

# Proxmox VE

- Basis Installation nach <https://www.sysorchestra.com/proxmox-5-on-hetzner-root-server-with-ipv4/>
- Referenz-Doku [https://pve.proxmox.com/wiki/Category:Reference\\_Documentation](https://pve.proxmox.com/wiki/Category:Reference_Documentation)
- Command line tools: <https://pve.proxmox.com/pve-docs/>
- Hetzner proxmox Doku: [https://community.hetzner.com/tutorials/install-and-configure-proxmox\\_ve/de?title=Proxmox\\_VE#netzwerkconfiguration-hostsystem-routed](https://community.hetzner.com/tutorials/install-and-configure-proxmox_ve/de?title=Proxmox_VE#netzwerkconfiguration-hostsystem-routed)
- Hetzner Netzwerk Doku: [https://wiki.hetzner.de/index.php/Netzkonfiguration\\_Debian](https://wiki.hetzner.de/index.php/Netzkonfiguration_Debian)
- Netzwerk mit zwei Routing Tabellen/Default Routes: <http://www.rjsystems.nl/en/2100-adv-routing.php>

## PVE Server Backup

<https://xcruft.com/content/proxmox-config-backups>

## Command Line

**qm** = Management der **kvm** VMs

**pct** = Management der **lxc** Container

Alle VMs (KVM und lxc) auf einmal runterfahren

```
pvesh create /nodes/localhost/stopall
```

VM löschen

```
qm destroy 105
```

Mounten eines Containers auf dem Host

```
pct mount 108 mounted CT 108 in '/var/lib/lxc/108/rootfs'
```

Gemeinsames Guest Share (smb)

```
mount.cifs \\\10.10.10.1\guests /mnt/guests
```

Speicherverbrauch ermitteln

```
du -a /home | sort -n -r | head -n 5 find / -type f -size +100M
```

Container umbenennen

```
pct set <VMID> --hostname <newname>
```

# Container

[https://pve.proxmox.com/wiki/Linux\\_Container](https://pve.proxmox.com/wiki/Linux_Container)

Erreichbare Templates auflisten

```
pveam update  
  
pveam available
```

Runterladen

```
pveam download local debian-10.0-standard_10.0-1_amd64.tar.gz
```

## SMB Mount mit Containern

Geht nur mit **privileged** container. Unter /Your LXC Option/Feature muss die "CIFS capability" aktiviert sein.

# Verzeichnisstruktur

Was	Wo	Anmerkung
kvm VM images	/mnt/data/images, /var/lib/vz/images	
kvm VM configs	/etc/pve/nodes/tokoeka/qemu-server	
LXC images	/var/lib/vz/images	
LXC configs	/etc/pve/nodes/tokoeka/lxc	

## Netplan Beispielconfigs

/etc/netplan/default.yaml für zwei Schnittstellen mit festen IPs, default gateway und routing

```
network:  
  version: 2  
  renderer: networkd  
  ethernets:  
    ens18:  
      dhcp4: no  
      addresses: [ 136.243.85.153/27 ]  
      gateway4: 136.243.85.129
```

```
nameservers:
  addresses: [ 213.133.98.98, 213.133.99.99, 213.133.100.100 ]
ens19:
  dhcp4: no
  addresses: [ 10.10.10.10/24 ]
  nameservers:
    addresses: [ 10.10.10.1 ]
  routes:
    - to: 10.10.10.0/24
      via: 10.10.10.1
      metric: 200
    - to: 0.0.0.0/0
      via: 136.243.85.129
      metric: 100
```

Für eine Schnittstelle (ct, nur internes Netz)

```
network:
  version: 2
  renderer: networkd
  ethernet:
    eth0:
      dhcp4: no
      addresses:
        - 10.10.10.16/24
      gateway4: 10.10.10.1
      nameservers:
        addresses: [ 10.10.10.1 ]
      routes:
        - to: 0.0.0.0/0
          via: 10.10.10.1
          on-link: true
/etc/netplan/01-netcfg.yaml (END)
```

Testen:

```
sudo netplan generate
```

Testen mit automatischem zurücksetzen auf den vorherigen Stand

```
sudo netplan try -timeout 180
```

Anwenden

```
sudo netplan apply
```

## Port 111 rpcbind

Siehe <https://www.taste-of-it.de/debian-rpc-port-111-offen/> Über iptables auf localhost einschränken:

```
iptables -A INPUT -p tcp !-s 127.0.0.1 -dport 111 -j DROP

ip6tables -A INPUT -p tcp ! -s IPv6-Adresse -dport 111 -j DROP
ip6tables -A INPUT -p tcp -s IPv6-Adresse -dport 111 -j ACCEPT

iptables -A INPUT -p udp ! -s 192.168.0.0/24 -dport 111 -j DROP
ip6tables -A INPUT -p udp -s! IPv6-Adresse -dport 111 -j DROP
```

## Datenwiederherstellung aus Dumpfile

Dumpfiles werden im Format \*.lzo oder \*.tar.zst komprimiert abgelegt. Sie enthalten die VM Daten im raw Format.

zst dekomprimieren

```
zstd -d vzdump-lxc-113-2022_05_14-01_30_57.tar.zst
```

danach mit tar xf ...tar auspacken.

```
lzop -x [*.lzo Datei]
```

erzeugt eine unkomprimierte \*.vma Datei. Daraus werden die Partitionen als \*.raw Dateien extrahiert:

```
vma extract [*.vma] /mnt/tmp/extract/
```

Mit file sieht man, was drin ist:

```
file tmp-disk-drive-scsi1.raw
```

```
tmp-disk-drive-scsi1.raw: DOS/MBR boot sector; partition 1 : ID=0xee, start-CHS (0x0,0,2), end-CHS (0x3ff,255,63), startsector 1, 204799999 sectors, extended partition table (last)
```

kpartx erzeugt daraus passende loop devices

```
root@tokoeke /mnt/data/tmp/extract # kpartx tmp-disk-drive-scsi0.raw
```

```
loop1p1 : 0 2048 /dev/loop1 2048
loop1p2 : 0 67102720 /dev/loop1 4096
```

Um diese zu mounten, braucht man den offset bis zur Partition:

```
root@tokoeke /mnt/data/tmp/extract # fdisk -l tmp-disk-drive-scsi0.raw
Disk tmp-disk-drive-scsi0.raw: 32 GiB, 34359738368 bytes, 67108864 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: gpt
Disk identifier: 074AFDD5-B8AF-4EB9-A630-1B6E2136EBE9

Device Start End Sectors Size Type
tmp-disk-drive-scsi0.raw1 2048 4095 2048 1M BIOS boot
tmp-disk-drive-scsi0.raw2 4096 67106815 67102720 32G Linux filesystem
```

Der Offset ist also **4096 x 512 = 2097152**. Der Mountbefehl sieht so aus:

```
mount -o ro,loop,offset=2097152 harddrive.img /mnt/loop
```

Am Ende unmountet man alles und detached die loop devices wieder

```
root@tokoeka /mnt/data/tmp/extract # losetup
NAME          SIZELIMIT OFFSET AUTOCLEAR R0 BACK-FILE
DIO LOG-SEC
/dev/loop1      0         0          0 0 /mnt/data/tmp/extract/tmp-disk-
drive-scsi0.raw 0        512
/dev/loop0      0         0          1 0 /var/lib/vz/images/106/vm-106-
disk-0.raw      0        512

root@tokoeka /mnt/data/tmp/extract # losetup -d /dev/loop1
root@tokoeka /mnt/data/tmp/extract # losetup
NAME          SIZELIMIT OFFSET AUTOCLEAR R0 BACK-FILE
DIO LOG-SEC
/dev/loop0      0         0          1 0 /var/lib/vz/images/106/vm-106-
disk-0.raw      0        512
```

## qcow2 Device mounten

```
modprobe nbd max_part=8
```

```
qemu-nbd -connect=/dev/nbd0 /var/lib/vz/images/100/vm-100-disk-1.qcow2
```

```
fdisk /dev/nbd0 -l
```

```
mount /dev/nbd0p1 /mnt/somepoint/
```

```
umount /mnt/somepoint/
```

```
qemu-nbd -disconnect /dev/nbd0
```

```
rmmod nbd
```

# pct Loop Device mounten

Ablauf manuell

```
losetup /dev/loop22 disk-drive-ide0.raw
partx -v --add /dev/loop22
mount /dev/loop22p1 /mnt/123
```

```
root@tokoeka /mnt/data/images/116 # losetup /dev/loop19 vm-116-disk-2.raw
root@tokoeka /mnt/data/images/116 # partx -v --add /dev/loop19
partition: none, disk: /dev/loop19, lower: 0, upper: 0
/dev/loop19: partition table type 'gpt' detected
range recount: max partno=1, lower=0, upper=0
/dev/loop19: partition #1 added
root@tokoeka /mnt/data/images/116 # losetup -l | grep 116
/dev/loop19          0          0          0  0 /mnt/data/images/116/vm-116-
disk-2.raw  0      512
/dev/loop8           0          0          1  0 /mnt/data/images/116/vm-116-
disk-0.raw  0      512
/dev/loop18          0          0          1  0 /mnt/data/images/116/vm-116-
disk-1.raw  0      512
```

Auflisten aller Loop-Devices

```
losetup -l
```

Devices abhängen, mit -D alle zugehörigen abhängen

```
losetup -d
```

```
losetup -D
```

Parsen der Partitionstabelle und anlegen von loop Einträgen nach Bedarf

```
partx -v --add /dev/loop20
```

Partitionstabelle zeigen

```
root@tokoeka /mnt # partx -s /dev/loop22
NR START      END  SECTORS  SIZE NAME  UUID
 1  2048 204799999 204797952 97.7G      7a11d514-01
```

pct set

```
pct set 116 -mp1 volume=data:116/vm-116-disk-2.raw,mp=/mnt/data2,backup=1
```

Offen: offset? p1 ?

# Firewall

zentrale Konfiguration

```
/etc/pve/firewall/cluster.fw
```

## Wenn die Firewall den Host blockiert:

Aktuell ist in rc.local dieser Befehl gesetzt:

```
#
#!/bin/sh -e
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

pve-firewall stop

exit 0
```

Beim Neustart läuft PVE also immer **ohne** Firewall

## Konvertierungen

<https://stafwag.github.io/blog/blog/2018/07/01/migrate-a-windows-vmware-virtual-machine-to-kvm/>

Split disks in eine Datei umwandeln

```
vmware-vdiskmanager -r leapsrv.vmdk -t 0 /tmp/leapsrv_combined.vmdk
```

Wandeln von vmdk nach qcow2

```
qemu-img convert -f vmdk -O qcow2 leapsrv_combined.vmdk
leapsrv_combined.vmdk.qcow2
```

Andersrum von qcow2 (kvm) nach raw (lxc):

```
qemu-img convert -f qcow2 -O raw vm-109-disk-1.qcow2  
/mnt/data/images/114/vm-114-disk-3.raw
```

# User Management

Gruppe erstellen:

```
pveum groupadd admin -comment "System Administrators"
```

Rechte zuweisen:

```
pveum aclmod / -group admin -role Administrator
```

Benutzer der Gruppe zuweisen:

```
pveum usermod testuser@pve -group admin
```

From:

<https://wiki.netzwissen.de/> - netzwissen.de Wiki

Permanent link:

<https://wiki.netzwissen.de/doku.php?id=proxmox&rev=1652735121>

Last update: **05/03/2024 - 10:52**

