Hailo■8 / Hailo■8L PCIe Passthrough – Proxmox → Ubuntu Guest (Quickstart)

Audience: Proxmox VE host with Hailo PCIe cards; Ubuntu 22.04+ VM guest
Goal: Pass a Hailo card (e.g., `1e60:2864`) directly to a VM and validate with `hailortcli`

---

0) Identify the device on the host

```
lspci -nn | egrep -i 'hailo|1e60'
# Example:
# 03:00.0 Co-processor [0b40]: Hailo Technologies Ltd. Hailo-8 AI Processor [1e60:2864]
# 04:00.0 Co-processor [0b40]: Hailo Technologies Ltd. Hailo-8 AI Processor [1e60:2864]
```

Record the BDF(s), e.g. `0000:03:00.0` and/or `0000:04:00.0`.

---

1) Enable IOMMU on the Proxmox host (one time)

Edit GRUB:

```
sudo sed -i 's/GRUB_CMDLINE_LINUX_DEFAULT=.*/GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on iommu=pt"/' /e
sudo update-grub
```

Enable VFIO modules:

```
cat | sudo tee /etc/modules-load.d/vfio.conf >/dev/null <<'EOF'
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
EOF
```

(Important) Prevent the host from binding the Hailo device with its native driver (so VFIO can claim it):

Bind Hailo to vfio-pci by vendor:device id echo "options vfio-pci ids=1e60:2864" | sudo tee /etc/modprobe.d/vfio-pci.conf

Rebuild initramfs sudo update-initramfs -u sudo reboot ```

After reboot, confirm the device is in its own IOMMU group and **uses vfio-pci**:

```
lspci -nnk -s 03:00.0
# Kernel driver in use: vfio-pci
```

---

2) Attach the device to a VM

In **Proxmox → VM → Hardware → Add → PCI Device**:

• **Device:** `0000:03:00.0` (or your BDF)

• Check **All Functions** (if multi-function), **Primary GPU** _off_ (not needed), **ROM-Bar** default

• Enable **PCI-Express** (pcie=1)

• Machine type: **q35**

• BIOS: **OVMF (UEFI)**

• CPU type: **host**

• (Optional) NUMA: On if your host/VM topology benefits

Config line (example in `/etc/pve/qemu-server/.conf`):

```
machine: q35
bios: ovmf
hostpci0: 0000:03:00.0,pcie=1
```

Start the VM.

---

3) Inside the Ubuntu guest

Verify the card is visible:

```
lspci -nn | egrep -i 'hailo|1e60'
```

Install Hailo Runtime/SDK (from Hailo's installer bundle), then check driver/module:

```
lsmod | grep -i hailo || true
sudo dmesg | egrep -i 'hailo|pci' | tail -n 50
hailortcli scan
```

If the module is missing, install DKMS package from the Hailo bundle or `hailort-driver` package for Ubuntu, then:

```
sudo modprobe hailo_pci
hailortcli scan
```

You should see each device listed (e.g., `0000:03:00.0`).

---

4) Quick functional test

If you have a HEF:
```bash
hailortcli run /path/to/model.hef --time-to-run 10 --measure-latency --measure-temp --bdf 0000:03:00.0
```

Successful output shows frames, HW latency ~2–8 ms depending on model and device.

---

5) Troubleshooting

• **Host driver in use is hailo_pci (not vfio-pci):** The device cannot passthrough.

• Ensure `/etc/modprobe.d/blacklist-hailo.conf` exists and `update-initramfs -u` + reboot.

• Ensure `vfio-pci` lists `1e60:2864` in `/sys/bus/pci/drivers/vfio-pci/new_id` or via the `options vfio-pci ids=…` file.

• **“Device is in use” when starting VM:** Some other host driver (or DPDK) grabbed it. Confirm with `lspci -nnk` and unbind/reboot.

• **IOMMU groups merge many devices:** Some motherboards/slots share ACS. Try a different slot; optionally enable ACS override (advanced).

• **VM boots but no device in guest:** Check VM config has `pcie=1`, machine `q35`, BIOS `ovmf`, and no host PCI errors in `dmesg`.

• **Multiple Hailo cards:** Add `hostpci1: ,pcie=1` for the second, etc. Verify each with `hailortcli scan`.

---

6) Security & persistence

• Keep the host blacklist + vfio bindings permanent (we used modprobe.d + initramfs).

• Snapshot the VM after a successful test.

• For production, restrict guest access to only the needed BDF(s).

Cheat sheet

• Vendor:Device ID: `1e60:2864`

• Host bind to VFIO: `/etc/modprobe.d/vfio-pci.conf` → `options vfio-pci ids=1e60:2864`

• Blacklist host driver: `/etc/modprobe.d/blacklist-hailo.conf` → `blacklist hailo_pci`

• Proxmox VM config example: `hostpci0: 0000:03:00.0,pcie=1` with `machine: q35`, `bios: ovmf`

• Guest validation: `hailortcli scan`