Proxmox

Nützliches zur Virtualisierungslösung Proxmox

- LXC Container umziehen
- qcow2 Images vergrößern / verkleinern
- Hetzner - IPs auf virtuelle Maschinen via opnSense routen

LXC Container umziehen

Einfaches Backup-Script um Container von einem Storage auf ein anderes umzuziehen

(C) by James Coyle

```
#!/bin/bash
#
# Filename : migrate
# Description : Migrate Proxmox OpenVZ container from one storage to another
# Author : James Coyle
#
# Version:
# -Date       -Author      -Description
# 20-11-2013  James Coyle  Initial
# 13-12-2017  James Coyle  Changes for LXC
#
#

# Variables
TMP=/tmp      #Location to use to create the backup for transferring to new storage. This
needs to be big enough to store the backup archive for the container.

# Do not edit
usage() {
echo "Usage: $0"
echo "          [-c Required: Container ID to migrate <int>] "
echo "          [-s Required: Target storage ID <string>]"
echo "          [-d Optional: Delete the backup file after CT restoration <boolean>]"
echo ""
echo "Example: $0 -c 100 -s nasarray"
echo ""
exit 1;
}
```

```
while getopts "c:s:d" o; do
  case "${o}" in
    c)
      CT=${OPTARG}
      ;;
    s)
      TARGET_STORAGE=${OPTARG}
      ;;
    d)
      DELETE=true
      ;;
    *)
      usage
      ;;
    esac
done
shift $((OPTIND-1))

# Check mandatory fields
if [ -z "${CT}" ] || [ -z "${TARGET_STORAGE}" ]; then
  usage
fi

RUNNING=false

set -e
set -o pipefail

echo "Moving $CT to $TARGET_STORAGE..."
if pct list| fgrep -w -q "$CT" | grep "running"
then
    RUNNING=true
fi

if $RUNNING
then
    pct stop $CT
fi

vzdump --dumpdir $TMP $CT
```

```
ARCHIVE=$(ls -t $TMP/vzdump-lxc-$CT-*.tar | head -n 1)

pct restore $CT $ARCHIVE -force -storage $TARGET_STORAGE

if $RUNNING
then
    pct start $CT
fi

if $DELETE
then
    LOG=$(ls -t $TMP/vzdump-lxc-$CT-*.log | head -n 1)
    echo "Deleting $LOG and $ARCHIVE"
    rm -rf $ARCHIVE $TMP/$LOG
fi
```

qcow2 Images vergrößern / verkleinern

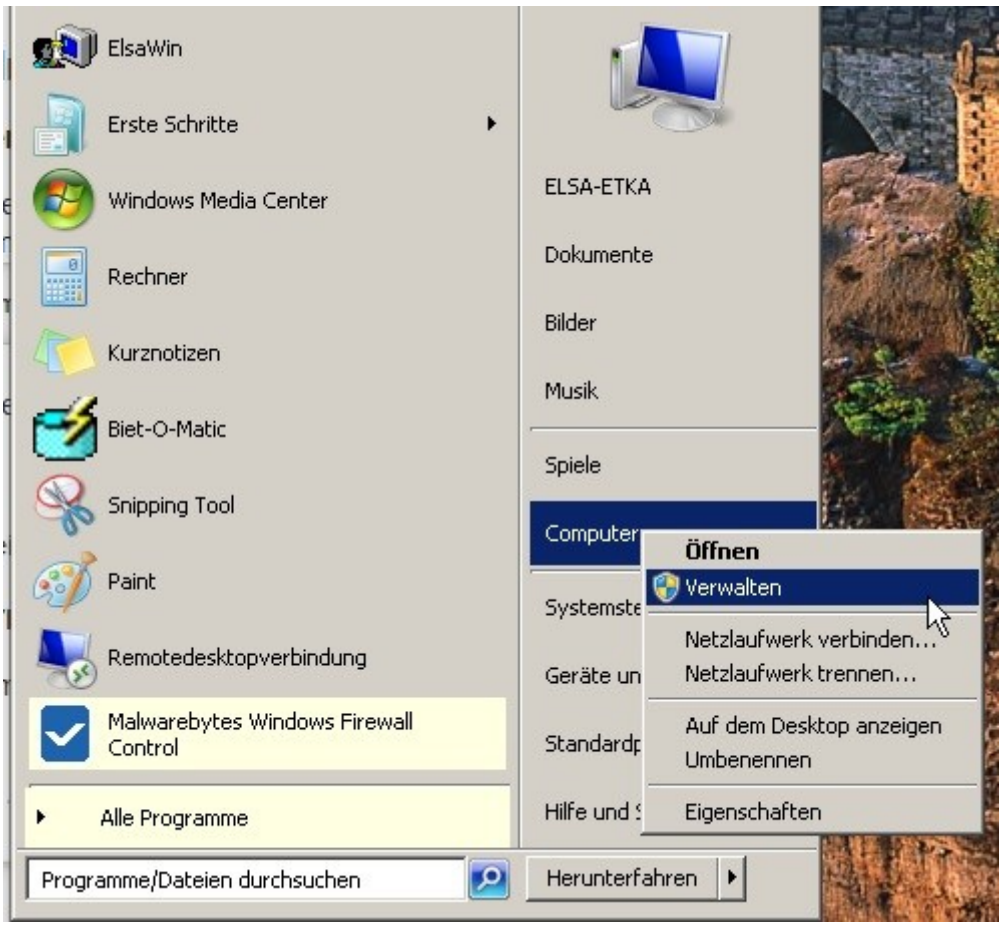
Images vergrößern

1. VM herunterfahren
2. folgenden Befehl eingeben:
   ```
   qemu-img resize image.qcow2 +SIZE
   ```
3. Ab hier unterscheiden die die Vorgehensweisen bei Linux- und Windows VMs

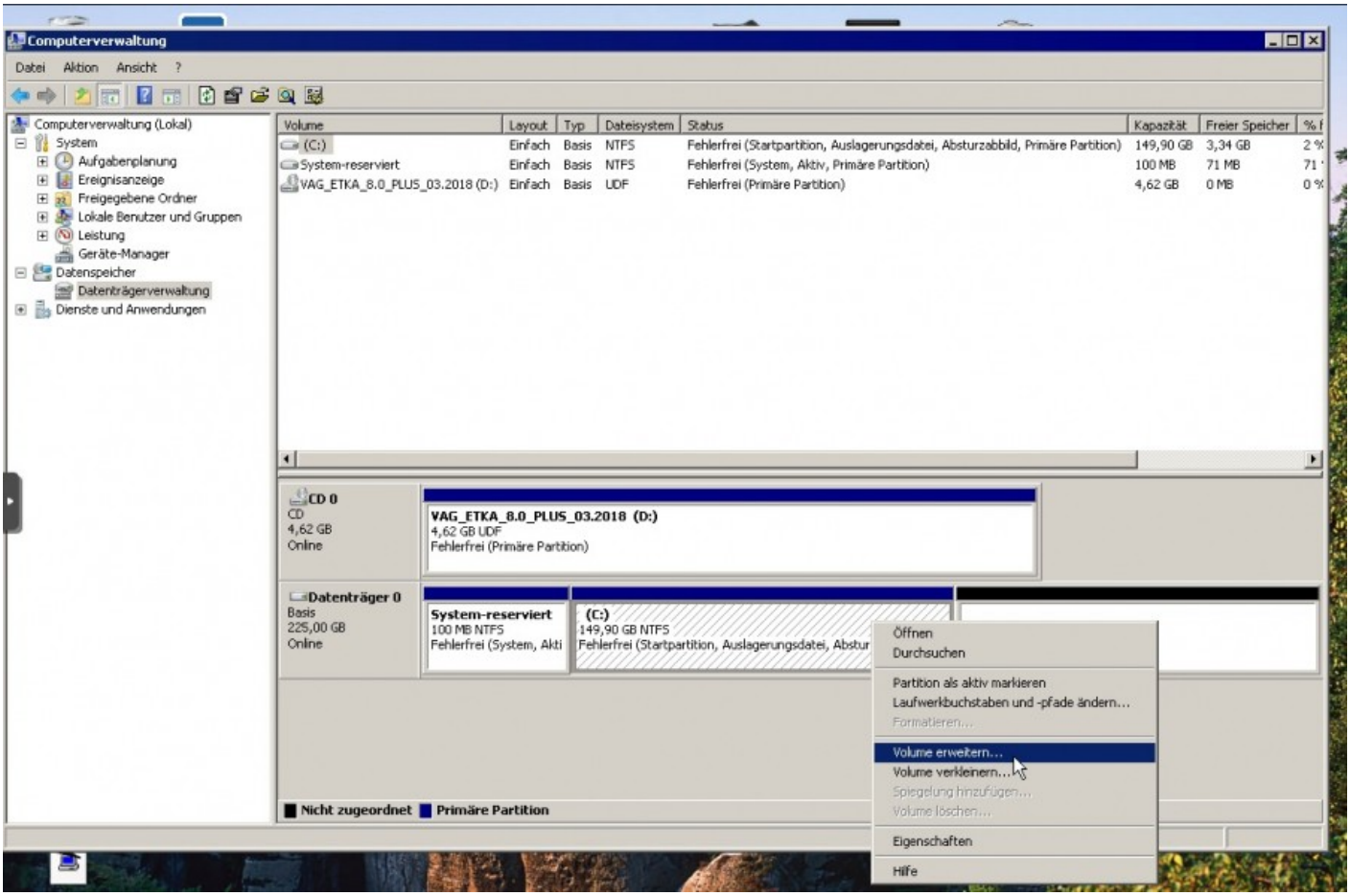
Linux-VM:

die VM mit einer Live-DVD von gParted booten und das Dateisystem auf die neue Größe anpassen,

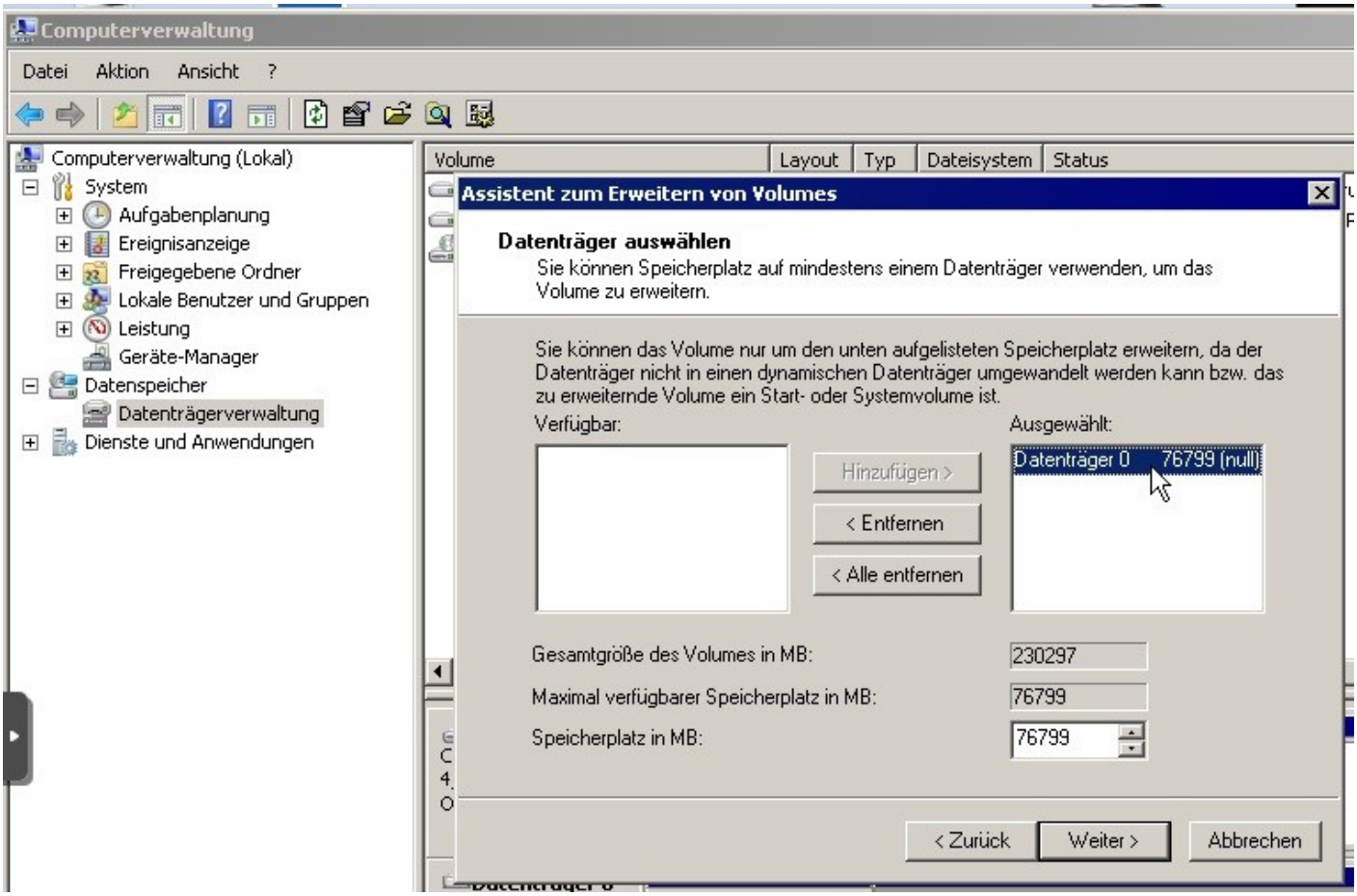
Windows VM:



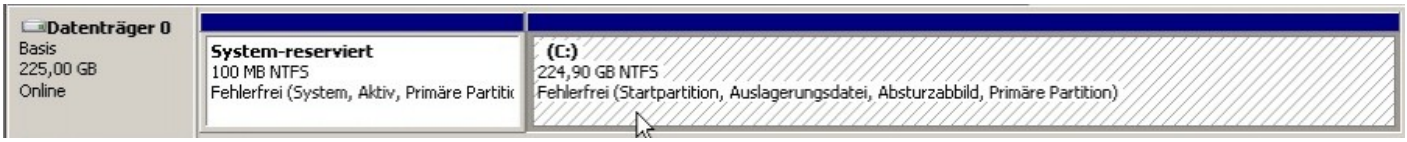
Per Rechtsklick auf „**Computer**“ im Startmenü in die **Verwaltung** wechseln



Über die Auswahl auf der linken Seite in die **Datenträgerverwaltung** wechseln und dort die Systemfestplatte auswählen und per Rechtsklick darauf die Funktion „**Volume erweitern…**“ auswählen



Im sich öffnenden Fenster durchklicken bis man zu dieser Ansicht gelangt, dabei ist sicherzustellen dass der dazugewonnene Speicher auf der **Ausgewählt** Seite steht



Nun das ganze mit Klicks auf **Weiter** und **Fertigstellen** abschließen, nach einem kurzen Moment wird die neue Datenträgergröße angezeigt

Images verkleinern

1. VM herunterfahren, anschließend mit einer Live-DVD von gParted booten und das Dateisystem verkleinern auf die gewünschte und vor allem mögliche Größe - Man kann eine Festplatte nicht kleiner machen als die insgesamt auf ihr gespeicherten Daten ohne einen kompletten Verlust eben jener befürchten zu müssen!!!
2. Ein neues Image erstellen in der neuen gewünschten Größe
   ```
   qemu-img create -f qcow2 -o preallocation=metadata newimage.qcow2 NEW_SIZE
   ```
3. Die Daten vom alten, größeren Image in das neue, kleinere Image kopieren
   ```
   virt-resize oldimage.qcow2 newimage.qcow2
   ```

Sollte das im vorherigen Schritt erstellte Image größer sein als die sich aktuell auf dem alten Image befindlichen Daten wird `virt-resize` eine Meldung ausgeben und ggf. eine zweite Partition mit dem übrig gebliebenem Speicherplatz erstellen - Im gegenteiligen Fall wird `virt-resize` die Aktion abbrechen und eine Meldung ausgeben wie viel Speicherplatz fehlt um die Aktion ausführen zu können
4. Die VM starten. Dabei kann es zu Fehlermeldungen kommen aufgrund von alten Datenträgerinformationen, diese sollte das Betriebssystem jedoch automatisch beheben können
5. Sollte nach dem erfolgreichen Booten der VM alles funktionieren kann das alte Image gelöscht werden

Hetzner - IPs auf virtuelle Maschinen via opnSense routen

1. **Netzwerk-Setup auf Proxmox**

Du brauchst mindestens zwei virtuelle Netzwerke in Proxmox:

- **vmbr0**: Das Hauptinterface, an das dein Hetzner-Host angebunden ist.
- **vmbr1**: Ein internes Netzwerk für deine VMs (z. B. `192.168.1.0/24`), das OPNsense als Gateway nutzt.

Konfiguriere die `/etc/network/interfaces` (falls nicht bereits geschehen):

```
auto vmbr0
iface vmbr0 inet static
    address <Hetzner-Haupt-IP>
    netmask 255.255.255.255
    gateway <Hetzner-Gateway>
    bridge-ports eno1
    bridge-stp off
    bridge-fd 0
    post-up echo 1 > /proc/sys/net/ipv4/ip_forward

auto vmbr1
iface vmbr1 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    bridge-ports none
    bridge-stp off
    bridge-fd 0
```

- `vmbr1` dient als internes Netzwerk für OPNsense und alle VMs.

---

2. **OPNsense einrichten**

Netzwerkkonfiguration in OPNsense:

- **WAN-Schnittstelle**: Verbinde OPNsense mit `vmbr0` (Hetzner-Hauptnetz)
- **LAN-Schnittstelle**: Verbinde OPNsense mit `vmbr1` (internes VM-Netz)

DHCP-Server für internes Netz:

- Aktiviere den DHCP-Server für `vmbr1` (`192.168.1.0/24`).
- Stelle sicher, dass OPNsense als Gateway fungiert (`192.168.1.1`).

NAT deaktivieren:

- OPNsense → Firewall → NAT → Outbound → "Hybrid Outbound NAT" oder "Manuell".
- Entferne oder deaktiviere alle NAT-Regeln für `vmbr1`, da du reines Routing nutzt.

---

3. **Hetzner /26-Subnetz auf OPNsense routen**

- Dein `/26`-Subnetz wird als "routed" Subnetz von Hetzner bereitgestellt.
- Trage eine der IPs als **WAN-Alias** in OPNsense ein:
  - **OPNsense → Interfaces → WAN → Aliases** → Füge z. B. `xxx.xxx.xxx.193/26` als Alias hinzu.
- Dann erstelle eine statische Route:
  - **OPNsense → System → Routes → Gateways** → Neues Gateway mit der Hetzner-Gateway-IP anlegen.
  - **OPNsense → System → Routes → Static Routes** → Route für dein `/26`-Subnetz eintragen.

---

4. **Zusätzliche IPs auf VMs routen**

- Gib einer VM eine interne IP (`192.168.1.100` via DHCP).
- Weise ihr eine externe IP (`xxx.xxx.xxx.194/26`) zu:
  - **OPNsense → Firewall → Virtual IPs** → "IP Alias" für `xxx.xxx.xxx.194` auf `vmbr1` (LAN).
  - Erstelle eine **Firewallregel** für eingehenden Verkehr auf diese IP.

Alternativ kannst du die VM so konfigurieren, dass sie eine externe IP selbst verwaltet:

- Auf der VM:

```
ip addr add xxx.xxx.xxx.194/32 dev eth0
ip route add default via 192.168.1.1
```

- In OPNsense eine statische Route anlegen:
  - Ziel: `xxx.xxx.xxx.194/32`
  - Gateway: `192.168.1.100`

Damit sollte jede VM eine interne IP per DHCP bekommen und du kannst ihr bei Bedarf eine externe IP zuweisen.