Proxmox PVE

- 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
- 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#netzwerkkonfiguration-hostsystem-routed
- Hetzner Netzwerk Doku: https://wiki.hetzner.de/index.php/Netzkonfiguration_Debian
- Netzwerk mit zwei Routing Tabellen/Default Routes: http://www.rjsystems.nl/en/2100-adv-routing.php
- Helper Scripts: https://community-scripts.github.io/ProxmoxVE/

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	

PVE Server Backup

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

User Management

User in PVE erstellen (entspricht dem shell User). Erst dann ist die Administration per Web GUI möglich.

```
pveum user add [user@pam]
```

User auflisten

```
root@pveroser:~# pveum user list
┌──────────────┬─────────┬──────────────────────┬─────────┬─────────┬──────────
┬─────────┬──────┬──────────┬──────────────┬─────────┐
│ userid       │ comment │ email                │ enable │ expire │ firstname
│ groups │ keys │ lastname │ realm-type │ tokens │
├──────────────┼─────────┼──────────────────────┼─────────┼─────────┼──────────
╪═══════╪══════╪══════════╪══════════════╪═══════╪
╪═══════╪══════╪══════════╪══════════════╪═══════╪
```

```
│ alex@pam     │          │                      │ 1       │       0 │
│         │       │          │ pam         │         │
├──────────────┼──────────┼──────────────────────┼─────────┼─────────┼──────────────
┼─────────┼───────┼──────────┼─────────────┼─────────┤
│ root@pam     │          │ admin@netzwissen.de │ 1       │       0 │
│         │       │          │ pam         │         │
├──────────────┼──────────┼──────────────────────┼─────────┼─────────┼──────────────
┼─────────┼───────┼──────────┼─────────────┼─────────┤
│ thommie@pam  │          │                      │ 1       │       0 │
│         │       │          │ pam         │         │
└──────────────┴──────────┴──────────────────────┴─────────┴─────────┴──────────────
┴─────────┴───────┴──────────┴─────────────┴─────────┘
```

Gruppe erstellen:

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

Gruppen auflisten

```
root@pveroser:~# pveum group list
┌─────────┬──────────────────────┬──────────────────────┐
│ groupid │ comment              │ users                │
├─────────┼──────────────────────┼──────────────────────┤
│ admin   │ System Administrators │ alex@pam,thommie@pam │
└─────────┴──────────────────────┴──────────────────────┘
```

Rechte zuweisen:

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

Benutzer einer Gruppe zuweisen:

```
pveum user modify testuser@pam -group admin
```

User disable

```
pveum user modify root@pam -enable 0
```

Andere Felder modifizieren

```
pveum user modify admin@pam -email admin@netzwissen.de
```

2FA Auth löschen: über einen gleichberechtigten User, dann Löschen von "x" im Feld "Key ID"

Kommandozeile und Container/VMs

qm = Management der **kvm** VMs

pct = Management der **lxc** Container

Alle VMs (KVM und lxc) auf einmal runterfahren

```
pvenode stopall
```

oder

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

Doku: https://www.historiantech.com/increasing-operational-efficiency-in-proxmox-with-pvesh/

VM: löschen

```
qm destroy 105
```

lxc: 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
```

Host: Speicherverbrauch ermitteln

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

lxc: Container umbenennen

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

lxc: Container betreten

```
pct enter <VMID>
```

lxc: Datei senden und empfangen

Datei senden

```
pct push <VMID> <file> <target>
```

Datei empfangen

```
pct pull <vmid> <path> <destination> [OPTIONS]
```

QM: Bei VMs geht das per scp. Dazu braucht man die IP der VM

VM: IP Adresse feststellen

```
qm agent 113 network-get-interface
```

Backups manuell

```
vzdump 102 118 122 --compress zstd --mode stop --prune-backups 'keep-last=2'
--mailnotification failure --mailto admin@netzwissen.de --quiet 1 --storage
storagebox_191707
```

Host: PVE Server

Auto-Start der VMs temporär abschalten

```
systemctl disable pve-guests.service.
```

Wieder einschalten

```
 systemctl enable pve-guests.service && /usr/bin/pvesh --nooutput create
```

```
/nodes/localhost/startall.
```

PVE Templates

Erreichbare Templates auflisten

```
pveam update

pveam available
```

Runterladen

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

PVE Firewall

zentrale Konfiguration

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

Ein/aus auf der command line:

```
 pve-firewall stop

 pve-firewall start
```

Host: Firewall blockiert

Wenn die Firewall den Host blockiert: Mit diesem Skript in rc.local wird die FW beim Neustart immer ausgeschaltet:

```
#
#!/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
```

Meckermails von CERT-BUND wegen RPC Port

Der Port 111 ist normalerweise offen, wird aber nicht gebraucht, Zitat Proxmox Staff

*regarding port 111 - it should work to just remove `rpcbind, nfs-common` if you don't need it We might consider doing so in a future version, but since it's expected to deploy PMG behind a firewall (or configure iptables/nft on it) it's not really high priori*ty

Service abschalten mit

```
root@kakariki ~ # systemctl stop rpcbind
Warning: Stopping rpcbind.service, but it can still be activated by:
  rpcbind.socket
root@kakariki ~ # systemctl stop rpcbind.socket
root@kakariki ~ # systemctl stop rpcbind
root@kakariki ~ # systemctl disable rpcbind
Synchronizing state of rpcbind.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable rpcbind
Removed "/etc/systemd/system/multi-user.target.wants/rpcbind.service".
Removed "/etc/systemd/system/sockets.target.wants/rpcbind.socket".
```

SMB Mount mit Containern

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

Netzwerk: Netplan Beispiel-Configs

/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
  ethernets:
    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)
```

Config testen mit "sudo netplan generate"

Testen mit automatischem zurücksetzen auf den vorherigen Stand: "sudo netplan try -timeout 180"

Netplan anwenden "sudo netplan apply"

LXC Container

https://pve.proxmox.com/wiki/Linux_Container

Debug Modus beim Start

pct start 110 –debug

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
```

LXC: raw Device mounten

Welche loop devices werden aktuell benutzt

```
losetup -l

NAME   SIZELIMIT OFFSET AUTOCLEAR RO BACK-FILE
DIO LOG-SEC
/dev/loop1         0      0         1  0 /mnt/data/images/108/vm-108-
disk-1.raw   0     512
/dev/loop27        0      0         1  0 /mnt/data/images/112/vm-112-
disk-0.raw   0     512
/dev/loop17        0      0         1  0 /mnt/data/images/111/vm-111-
disk-2.raw   0     512
/dev/loop8         0      0         1  0 /mnt/data/images/110/vm-110-
disk-0.raw   0     512
/dev/loop25        0      0         1  0 /mnt/data/images/125/vm-125-
disk-1.raw   0     512
/dev/loop6         0      0         1  0 /mnt/data/images/100/vm-100-
disk-0.raw   0     512
/dev/loop23        0      0         1  0 /mnt/data/images/122/vm-122-
disk-1.raw   0     512
/dev/loop13        0      0         1  0 /mnt/data/images/113/vm-113-
disk-0.raw   0     512
```

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
```

Datensicherung: Wiederherstellung 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@tokoeka /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@tokoeka /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 RO 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 RO BACK-FILE
DIO LOG-SEC
/dev/loop0         0      0         1  0 /var/lib/vz/images/106/vm-106-
disk-0.raw   0     512
```

KVM VMs

qcow2 Device mounten

```
modprobe nbd max_part=8

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

Paritionierung ermitteln und mounten

```
fdisk /dev/nbd0 -l

root@tokoeka /mnt/data/images/101 # fdisk /dev/nbd0 -l
Disk /dev/nbd0: 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: 9D35B183-C931-43A4-88DD-659ED1FAA7EE

Device      Start      End  Sectors Size Type
/dev/nbd0p1  2048     4095     2048   1M BIOS boot
/dev/nbd0p2  4096 67106815 67102720  32G Linux filesystem
```

Danach mounten

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

Aufräumen

```
umount /mnt/somepoint/

qemu-nbd --disconnect /dev/nbd0

rmmod nbd
```

KVM: Platte einer VM vergrössern

https://tech.surveypoint.com/tips/resize-kvm-guest-virtual-disk-partition/

Daten-Konvertierungen

https://stafwag.github.io/blog/blog/2018/07/01/migrate-a-windows-vmware-vrtual-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
```

SPICE

Doku: https://pve.proxmox.com/wiki/SPICE#Requirements_for_SPICE Hilfs-Skript in /etc/scripts/spice.sh

```
Usage: ./spice.sh [-u <string>] [-p <string>] vmid [node [proxy]]

-u username. Default root@pam
-p password. Default ''

vmid: id for VM
node: Proxmox cluster node name
proxy: DNS or IP (use <node> as default)
```

Client: virt-viewer, Remmina

ZFS Installation

ZFS installieren

apt install linux-headers-amd64 zfsutils-linux zfs-dkms zfs-zed

https://openzfs.github.io/openzfs-docs/Getting%20Started/Debian/Debian%20Bullseye%20Root%20on%20ZFS.html](https://openzfs.github.io/openzfs-docs/Getting%20Started/Debian/Debian%20Bullseye%20Root%20on%20ZFS.html)

Partitionieren

root@kakariki /etc/apt # fdisk /dev/disk/by-id/nvme-eui.0025388511c55959

(Achtung: gdisk konvertiert MBR nach GPT)

/dev/disk/by-id/nvme-eui.0025388511c55959 /dev/disk/by-id/nvme-eui.0025388511c5595b

DISK1=/dev/disk/by-id/nvme-eui.0025388511c55959-part7 DISK2=/dev/disk/by-id/nvme-eui.0025388511c5595b-part7

Dannach zpool anlegen. "mirror" entspricht RAID1

```
zpool create [-dfn] [-m mountpoint] [-o property=value]… [-o
feature@feature=value]
           [-o compatibility=off|legacy|file[,file]…] [-O file-system-
property=value]… [-R root] [-t tname] pool vdev…
```

```
zpool create \
    -o ashift=12 \
    -o autotrim=on -d \
    -o cachefile=/etc/zfs/zpool.cache \
    -o feature@async_destroy=enabled \
    -o feature@bookmarks=enabled \
    -o feature@embedded_data=enabled \
    -o feature@empty_bpobj=enabled \
    -o feature@enabled_txg=enabled \
    -o feature@extensible_dataset=enabled \
    -o feature@filesystem_limits=enabled \
    -o feature@hole_birth=enabled \
    -o feature@large_blocks=enabled \
    -o feature@livelist=enabled \
    -o feature@lz4_compress=enabled \
    -o feature@spacemap_histogram=enabled \
    -o feature@zpool_checkpoint=enabled \
    -O devices=off \
```

```
    -O acltype=posixacl -O xattr=sa \
    -O compression=lz4 \
    -O normalization=formD \
    -O relatime=on \
    -O canmount=off -O mountpoint=/ -R /mnt \
    rpool mirror \
    /dev/disk/by-id/nvme-eui.0025388511c55959-part7 \
   /dev/disk/by-id/nvme-eui.0025388511c5595b-part7
```

zfs Datasets erstellen

zfs create rpool/mirror

Pool und datasets wieder löschen

zpool destroy -f rpool

Host: LE Zertifikate für PVE

```
pvenode acme account register default admin@netzwissen.de

pvenode config set --acme domains=kakariki.netzwissen.de

root@kakariki /etc/pve # pvenode acme cert order

Loading ACME account details Placing ACME order Order URL:
<https://acme-v02.api.letsencrypt.org/acme/order/1232182246/198286068416>

Getting authorization details from
'<https://acme-v02.api.letsencrypt.org/acme/authz-v3/250346582026>' The
validation for kakariki.netzwissen.de is pending! Setting up webserver
Triggering validation Sleeping for 5 seconds Status is 'valid', domain
'kakariki.netzwissen.de' OK!

All domains validated!

Creating CSR Checking order status Order is ready, finalizing order valid!

Downloading certificate Setting pveproxy certificate and key Restarting
pveproxy Task OK
```

ACME DNS validation Hetzner DNS API

export HETZNER_Token="<token>"

```
./acme.sh --issue --dns dns_hetzner -d example.com -d *.example.com
```

From:
https://wiki.netzwissen.de/ - **netzwissen.de Wiki**

Permanent link:
https://wiki.netzwissen.de/doku.php?id=proxmox

Last update: **16/05/2025 - 15:37**