

Comprehensive Guide: Configuring Proxmox for GPU Passthrough with Windows 10

Proxmox Virtual Environment (Proxmox VE) offers robust virtualization capabilities, including GPU passthrough, which allows users to dedicate a physical GPU to a virtual machine. This guide provides a comprehensive walkthrough to configure Proxmox for GPU passthrough, ensuring seamless integration with Windows 10 virtual machines.

Prerequisites

- Installed Proxmox VE with access to the WebGUI and server node's Shell terminal.

Step 1: Configuring the Grub

Ensure compatibility with GPU passthrough and ZFS by modifying the Grub configuration:

- For Intel CPUs:

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

- For AMD CPUs:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet amd_iommu=on root=ZFS=rpool/ROOT/pve-1 boot=zfs"
```

Step 2: VFIO Modules

Load necessary VFIO modules to facilitate device assignment to virtual machines:

- Edit `/etc/modules` and add:

```
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
```

Step 3: IOMMU Interrupt Remapping

Configure IOMMU interrupt remapping for enhanced system stability and security:

- Run the following commands in the shell:

```
echo "options vfio_iommu_type1 allow_unsafe_interrupts=1" >
/etc/modprobe.d/iommu_unsafe_interrupts.conf
echo "options kvm ignore_msrs=1" > /etc/modprobe.d/kvm.conf
```

Step 4: Blacklisting Drivers

Prevent the host system from utilizing the GPU by blacklisting relevant drivers:

- Add the following lines to `/etc/modprobe.d/blacklist.conf`:

```
blacklist radeon
blacklist nouveau
blacklist nvidia
```

Step 5: Adding GPU to VFIO

Identify and add the GPU to the VFIO driver for exclusive assignment:

- Edit `/etc/modprobe.d/vfio.conf` and add:

```
options vfio-pci ids=<vendorID1>,<vendorID2> disable_vga=1
```

ZFS Integration

Ensure ZFS compatibility by editing `/etc/kernel/cmdline`:

- For Intel CPUs:

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

- For AMD CPUs:

```
root=ZFS=rpool/ROOT/pve-1 boot=zfs quiet amd_iommu=on iommu=pt
```

Save and refresh changes with `pve-efiboot-tool refresh`.

Configuring the VM (Windows 10)

Setting up a virtual machine to utilize the GPU involves several steps:

Step 1: Create a VM

1. In the Proxmox web interface, navigate to the "Create VM" wizard.
2. Specify necessary parameters like CPU cores, memory, and network settings.
3. Ensure to allocate enough resources to meet the requirements of your intended workload.

Step 2: Enable OVMF (UEFI) for the VM

1. In the VM settings, navigate to the "Options" tab.
2. Set the BIOS type to "UEFI (OVMF)".
3. Add an "EFI Disk" under the hardware tab to comply with the UEFI boot requirements.

Step 3: Edit the VM Config File

1. Access the Proxmox host terminal or SSH into the host.
2. Navigate to `/etc/pve/qemu-server/` and locate the configuration file for your VM (named `<vmid>.conf`).
3. Edit the configuration file using your preferred text editor.
4. Add the following lines to the configuration file:

```
machine: q35
cpu: host,hidden=1,flags=+pcid
args: -cpu 'host,+kvm_pv_unhalt,+kvm_pv_eoi,hv_vendor_id=NV43FIX,kvm=off'
note you can also change it to host
```

5. Save and exit the editor.

Step 4: Add PCI Devices (Your GPU) to VM

1. In the Proxmox web interface, navigate to the hardware tab of your VM.
2. Click on "Add" and select "PCI Device" from the dropdown menu.

3. Choose your GPU from the list of available PCI devices.
4. Ensure to check "All Functions", "Rom-Bar", and set "Primary GPU" to "NO".
5. Click "Add" to confirm the selection.

Step 5: Start the VM

1. Once the VM configuration is complete, start the virtual machine.
2. Access the VM console through the Proxmox web interface or using a VNC client.

Final Setup: Installing / Configuring Windows 10

During the Windows installation process:

1. **Install VirtIO Drivers:** Mount the VirtIO drivers ISO to the VM's virtual CD-ROM drive and follow the prompts to install VirtIO drivers during the Windows installation process.
2. **Configure Windows 10:** Once Windows is installed, install any necessary software and drivers for your intended workload, ensuring that the GPU passthrough is functioning correctly.

By following these comprehensive steps, users can effectively configure Proxmox for GPU passthrough, enabling the creation of powerful virtualized environments with dedicated GPU resources. Remember to thoroughly test configurations and ensure compatibility with your hardware setup before deploying production workloads.

Revision #1

Created 17 March 2024 16:55:09 by Admin

Updated 22 March 2024 18:32:26 by Admin