

Proxmox FAQ

- [Proxmox local erweitern / neu erstellen](#)
- [externe USB Festplatte mounten](#)
- [Synology vDSM als Docker in LXC](#)
- [IMG Datei in Proxmox importieren](#)
- [Passthrough von Festplatten in virtuelle Maschinen](#)

Proxmox local erweitern / neu erstellen

bash vom pve öffnen und folgende Befehle ausführen:

- `mkdir /pool`
- `pvesm remove local-lvm`
- `pvesm remove local`
- `lvremove /dev/pve/data`
- `lvresize -l +100%FREE --resizefs /dev/pve/root`

im Proxmox danach auf Rechenzentrum > Storage > Verzeichnis → /pool → Images, ISO, etc... anlegen

Danach Storage "local" deaktivieren in RZ > Storage

externe USB Festplatte mounten

per ssh mit Proxmox verbinden

- Mount Ordner erstellen : ***mkdir /mnt/usbplatte***
- per Proxmox GUI unter Rechenzentrum > Storage > Hinzufügen (Verzeichnis) (z.B. VZDUMP als Inhalt)
- per ssh Festplatte raussuchen: ***fdisk -l***
- Mouneten der Festplatte per Befehl: ***mount /dev/sdb1 /mnt/backup***

“ Für permanentes Mouneten:

UUID auslesen per ssh auf Proxmox: ***ls -l /dev/disk/by-uuid***

UUID irgendwo notieren und fstab bearbeiten per: ***nano /etc/fstab***

und als Zeile folgendes hinterlegen:

***UUID=c1d271f9-ebad-4e3d-8b87-fa654044xyz /mnt/usbfestplatte ext4
defaults,noatime,nofail 0 2***

Sollte es eine neue Festplatte sein, muss diese vorher formatiert sein

Synology vDSM als Docker in LXC

1. LXC Container erstellen, Updates und Docker Compose installieren
2. Script auf Proxmox Server ausführen:

```
#!/bin/bash

# Constants
CONFIG_DIR="/etc/pve/lxc"
TMP_DIR="/tmp"

# Function to display log messages
log() {
    echo -e "$1"
}

# Function to display error and exit
function display_error_and_exit() {
    log "Error: $1 Exiting."
    exit 1
}

# Function to display information
function display_info {
    clear
    log "This script is used to configure prerequisites to run Synology Virtual DSM"
    log "in a Docker container inside an unprivileged Proxmox LXC container."
    log "Please run this script on the Proxmox host, not inside the LXC container.\n"
}

# Check for root privileges
if [ "$EUID" -ne 0 ]; then
    display_error_and_exit "Please run this script as root."
fi
```

```
display_info

read -p "Do you want to continue? (y/n): " choice

if [[ $choice == "y" || $choice == "Y" ]]; then
    read -p "Enter the LXC Container ID (CT ID): " ct_id

    # Check if ct_id is a non-empty numeric value
    if [[ ! $ct_id =~ ^[0-9]+$ ]]; then
        display_error_and_exit "Invalid LXC Container ID. Please enter a numeric value."
    fi

    # Check if the configuration file exists
    config_file="$CONFIG_DIR/$ct_id.conf"
    if [[ ! -f "$config_file" ]]; then
        display_error_and_exit "Configuration file $config_file does not exist."
    fi

    # Check if the LXC container is running
    container_status=$(pct status $ct_id 2>&1)
    if [[ "$container_status" == *"running"* ]]; then
        log "Stopping running LXC container $ct_id..."
        pct stop $ct_id || display_error_and_exit "Failed to stop LXC container $ct_id."
    fi

    # Remove existing dev folder and tun, kvm, and vhost-net devices
    if [[ -d "/dev-$ct_id" ]]; then
        log "Removing existing /dev-$ct_id folder..."
        rm -r "/dev-$ct_id" || display_error_and_exit "Failed to remove existing /dev-$ct_id
folder."
    fi

    # Function to configure devices
    function configure_device() {
        device=$1
        module=$2
        major=$3
        minor=$4

        log "Configuring $device..."
    }
}
```

```

    mkdir -p "/dev-$ct_id/net" || display_error_and_exit "Failed to create /dev-
$ct_id/net"
    mknod "/dev-$ct_id/$device" c $major $minor || display_error_and_exit "Failed to mknod
/dev-$ct_id/$device"
    chown 100000:100000 "/dev-$ct_id/$device" || display_error_and_exit "Failed to chown
/dev-$ct_id/$device"

    #log "Checking if /dev-$ct_id/$device exists..."
    if ! [[ -e "/dev-$ct_id/$device" ]]; then
        display_error_and_exit "/dev-$ct_id/$device should have been created but does not
exist."
    fi
}

# Configure devices
configure_device "net/tun" "tun" 10 200
configure_device "kvm" "kvm" 10 232
configure_device "vhost-net" "vhost-net" 10 238

# Check and add configuration lines to /et/pve/lxc/<CT ID>.conf
log "Checking and adding configuration to $config_file..."
lines_to_add=(
    "lxc.mount.entry: /dev-$ct_id/net/tun dev/net/tun none bind,create=file 0 0"
    "lxc.mount.entry: /dev-$ct_id/kvm dev/kvm none bind,create=file 0 0"
    "lxc.mount.entry: /dev-$ct_id/vhost-net dev/vhost-net none bind,create=file 0 0"
)

# Error handling for config file changes
for line in "${lines_to_add[@]}; do
    if ! grep -qF "$line" "$config_file"; then
        echo "$line" >> "$config_file" || display_error_and_exit "Failed to add line
'$line' to $config_file."
    fi
done

log "Configuration completed successfully.\n\nStart the docker image (vdsm/virtual-
dsm:latest) inside the LXC container."

else
clear

```

```
log "\nScript aborted. No changes were made."
fi
```

virtuelle Festplatte erstellen, Mountpoint ist aus der `docker-compose.yaml` zu entnehmen / anzupassen:

```
services:
  dsm:
    container_name: dsm
    image: vdsm/virtual-dsm
    environment:
      DISK_SIZE: "10G"
    devices:
      - /dev/kvm
    cap_add:
      - NET_ADMIN
    ports:
      - 5000:5000
      - 5001:5001
    volumes:
      - /syno-space/dsm:/storage
    stop_grace_period: 2m
```

[GitHub - vdsm/virtual-dsm: Virtual DSM in a Docker container.](#)

IMG Datei in Proxmox importieren

Download der Datei über z.B wget

```
wget https://tombehrendt.de/download.img
```

Danach kann das IMG in die entsprechende VM sowie dem richtigen Storage importiert werden:

```
qm importdisk 110 download.img mypool
```

110 = VM ID

download.img = IMG Datei, die importiert werden soll

mypool = Storage-Pool von Proxmox

 mypool (pve-hp)

 pve-hp-localdir (pve-hp)

Passthrough von Festplatten in virtuelle Maschinen

Mittels per Befehl die aktuellen Festplatten anzeigen lassen:

```
ls -lah /dev/disk/by-id
```

```
ata-Samsung_SSD_860_EVO_1TB_S3Z6NB0K801659H -> ../../sda
ata-Samsung_SSD_860_EVO_1TB_S3Z6NB0K801659H-part1 -> ../../sda1
ata-Samsung_SSD_860_EVO_1TB_S3Z6NB0K801659H-part2 -> ../../sda2
ata-Samsung_SSD_860_EVO_1TB_S3Z6NB0K801659H-part3 -> ../../sda3
dm-name-pve-root -> ../../dm-1
dm-name-pve-swap -> ../../dm-0
dm-uuid-LVM-izHdpAXt103UZqUQrkqNzv2fM893bL3nqf6IlpaPoyZ2dG16Sfwv7UmZ2Q1WwC7U -> ../../dm-0
dm-uuid-LVM-izHdpAXt103UZqUQrkqNzv2fM893bL3nut4br49f3qfXXl2mqVzDHZLx7Pi9ySj7 -> ../../dm-1
lvm-pv-uuid-afcoq9-ygpW-pXbA-xL5i-dnzY-VGWB-9NkoVz -> ../../nvme0n1p3
nvme-eui.0025388881b88d94 -> ../../nvme0n1
nvme-eui.0025388881b88d94-part1 -> ../../nvme0n1p1
nvme-eui.0025388881b88d94-part2 -> ../../nvme0n1p2
nvme-eui.0025388881b88d94-part3 -> ../../nvme0n1p3
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328 -> ../../nvme0n1
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328_1 -> ../../nvme0n1
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328_1-part1 -> ../../nvme0n1p1
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328_1-part2 -> ../../nvme0n1p2
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328_1-part3 -> ../../nvme0n1p3
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328-part1 -> ../../nvme0n1p1
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328-part2 -> ../../nvme0n1p2
nvme-SAMSUNG_MZVLB256HAHQ-000L7_S41GNX1K832328-part3 -> ../../nvme0n1p3
wwn-0x5002538e406c4678 -> ../../sda
wwn-0x5002538e406c4678-part1 -> ../../sda1
wwn-0x5002538e406c4678-part2 -> ../../sda2
wwn-0x5002538e406c4678-part3 -> ../../sda3
```

Part 1, Part 2, etc sind Partitionen!

Um eine Festplatte an eine VM durchzureichen, folgenden Befehl ausführen:

```
qm set 222 -sata1 /dev/disk/by-id/ata-Samsung_SSD_860_EVO_1TB_XXXXXXXX
```

Die "222" ist dem Fall die VM ID! Sata1 ist der SATA-Port, der virtuell auf der VM steckt.

Danach ist die SATA Festplatte der VM zugewiesen und wird direkt durchgereicht:

Übersicht

Hinzufügen ▾

Entfernen

Bearbeiten

Disk-Aktion ▾

Zurücksetzen

>_ Konsole

Speicher 4.00 GiB

Hardware

Prozessoren 1 (1 sockets, 1 cores) [x86-64-v2-AES]

Cloud-Init

BIOS Standardeinstellung (SeaBIOS)

Optionen

Anzeige Standardeinstellung

Task History

Maschinentyp Standardeinstellung (i440fx)

Monitor

SCSI Controller VirtIO SCSI single

Backup

CD/DVD Laufwerk (ide2) none,media=cdrom

Replizierung

Laufwerk (sata0) pve-hp-localdir:222/vm-222-disk-1.raw,size=2G

Snapshots

Laufwerk (sata1) /dev/disk/by-id/ata-Samsung_SSD_860_EVO_1TB_S3Z6NB0K801659H,size=976762584K

Firewall ▶

Netzwerkkarte (net0) virtio=02:C8:9F:B2:18:E5,bridge=vibr0,firewall=1

Rechte