Proxmox

Tuto concernant l'installation de proxmox.

- Installation proxmox
- GPU passthrough

Installation proxmox

Lien utile pour proxmox

Suppression volumes local

On supprime d'abord le** volume local** sur le tableau de bord proxmox et on supprime le volumes local proxmox par le terminal et on redimentionne **le volume root**.

```
lvremove pve/data \
&& lvextend -L +104g /dev/pve/root \
&& resize2fs /dev/mapper/pve-root && lvs
```

Création des volumes avec lvm2

On crée ensuite notre répertoire local dans le terminal `mkdir -p /datastores` on créee notre partion à l'aide de la commande `fdisk /dev/sdb` entrer **n** pour créer une nouvelle partition et **w** pour enregistrer la partition.

On crée nos volumes logiques, on le formate et on le monte directement au dossier **/datastores**:

```
mkdir -p /datastores \
&& vgcreate ares-ssd-vg /dev/sdd4 \
&& lvcreate -n ares-ssd-lv -l 100%free ares-ssd-vg && lvs \
&& mkfs.ext4 /dev/ares-ssd-vg/ares-ssd-lv \
&& mount /dev/ares-ssd-vg/ares-ssd-lv /datastores \
&& echo "" >> /etc/fstab && cat /etc/fstab \
&& echo "# Mount SSD Store" >> /etc/fstab && cat /etc/fstab \
&& echo "/dev/ares-ssd-vg/ares-ssd-lv /datastores□ext4□defaults,discard□0 2" >> /etc/fstab && c
/etc/fstab \
&& df -h
```

ensuite on crée simplement nos volumes logiques:

```
# Création de partion pci & nvme
vgcreate ares-pci-nvme-stores-vg /dev/sdd1 /dev/nvme0n1p1 \
&& lvcreate -c 128k -l 100%FREE --thinpool ares-pci-nvme-stores-lv ares-pci-nvme-stores-vg &&
lvs

# Création de partion: fdisk /dev/sda & fdisk /dev/sdc
vgcreate ares-hdd-stores-vg /dev/sdb1 /dev/sdc1 \
&& lvcreate -c 128k -l 100%FREE --thinpool ares-hdd-stores-lv ares-hdd-stores-vg && lvs
```

On execute les scripts post install avant de lancer les mises à jours:

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/post-install-v3.sh)" \
&& apt update && apt upgrade -y && apt dist-upgrade -y && apt autoremove -y && apt purge &&
apt autoclean \
&& reboot
```

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/kernel-clean-v3.sh)" \
&& bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/update-lxcs.sh)" \
&& bash <(curl -s
https://raw.githubusercontent.com/Weilbyte/PVEDiscordDark/master/PVEDiscordDark.sh ) install
&& reboot
```

```
apt update && apt install -y neofetch htop nmon curl vim wget pve-headers pkg-config libglvnd-
dev gcc make build-essential && apt autoremove -y && apt purge && apt autoclean && reset
```

GPU passthrough

> Ici on va configurer une carte graphique qui sera utilisable par une **VM**.
> [Lien utile]

On Modifie le grub

`vim /etc/default/grub`

```
# on ajoute cette ligne amd proc: amd_iommu=on intel proc: intel_iommu=on
GRUB_CMDLINE_LINUX_DEFAULT="quiet amd_iommu=on iommu=pt
pcie_acs_override=downstream,multifunction nofb nomodeset video=vesafb:off,efifb:off"
```

Mise à jours du grub et verif virtualisation actif

```
update-grub && proxmox-boot-tool refresh \
&& dmesg | grep -e DMAR -e IOMMU
```

Ajout des modules

```
echo "nvidia" >> /etc/modules \
&& echo "nvidia_uvm" >> /etc/modules \
&& echo "vfio" >> /etc/modules \
&& echo "vfio_iommu_type1" >> /etc/modules \
&& echo "vfio_pci" >> /etc/modules \
&& echo "vfio_virqfd" >> /etc/modules && cat /etc/modules \
&& dmesg | grep 'remapping' \
&& 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 && find
/sys/kernel/iommu_groups/ -type l
```

On repère la GPU

```
lspci -v && lspci -nnv | grep VGA
# Résultat
[07:00.0 VGA compatible controller [0300]: NVIDIA Corporation GP108 [GeForce GT 1030]
[10de:1d01] (rev a1) (prog-if 00 [VGA controller])]
```

installation de la rom NVIDIA

```
cd /sys/bus/pci/devices/ \
&&  cd 0000:07:00.0 && echo 1 > rom \
&&  cd /usr/share/kvm/ \
&&  wget https://www.techpowerup.com/vgabios/192498/MSI.GT1030.2048.170413.rom \
&& mv MSI.GT1030.2048.170413.rom GT1030.rom

ls && cd /sys/bus/pci/devices/0000:07:00.0 \
&& chmod 777 rom \
&& cat rom > /usr/share/kvm/GT1030.rom
```

Blacklist des GPU

> “Unable to load the kernel module 'nvidia.ko
>
> Nvidia drivers on Proxmox

```
echo "blacklist vga16fb" >> /etc/modprobe.d/blacklist.conf \
&& echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf \
&& echo "blacklist rivafb" >> /etc/modprobe.d/blacklist.conf \
&& echo "blacklist nvidiafb" >> /etc/modprobe.d/blacklist.conf \
&& echo "blacklist rivatv" >> /etc/modprobe.d/blacklist.conf && cat
/etc/modprobe.d/blacklist.conf
```

```
ls /usr/src/


lcpci -k
ls /etc/modprobe.d/
touch /etc/modprobe.d/nvidia-installer-disable.conf \
&& echo "blacklist nouveau" >> /etc/modprobe.d/nvidia-installer-disable.conf \
&& echo "options nouveau modeset=0" >> /etc/modprobe.d/nvidia-installer-disable.conf \
&& echo "blacklist vfio-pci" >> /etc/modprobe.d/nvidia-installer-disable.conf \
&& cat /etc/modprobe.d/nvidia-installer-disable.conf
&& mv /etc/modprobe.d/pve-blacklist.conf /etc/modprobe.d/pve-blacklist.conf.disable \
&& update-initramfs -u && proxmox-boot-tool refresh && reboot
```

Téléchargement de driver nvidia

```
wget https://fr.download.nvidia.com/XFree86/Linux-x86_64/515.48.07/NVIDIA-Linux-x86_64-515.48.07.run \
&& chmod -x NVIDIA-Linux-x86_64-515.48.07.run \
&& chmod 777 NVIDIA-Linux-x86_64-515.48.07.run \
ls /sys/bus/pci/devices/
echo "0000:07:00.0" > /sys/bus/pci/devices/0000:07:00.0/driver/unbind \
&& ./NVIDIA-Linux-x86_64-510.73.05.run --kernel-source-path /usr/src/linux-headers-5.15.35-1-pve \
&& proxmox-boot-tool refresh && update-initramfs -u

apt update && apt upgrade -y && apt dist-upgrade -y && apt autoremove -y && apt purge && apt autoclean
```