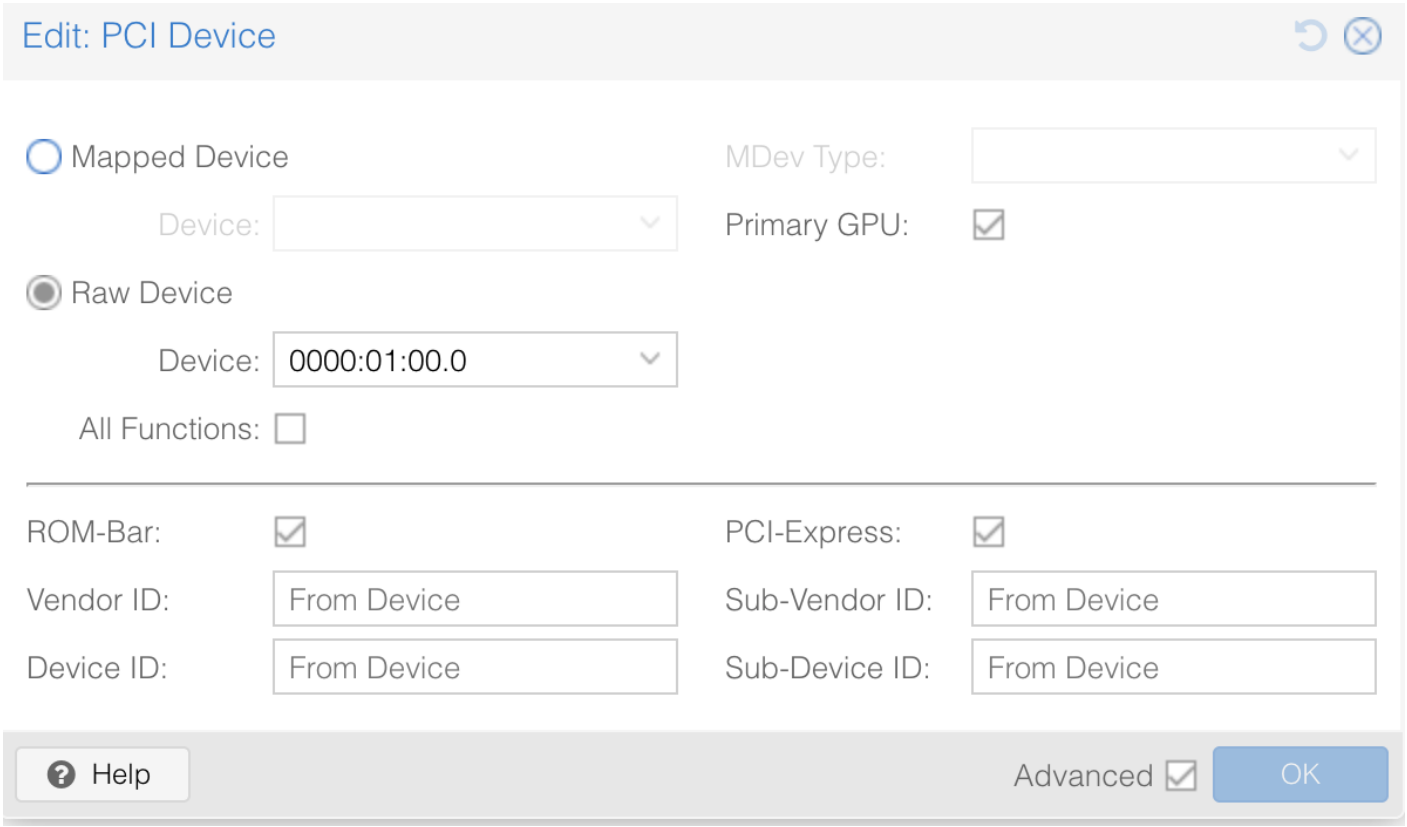
- **Swappiness reduzieren** (damit RAM bevorzugt genutzt wird):

  ```
  echo 'vm.swappiness=10' | sudo tee -a /etc/sysctl.conf
  sudo sysctl -p
  ```
- **Transparent Hugepages deaktivieren** (manchmal Performance-Gewinn bei LLMs):

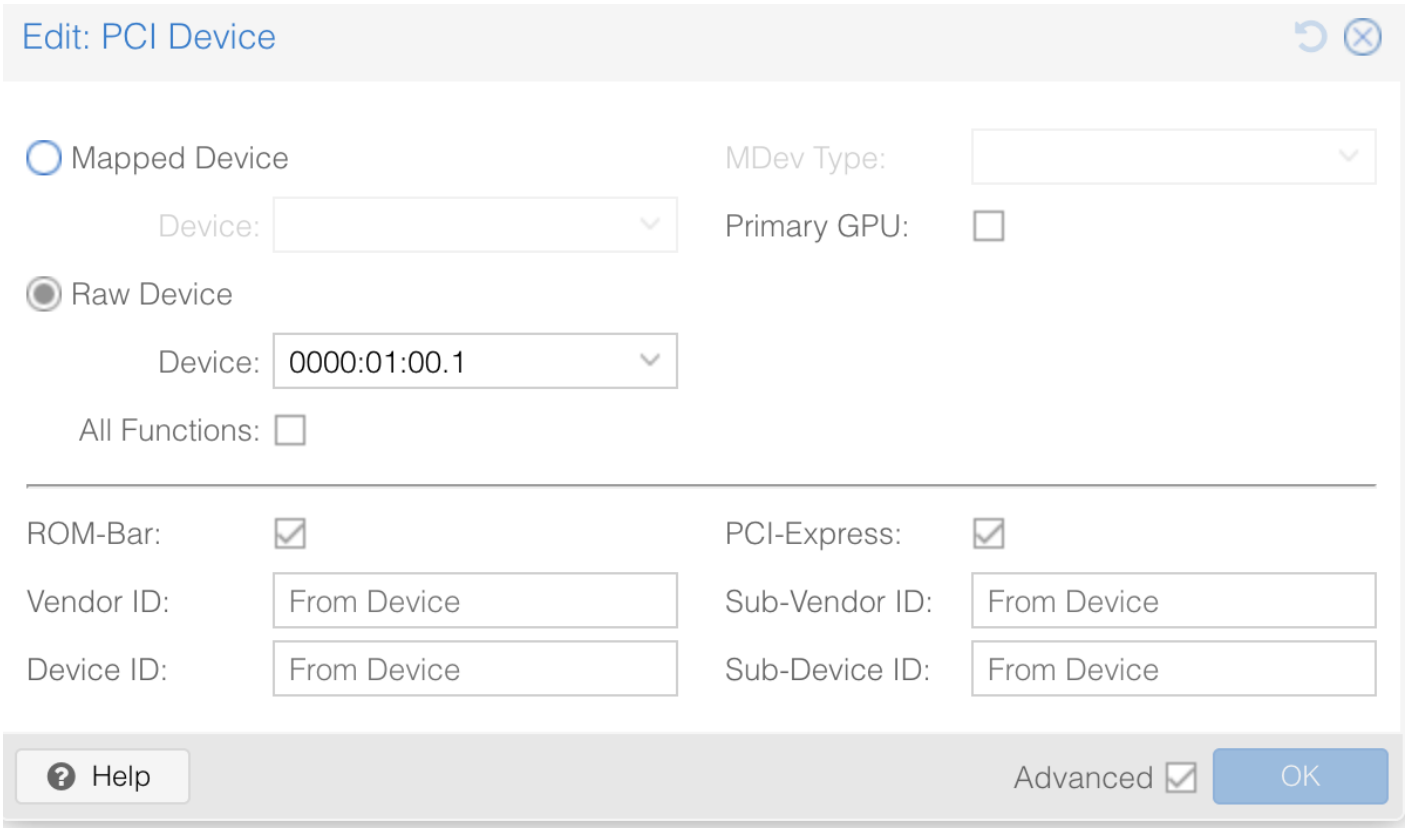
  ```
  echo never | sudo tee /sys/kernel/mm/transparent_hugepage/enabled
  ```
- **ufw aktivieren** (Firewall, falls nicht durch Docker geregelt):

  ```
  sudo apt install ufw -y
  sudo ufw default deny incoming
  sudo ufw default allow outgoing
  sudo ufw allow ssh
  sudo ufw enable
  ```

Konfiguration GPU:



Konfiguration GPU-Audio:



Revision #3
Created 2025-09-23 06:04:14 UTC by Peter Leibling
Updated 2025-09-23 06:40:19 UTC by Peter Leibling