

Configure AMD GPU passthrough to a Windows 11 VM in Proxmox VE

Give a Windows 11 virtual machine exclusive use of an AMD Radeon with PCI Express passthrough: host IOMMU and isolation checks, an OVMF, Secure Boot, and TPM 2.0 build, exact-ID VFIO binding, and a staged cutover that keeps a recovery console until the GPU works.

PROXMOX VE 9.2 · VFIO PCIE PASSTHROUGH · AMD RADEON · WINDOWS 11 · HOW-TO GUIDE

General educational guidance, not affiliated with or endorsed by Proxmox Server Solutions GmbH, AMD, or Microsoft. Passthrough depends on the CPU, firmware, PCI topology, kernel, and drivers, so run the restart and power-cycle tests at the end before treating the build as operational. Validated on Proxmox VE 9.2.10. Published 2026-09-08.

2

components of the GPU move to the VM

the video component and its HDMI/DP audio; both must move together

2

recovery options

give the VM a virtual display again, or give the GPU back to Proxmox

PCI Express passthrough assigns a physical GPU to one virtual machine, which then runs the real AMD driver against real hardware. This guide builds a Windows 11 VM that meets Microsoft's firmware requirements, proves a recovery path, then reserves a Radeon and its audio function for VFIO (the Linux framework that holds a device for a VM instead of a host driver) and attaches them in two stages. That order keeps a working console available at every step, and the lifecycle tests at the end show whether the build survives restarts and shutdowns.

CONTEXT What passthrough costs

One VM gets exclusive control

A passed-through device is unavailable to the Proxmox host and every other VM while it is assigned. This design dedicates the whole consumer Radeon to one VM; it does not divide the card into shared vGPUs.

The host goes headless

The validated host is a single-GPU machine, so binding its one display adapter to VFIO leaves Proxmox without a local graphical console once the kernel initializes. Management continues over the web interface (<https://pve01.lab.example.com:8006> or the management address) and SSH, so do not reserve the GPU until both have been tested from another computer. Where the host CPU has integrated graphics, enabling it in firmware and setting it as the boot display keeps a local console as well, and the recovery steps below stay usable without the network.

The console cannot show the card

Proxmox cannot display a physical GPU's framebuffer through noVNC or SPICE. After the cutover, the picture comes from the monitor attached to the Radeon or from remote access inside Windows.

MAP The validated example

ITEM	VALIDATED VALUE
Node / Proxmox VE	pve01 , Proxmox VE 9.2.10, management at 192.168.1.70/24
VM network	Bridge vmbr0
Storage roles	local for ISOs; vmfast (NVMe LVM-thin) for VM disks; backup-local for local backups

FUNCTION	DEVICE	ID	HOST DRIVER BEFORE	IOMMU GROUP
Graphics	AMD Navi 32, reported as Radeon RX 7700 XT / RX 7800 XT	1002:747e	amdgpu	29
HDMI/DP audio	AMD Navi 31 HDMI/DP Audio	1002:ab30	snd_hda_intel	30

The subsystem identifies a Sapphire card, and the PCI output does not distinguish the RX 7700 XT from the RX 7800 XT marketing model, so the guide works from the PCI identity **1002:747e** rather than asserting a retail model. The audio function reports as Navi 31 on this Navi 32 card; that is the name the PCI ID database gives it. The two functions sit in separate IOMMU groups, which is an acceptable passthrough topology. Both go to the same Windows VM.

WHY Design decisions

Windows installs before the GPU moves

STAGED CUTOVER · A CONSOLE SURVIVES EVERY STEP

Install Windows on the Proxmox virtual display first, so the installation runs through noVNC. That also leaves time to verify the VirtIO drivers and the QEMU Guest Agent, and to configure and test remote access. Reserve and attach the physical GPU only after that recovery path works: first as a secondary adapter with the virtual console kept, then as the primary GPU.

Q35, OVMF, Secure Boot, and TPM 2.0

WINDOWS 11 REQUIREMENTS · PROXMOX'S RECOMMENDED PASSTHROUGH SETUP

Windows 11 requires UEFI with Secure Boot capability, TPM 2.0, two or more processor cores, 4 GB of RAM, and 64 GB of storage; the build meets all of those, with more processor, memory, and disk than the minimum.

Proxmox's passthrough documentation pairs the q35 machine type with PCI Express mode, and OVMF (the UEFI firmware for VMs) needs the GPU to carry a UEFI-capable ROM, one holding a Graphics Output Protocol (GOP) image, so check the card's ROM if firmware-stage output ever fails; the troubleshooting section says how.

Exact PCI IDs, because the audio module is shared

1002:747E + 1002:AB30 · MOTHERBOARD AUDIO KEEPS ITS DRIVER

The host's motherboard audio controller at **0f:00.4** also uses **snd_hda_intel**, so blacklisting that module would take out more than the GPU audio. The configuration instead binds only the two listed vendor/device IDs to **vfio-pci**. Do not bind every device from vendor **1002**, and do not blacklist **amdgpu** or **snd_hda_intel** globally.

The host CPU type, because this VM cannot move hosts

FULL CPU FEATURES • NO LIVE MIGRATION EITHER WAY

The `host` CPU type gives the guest the host CPU's features at the cost of portability between differing hosts. A running VM holding a raw local PCI device cannot be live-migrated anyway, and the raw address ties the configuration to this node, so the trade-off costs nothing here.

The same setting also sizes OVMF's 64-bit PCI MMIO window from the host CPU, which Proxmox names as the first fix when a large-VRAM GPU reports Code 12 in the guest, and it is what puts the host processor in front of Windows 11's requirements check in step 01. In a cluster that standardizes passthrough, define a mapped device under **Datacenter > Resource Mappings > PCI Devices** instead of using a raw address, and re-test the GPU if the CPU type changes with it.

01 Approvals, capacity, and the recovery plan

- Confirm the change is approved by the host, VM, application, and data owners, current backups exist, and `pve01` answers on both the web interface and SSH from another system.
- Connect the VM's monitor to the Radeon card itself rather than to a motherboard video connector. Where you are also keeping the integrated-graphics console described above, that console needs its own screen or a second input on the same one. Plan keyboard and mouse input as well: remote access, per-device USB passthrough, or a separately assessed USB controller.
- Use a supported Windows 11 x64 ISO and the official stable VirtIO driver ISO. The Windows license has to cover a virtual machine: Microsoft's terms allow one instance on one device, physical or virtual, and a license acquired preinstalled on a device transfers only with that device, so the copy that came with the host does not cover this VM. Read the terms that came with the license you plan to use.
- Check the host CPU against Microsoft's Windows 11 processor requirements for the release being installed, since CPU type `host` is what puts that processor in front of the guest. Then confirm the planned VM resources fit the node:

```
lscpu | grep -E '^(Model name|CPU\(s\)|Socket\(s\)|Core\(s\) per socket):'  
free -h
```

The example profile uses six vCPUs and 16 GiB of fixed RAM. Those are deployment choices, and the right numbers come from node capacity and the workload. Schedule a host reboot window, and keep physical recovery access available in case network management does not return.

02 Prove the host is ready

Run everything from **Shell** on the node or over SSH. First the release, then the IOMMU (the input/output memory management unit that isolates device memory access, called AMD-Vi on this platform):

```
pveversion --verbose  
uname -r  
dmesg | grep -Ei 'AMD-Vi|IOMMU|Interrupt remapping'
```

The validated host printed `AMD-Vi: Using global IVHD`, `iommu: Default domain type: Translated`, and `AMD-Vi: Interrupt remapping enabled`. IOMMU is enabled by default on AMD hosts, so no `amd_iommu=on` parameter is needed, and `iommu=pt` is an optional performance mode rather than part of this baseline. An Intel host prints DMAR messages instead and needs `intel_iommu=on` on kernels before 6.8; the remaining checks apply unchanged.

If IOMMU or interrupt remapping is missing, stop: enable the IOMMU setting in the board's UEFI firmware and re-check. Do not enable unsafe interrupts to work around it. Interrupt remapping is what checks that a device only raises interrupts it was assigned, so switching that check off lets a compromised guest raise interrupts on the host or on another VM.

Set two more options in that firmware whether or not the check above passed, so plan the trip into the board's setup screen either way. Set **Resizable BAR (Smart Access Memory** on AMD boards) to off: Proxmox reports that some AMD

GPUs from Vega onward show Code 43 in a Windows guest while it is on, and a VM cannot use the feature either way. The other is **CSM**; turning it off leaves a BIOS-mode install unbootable, so confirm the boot mode first:

```
efibootmgr -v
```

Proxmox states that a reply saying EFI variables are not supported means the host booted in BIOS/Legacy mode. Where the output instead lists a boot entry, turn **CSM** or **Legacy boot** off, which skips legacy VGA initialization and leaves the card able to initialize again inside the VM; Proxmox says that can help and names single-GPU hosts in particular. On a host that booted in BIOS/Legacy mode, leave CSM alone and carry on: passthrough does not need a UEFI-mode install, and turning CSM off there stops the host booting. Plan a reinstall in UEFI mode only if the card will not initialize in the VM and the troubleshooting below does not recover it.

Then the exact devices, their drivers, and their isolation. On different hardware, discover the card and its audio function first with `lspci -nn | grep -Ei 'vga|display|audio'` and substitute the addresses and group numbers throughout:

```
lspci -nnk -s 0d:00.0
lspci -nnk -s 0d:00.1
readlink /sys/bus/pci/devices/0000:0d:00.0/iommu_group
readlink /sys/bus/pci/devices/0000:0d:00.1/iommu_group
find /sys/kernel/iommu_groups/29/devices -maxdepth 1 -type l -printf '%f\n'
find /sys/kernel/iommu_groups/30/devices -maxdepth 1 -type l -printf '%f\n'
```

- Before binding, `0d:00.0` shows `[1002:747e]` with driver `amdgpu`, and `0d:00.1` shows `[1002:ab30]` with `snd_hda_intel`. A different address or ID means different hardware: stop and reassess rather than copying these values.
- On this host each function has its own group, and some sharing is acceptable: Proxmox states a device may share its IOMMU group with its own functions (a GPU with its HDMI audio device) or with its root port or PCI(e) bridge. Stop if an unrelated device sits in the group. Moving the card to another slot can change grouping, and the ACS override is a documented risk: it presents software-separated groups without real hardware isolation. Apply it only where that risk has been accepted.
- The full inventory is also available with `pvesh get /nodes/pve01/hardware/pci --pci-class-blacklist ""`.

Last, the Secure Boot firmware package:

```
pve_edk2="$(dpkg-query -W -f='${Version}' pve-edk2-firmware-ovmf 2>/dev/null)"
dpkg --compare-versions "$pve_edk2" ge 4.2025.05-1 \
  && echo "ok: $pve_edk2" \
  || echo "check pve-edk2-firmware-ovmf (found: ${pve_edk2:-not installed})"
```

The firmware is in the package `pve-edk2-firmware-ovmf`; the older `pve-edk2-firmware` is a metapackage a Proxmox VE 9 host may not carry, so querying that name can report nothing on a current host. Proxmox states EFI disks created from `4.2025.05-1` or newer carry both the Microsoft 2011 and 2023 Secure Boot certificates and the `ms-cert=2023k` marker. A Proxmox VE 9.2 host is past that already, since `qemu-server` requires `4.2025.05-2` or newer.

03 Upload the two ISOs

Upload both to **local > ISO Images** on the node: the Windows 11 x64 ISO, and the official stable `virtio-win.iso` from `fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso` (the stable build roughly matches what the most recent RHEL release included; the "latest" build can be pre-release). The Windows 10 and 11 drivers on that ISO carry a Microsoft attestation signature rather than a WHQL signature, and the project provides WHQL-signed builds only with a paid Red Hat Enterprise Linux subscription, so check that against any driver-signing rule the

organization applies. Verify the publisher's checksum where provided, and record filenames and download dates in the change record.

04 Create the VM to Windows 11's requirements

Select **Create VM**, use the next free VM ID (the example is named `win11-gpu`), and enable the wizard's **Advanced** view: startup behavior, SSD emulation, backup, and ballooning stay hidden without it. Proxmox keeps that checkbox in the browser, so it stays on for the PCI dialog in step 08. Selecting the Windows 11 guest type makes the wizard default to q35 and OVMF; verify the values rather than assuming.

TAB	SETTING	VALUE
General	Name / Start at boot	<code>win11-gpu</code> ; start at boot disabled until validation is complete
OS	ISO / Type / Version	The Windows 11 ISO; Microsoft Windows, 11/2022/2025 ; additional VirtIO drive with <code>virtio-win.iso</code>
System	Graphic card / Machine / BIOS	Default display during installation; q35 at the version Proxmox pins; OVMF (UEFI)
System	EFI / TPM	EFI disk on <code>vmfast</code> with Pre-Enroll keys enabled; TPM state on <code>vmfast</code> , version v2.0
System	SCSI Controller / Qemu Agent	VirtIO SCSI single ; agent enabled
Disks	SCSI 0 on <code>vmfast</code>	128 GiB baseline; Discard, IO thread, SSD emulation, and Backup enabled; cache and Async IO at their defaults
CPU	Sockets / Cores / Type	1 / 6 baseline / host
Memory	Fixed allocation	16384 MiB baseline; Ballooning Device disabled. The fixed allocation already holds the memory steady, so the one effect is that Proxmox stops reporting how much the guest is using
Network	Bridge / Model	<code>vmbro</code> ; VirtIO (paravirtualized) ; Firewall flag per site policy

A guest vIOMMU is unnecessary for whole-GPU passthrough, so leave it off. The per-NIC **Firewall** flag only makes the interface eligible for Proxmox firewall processing. On LVM-thin, the 128 GiB disk consumes space as Windows writes, and thin-pool data and metadata usage must be monitored. Review every value on the Confirm tab before finishing.

05 Install Windows with the VirtIO drivers

- Open the VM **Console**, start the installer, and choose the custom option, labelled **Custom (advanced)** or **Custom: Install Windows only (advanced)** depending on the media.
- At the disk screen, select **Load driver** and browse the VirtIO CD to `vioscsi\w11\amd64`; select the Red Hat VirtIO SCSI pass-through controller, and the 128 GiB disk appears.
- Select **Load driver** again with `NetKVM\w11\amd64` for the Red Hat VirtIO Ethernet Adapter, then complete setup on the virtual disk.
- After first sign-in, run `virtio-win-guest-tools.exe` from the VirtIO CD as administrator: it installs the remaining VirtIO drivers and the QEMU Guest Agent. Resolve any leftover Device Manager entries from the matching `w11\amd64` folders, and confirm the guest IP appears in the Proxmox Summary after a restart.

Microsoft requires internet connectivity and a Microsoft account during initial setup for Windows 11 Home and for Pro in personal use; enterprise-managed editions follow the organization's deployment process.

06 Prepare the guest and prove the recovery path

- Install current Windows updates, confirm licensing, and give the guest its planned address (DHCP reservation or static).
- Configure remote access and test it from another system. Windows Home cannot act as a Remote Desktop host: RDP into the guest requires Pro, Enterprise, or Education, a separately approved remote-access product, or the physical monitor.
- Check the system volume's encryption state from an elevated terminal: `manage-bde -status C:`. If protection is on, confirm the BitLocker or Device Encryption recovery key is stored outside the VM and reachable by the authorized recovery owner, and record the storage location (never the key itself) in the change record. Encryption can be active without anyone having enabled it by hand.
- Keep the virtual display at **Default**, confirm noVNC still works, and shut Windows down cleanly.

Do not continue until remote access or the physical monitor is a tested recovery path.

07 Reserve the Radeon for VFIO

Back up the files this step touches, as `root` on the node:

```
gpu_backup_dir="/root/gpu-passthrough-backup-$(date +%Y%m%d-%H%M%S)"
install -d -m 0700 "$gpu_backup_dir"
cp -a /etc/modules "$gpu_backup_dir/modules"
cp -a /etc/modprobe.d "$gpu_backup_dir/modprobe.d"
[ ! -e /etc/default/grub ] || cp -a /etc/default/grub "$gpu_backup_dir/grub"
[ ! -e /etc/kernel/cmdline ] || cp -a /etc/kernel/cmdline "$gpu_backup_dir/kernel-cmdline"
printf '%s\n' "$gpu_backup_dir"
```

Add the VFIO modules to `/etc/modules` if absent, one per line: `vfio`, `vfio_iommu_type1`, `vfio_pci`. The old `vfio_virqfd` module went into the vfio core in Linux 6.2, so Proxmox lists three and then tells you to check for four. Then bind exactly the two Radeon functions in a dedicated file, `/etc/modprobe.d/vfio-pve01-radeon.conf`:

```
options vfio-pci ids=1002:747e,1002:ab30
```

• What stays out of the baseline

No global `blacklist amdgpu` or `blacklist snd_hda_intel`, no `allow_unsafe_interrupts=1`, no ACS override, no downloaded ROM file, no `rombar=off`, and no `amd_iommu=on` on this already working AMD host. Each of those is a targeted workaround with its own costs; add one only when a specific failure calls for it. The `ids=` rule matches every device carrying those IDs, so reassess it if a second matching AMD card is ever installed.

Apply and verify:

```
update-initramfs -u -k all
reboot
# after the reboot, over web or SSH:
lsmod | grep -E '^vfio'
lspci -nnk -s 0d:00.0
lspci -nnk -s 0d:00.1
```

The physical display may go dark during Linux startup; that is the expected headless state, so manage over the network. Both functions must now report `Kernel driver in use: vfio-pci`. If either still belongs to `amdgpu` or `snd_hda_intel`, do not attach it; the troubleshooting section has the documented next step.

08 Attach the GPU in two stages

Stage 1, secondary GPU with the console kept. With the VM stopped, select **Hardware > Add > PCI Device** and tick **Advanced** to reveal the **PCI-Express** and **ROM-Bar** options. Choose **Raw Device** and select `0000:0d:00.0`. Enable **All Functions** (the selection shortens to `0d:00` and brings the audio function along) and **PCI-Express**, which is off until you set it. Leave **Primary GPU** disabled and **ROM-Bar** enabled, and set no ID overrides.

Keep the virtual **Display** at **Default**, start the VM, and use noVNC: Windows enumerates the Radeon as an additional display adapter. Windows picks which adapter it draws on while two are attached, so press **Windows+P** in the guest and choose **Duplicate** or **Second screen only** to bring the physical monitor up.

Have the step 06 recovery key to hand first, because changing virtual hardware can send Windows to the recovery screen, and from Stage 2 on that screen appears only on the physical monitor.

Install the AMD driver. Download the current Windows 11 driver from AMD's official driver page; the PCI output is model-ambiguous, so AMD's Auto-Detect and Install tool is the right way to pick the package. Install, restart, and confirm Device Manager shows no error on the Radeon or its HDMI/DP audio device, the attached monitor can display Windows, and remote access still works. Use the AMD-recommended package current at deployment time.

Stage 2, make the Radeon primary. Shut Windows down cleanly, edit the PCI device, keep **All Functions** and **PCI-Express**, and enable **Primary GPU**. Start the VM while watching the physical monitor.

With **Primary GPU** enabled, Proxmox writes `x-vga=1` and ignores the virtual VGA setting, so a blank noVNC console is expected from here on. Setting the Display device to **None** is optional cleanup once physical and remote access are proven.

09 Optional: pass through keyboard and mouse

With remote access, no keyboard or mouse has to be dedicated to the VM. For a physical desktop, stop the VM and use **Hardware > Add > USB Device**, adding keyboard and mouse separately: by **USB Vendor/Device ID** for a unique device, by **USB Port** when a fixed port should always follow the VM, or by **mapped Device** where a Proxmox USB resource mapping exists. Leave **Use USB3** enabled for USB 3-capable devices.

Passing an entire USB controller needs its IOMMU group and every attached device reviewed first: taking the controller that carries the boot media, management adapter, or recovery keyboard can make the node hard to administer.

10 Validate the build and test the lifecycle

On the Proxmox side, read the effective configuration back (replace the ID with the one the wizard assigned):

```
qm config 100
```

Confirm:

- `bios: ovmf` and a pinned q35 machine version.
- CPU `host`, the planned cores, and fixed memory with ballooning `0`.
- `virtio-scsi-single`, with the OS disk on `vmfast` with discard and IO thread.
- The EFI disk carrying `efitype=4m`, `pre-enrolled-keys=1`, and `ms-cert=2023k`.
- TPM v2.0 state.
- VirtIO networking on `vmbr0`, and the agent enabled.
- `hostpci` carrying `0d:00` with PCIe and, after Stage 2, primary-GPU mode.

The markers `ms-cert=2023` and `ms-cert=2023w` may indicate partial enrollment: the VM start task log warns about them, and Proxmox's enrollment procedure should be run for such disks too. Correct mismatches through the Hardware and Options views or documented `qm set` commands rather than editing the VM configuration file directly.

Inside Windows, from an elevated PowerShell session:

```
Confirm-SecureBootUEFI
Get-Tpm
tpmtool getdeviceinformation
manage-bde -status C:
Get-PnpDevice -Class Display | Format-Table Status, FriendlyName, InstanceId -AutoSize
Get-CimInstance Win32_VideoController | Select-Object Name, DriverVersion
```

- `Confirm-SecureBootUEFI` returns `True`; `Get-Tpm` reports present and ready; `tpmtool` reports version 2.0; `msinfo32` shows **BIOS Mode: UEFI** and **Secure Boot State: On**.
- The Radeon reports OK in Device Manager with an AMD driver, the attached monitor displays Windows, the HDMI/DP audio device is present where GPU audio is needed, and the guest IP shows in Proxmox through the agent.
- The `manage-bde` output has been reviewed, and any recovery key is confirmed stored outside the VM.

• **The green checkmark does not confirm the certificate updates**

Open **Windows Security > Device security > Secure Boot**. Sign off only when the status reads "Secure Boot is on and all required certificate updates have been applied. No further certificate changes are needed." Microsoft states a green checkmark alone does not confirm the 2023 certificate updates; if the text is absent, install all updates, restart, and follow Microsoft's current Secure Boot certificate guidance. Microsoft also states the status text is visible on all supported Windows versions, while the badge changes and app notifications are off by default on managed and enterprise devices, so on a managed VM read the text and disregard the badge.

Lifecycle tests, before any autostart:

- Restart Windows from inside the guest.
- Shut down and start the VM again.
- Stop and start it once from Proxmox after a clean shutdown.
- Confirm GPU display and driver health after every cycle.
- Run a manual backup to `backup-local` and confirm it completes.
- Schedule a restore test into an isolated VM.

Do not assign the same physical GPU to the original and the restored VM at once. A GPU can pass through cleanly on the first boot and still fail to reset for the next start; these cycles catch that. Only then consider **Start at boot**, with a startup order and delay that give networking and storage time to come up.

LIMITS **Operational limits to hand the owner**

TOPIC	GUIDANCE
Exclusivity and console	The Radeon serves one VM; the host stays headless and is managed over web and SSH; noVNC and SPICE cannot show the card's output.
GPU reset	If the device does not recover after a guest stop, collect logs and follow the recovery sequence in the troubleshooting section: a controlled host reboot, then a clean shutdown and a cold power cycle. No unsupported reset module in the baseline.

TOPIC	GUIDANCE
Power states	Prefer clean shutdowns. Proxmox refuses Hibernate on a VM holding a passed-through PCI device, because the device cannot be migrated. Disabling Windows Fast Startup is a reasonable choice for a dedicated passthrough desktop, and leave sleep and hibernation inside the guest unused until they are tested with this exact GPU, firmware, kernel, and driver mix.
Mobility	No live migration with a raw local device, and the raw address <code>0d:00</code> ties the config to this host and can change if the card moves slots. Clusters use Resource Mappings.
Snapshots and backups	Proxmox backs up disks, configuration, EFI state, and TPM state. A backup does not include the physical GPU, so restoring elsewhere needs compatible hardware and a reassigned device. A snapshot records this VM's disks and configuration on this node, so it is a way back from a change rather than a copy that survives losing the node. Leave Include RAM clear: a snapshot that includes memory is refused for the same reason Hibernate is, though the dialog recommends it for a running VM. <code>backup-local</code> also sits inside the same server, so keep an additional approved copy.
TPM and disk encryption	The TPM is emulated and its state is a volume beside the VM disk, so anyone who can read that storage or a backup holds the encrypted disk and the state that unlocks it. Proxmox states an emulated TPM gives no real security benefit over a physical one. Where BitLocker uses a TPM-only protector, protect the VM's storage and its backups to the level the data itself requires.
BitLocker and Secure Boot	Changes to TPM, EFI variables, Secure Boot certificates, or virtual hardware can trigger BitLocker recovery, so do not delete and recreate the TPM or EFI disk unless the change is planned. For an older VM, follow the Proxmox certificate-enrollment procedure and suspend BitLocker protectors first; do not re-enroll keys unnecessarily.
Updates	Keep a known-good kernel in the boot menu and retest passthrough after material Proxmox kernel, board firmware, GPU firmware, or AMD driver updates.
Other host devices	Motherboard audio at <code>0f:00.4</code> keeps <code>snd_hda_intel</code> ; the VFIO rule must be reassessed if another device with the same IDs arrives; and some licensing, anti-cheat, DRM, or attestation systems may detect or reject a VM, so test the applications themselves.

HELP Troubleshooting

- **The host display is blank after the reboot**

Expected, when web and SSH still answer: connect to the management address and finish the VFIO verification. A blank monitor alone does not mean the host failed to boot. If the network is also gone, the display comes back only through hardware the VFIO binding does not reach: a second display adapter, the CPU's integrated graphics where the board can switch to it, a management controller such as a BMC, or moving the boot disk to another machine. Booting an older kernel does not help, because step 07 writes the binding into every kernel's initramfs.

- **A function did not bind to vfio-pci**

Confirm the IDs in the modprobe file match `lspci -nn`, then inspect `dmesg` for vfio and amdgpu messages. When the IDs are right but the host driver loads first, Proxmox documents a soft dependency as the next step: create `/etc/modprobe.d/vfio-softdep.conf` with `softdep amdgpu pre: vfio-pci` and `softdep snd_hda_intel pre: vfio-pci`, then `update-initramfs -u -k all` and a reboot. The softdep only orders loading; it blacklists nothing. Remove it again if the plain binding works without it.

If `dmesg` instead reports vfio-pci unable to reserve a memory BAR, the host's boot framebuffer is holding the range. Proxmox documents `video=efifb:off` for that, and it does nothing on Proxmox VE 9: that kernel is built with `CONFIG_FB_EFI` off, so no efifb driver exists, and the boot framebuffer is a simple-framebuffer device that

simplifiedrm drives. Use `initcall_blacklist=sysfb_init`, which stops the kernel creating it. Try the plain binding first: since Linux 6.0 vfio-pci evicts a conflicting boot framebuffer when it claims a VGA device.

- **The Windows installer sees no disk**

Confirm the VirtIO ISO is attached, the controller is **VirtIO SCSI single**, the disk is SCSI 0, and the driver path is `vioscsi\w11\amd64` from the current stable ISO. Load the supported driver rather than switching the disk to IDE to bypass it.

- **Windows has no network**

Load `NetKVM\w11\amd64` or run the guest-tools installer, then confirm the NIC is VirtIO on `vmbr0`, not marked disconnected, and permitted by the applicable firewall rules.

- **The GPU is present but no physical display appears**

Press **Windows+P** in the guest and choose **Duplicate** or **Second screen only** before treating a dark monitor as a passthrough fault, since Windows chooses which adapter it draws on. Then return to Stage 1 (Primary GPU off, Display Default) and confirm both functions are attached through **All Functions** with **PCI-Express** and **ROM-Bar** on, the current AMD driver is error-free in Device Manager, and the monitor and cable are on the passed-through card (test another port). Confirm Resizable BAR is off in the host firmware, which Proxmox ties to Code 43 on AMD cards from Vega onward.

When output fails before Windows loads, check whether the card's ROM is UEFI capable: look the card up in the TechPowerUp VBIOS database, or dump the ROM and run rom-parser, where a UEFI-capable ROM reports a type 3 entry. Add a downloaded ROM file or disable ROM-Bar only on evidence the device requires it.

- **Works once, fails on the next start**

Shut the guest down cleanly and reboot the host once; if the device still does not recover, power the host off cleanly and cold power cycle it. Record the kernel, firmware, and driver versions and which action recovered it, then investigate as a device-reset issue, which Proxmox calls the AMD reset bug and works around with an out-of-tree vendor-reset patch outside this baseline. Proxmox names Navi 5xxx and 6xxx as affected and not this card's generation, so read that as a starting point rather than a verdict. Keep the VM off autostart until repeated cycles succeed.

ROLLBACK Two ways to roll back

Restore the virtual console without releasing the GPU. Stop the VM, clear **Primary GPU** (or remove the PCI device temporarily), set Display back to **Default**, and start the VM on noVNC. The card can stay reserved by VFIO while Windows is repaired.

Return the Radeon to the host. Stop the VM first and remove the PCI device from its hardware: Proxmox binds a passed-through device to vfio-pci when the VM starts, so a VM that still holds the card takes it back from the host. Clearing **Start at boot** instead leaves the card free only until somebody starts the VM. Then, over SSH, disable the ID rule without deleting it, rebuild, and reboot:

```
mv /etc/modprobe.d/vfio-pve01-radeon.conf /etc/modprobe.d/vfio-pve01-radeon.conf.disabled
[ ! -e /etc/modprobe.d/vfio-softdep.conf ] ||
mv /etc/modprobe.d/vfio-softdep.conf /etc/modprobe.d/vfio-softdep.conf.disabled
update-initramfs -u -k all
reboot
```

Afterwards `lspci -nnk` shows `amdgpu` and `snd_hda_intel` back on their functions. The VFIO lines in `/etc/modules` reserve nothing without a matching `ids=` rule and can wait for a controlled cleanup. If any edited file is in doubt, restore the copies from the step 07 backup directory, rebuild the initramfs, and reboot.

DONE What you achieved

- A Windows 11 VM that meets the firmware requirements: OVMF with pre-enrolled Secure Boot keys and current certificates, TPM 2.0, q35, VirtIO storage and networking, and the guest agent reporting.
- The Radeon and its audio function bound to VFIO by exact ID, with the motherboard audio untouched and no other device on this host matching those IDs.
- A cutover with a proven way back at every step: recovery access was tested before the GPU moved, and both rollback options are documented above.
- The build passed restart, stop/start, backup, and Secure Boot certificate checks, and its limits (reset behavior, no live migration, BitLocker recovery triggers, VM-sensitive applications) are recorded for the owner, along with versions, IDs, and the recovery-key location in the change record.

REF References

Proxmox VE Administration Guide: PCI(e) Passthrough, OVMF, Secure Boot certificates, TPM, machine version, memory · pve.proxmox.com/pve-docs/chapter-qm.html · which bootloader is used · .../chapter-sysadmin.html

Proxmox VE wiki: PCI Passthrough, for BIOS options, Code 43, Code 12, `video=efifb:off`, rom-parser and the AMD reset bug · pve.proxmox.com/wiki/PCI_Passthrough · Windows 11 guest best practices · .../wiki/Windows_11_guest_best_practices

virtio-win project: stable ISO downloads and driver signatures · github.com/virtio-win/virtio-win-pkg-scripts

Microsoft: Windows 11 system requirements · support.microsoft.com/windows/experience/compatibility/windows-11-system-requirements · Secure Boot certificate update status in the Windows Security app · support.microsoft.com/servicing/os/secure-boot/2026/03/secure-boot-certificate-update-status-in-the-windows-security-app

Microsoft: enable Remote Desktop on your PC, for the editions that can host it · learn.microsoft.com/windows-server/remote/remote-desktop-services/remotepc/remote-desktop-allow-access · Windows Operating System license terms, sections 2(a), 2(b), 4(a) · microsoft.com/en-us/useterms

AMD: Drivers and Support for Radeon graphics · amd.com/en/support