

# KVM (Kernel-based Virtual Machine)

## Quellen:

- <https://virt-manager.org/>
- <https://www.florian-fritsch.com/kvm-unter-debian-10-mit-windows-10-vm/>
- <https://blog.thul.org/technik/anwendungen/virtualisierung-unter-debian/>
- [https://wiki.libvirt.org/page/The\\_daemon\\_cannot\\_be\\_started](https://wiki.libvirt.org/page/The_daemon_cannot_be_started)

Virtuelle Maschinen werden auf einem laufenden OS (Server) installiert und können dort mit einem beliebigen (anderen) OS (Linux, Windows) betrieben werden. Die Verwaltung der VM's erfolgt über die GUI des Virt-Manager, das auch auf einer anderen Maschine laufen kann → Fernwartung.

## Server, Wirt der VM

Prüfen ob die CPU des Servers geeignet ist, um virtuelle Maschinen zu betreiben

```
egrep '^flags.*(vmx|svm)' /proc/cpuinfo
```

Sofern farblich „vmx“ (Intel) oder „svm“ (AMD) auftaucht, unterstützt die CPU Virtualisierung.

Installation

```
sudo apt install qemu-kvm libvirt-clients libvirt-daemon-system bridge-utils  
virtinst libvirt-daemon virt-manager qemu-utils -y
```

check Status

```
sudo systemctl status libvirtd.service
```

Zugriffrechte anpassen, User zu Gruppen hinzufügen

```
sudo usermod -aG libvirt <USER>
```

```
sudo usermod -aG libvirt-qemu <USER>
```

Spezial bei SRV134 (intern):

```
sudo usermod -aG alle libvirt-qemu
```

## Speicherort KVM-Images

<https://ostechnix.com/how-to-change-kvm-libvirt-default-storage-pool-location/>

Zeige laufende VM's

```
sudo virsh list --all
```

Zeige Pools

```
sudo virsh pool-list
```

Zeige Details des Pools „default“

```
sudo virsh pool-info default
```

Zeige den Datenpfad für den Pool „default“

```
sudo virsh pool-dumpxml default | grep -i path
```

Zeige alle VM's auf diesem Datenpfad

```
sudo virsh vol-list default | grep "/var/lib/libvirt/images/"
```

```
sudo virsh vol-list default | grep "/home/kvm/images/"
```

Editiere den Datenpool default → neuen Datenpfad eingeben (z.B. /home/kvm/)

```
sudo virsh pool-edit default
```

```
<pool type='dir'>
  <name>default</name>
  <uuid>76f47a3c-7656-4c34-9ab5-c87e5136d532</uuid>
  <capacity unit='bytes'>396452147200</capacity>
  <allocation unit='bytes'>71178977280</allocation>
  <available unit='bytes'>325273169920</available>
  <source>
</source>
  <target>
    <path>/home/kvm</path>
    <permissions>
      <mode>0755</mode>
      <owner>1000</owner>
      <group>1000</group>
      <label>unconfined_u:object_r:svirt_home_t:s0</label>
    </permissions>
  </target>
</pool>
```

Anpassen: PATH /home/kvm

Stop Pool „default“

```
sudo virsh pool-destroy default
```

```
sudo virsh pool-undefine default
```

Default pool neu erstellen

```
sudo virsh pool-define-as --name default --type dir --target /home/kvm/
```

Anpassen: PATH /home/kvm

Pool starten und in zukunft automatisch starten

```
sudo virsh pool-start default
```

```
sudo virsh pool-autostart default
```

Check den Pfad für default pool

```
sudo virsh pool-dumpxml default | grep -i path
```

Check pool-list

```
sudo virsh pool-list
```

Restart an Check libvirt

```
sudo systemctl restart libvirtd
```

```
sudo systemctl status libvirtd
```

## Umzug KVM-Image

### Auf altem System:

Registrierte Images ansehen

```
sudo virsh list --all
```

Altes System (VNAME ersetzen mit korrektem Namen):

```
sudo virsh dumpxml VMNAME > /tmp/VMNAME.xml
```

```
sudo virsh undefine VMNAME
```

Die Datei VNAME.qcow2 an den neuen Ort kopieren.

Bei Systemwechsel auch die XML-Datei auf das neue System kopieren.

### Auf neuem System:

Pfad in der XML-Datei korrigieren, z.B. auf /var/lib/libvirt/images

```
sudo nano /tmp/VMNAME.xml
```

```
<source file='/var/lib/libvirt/images/VMNAME.qcow2' />
```

Virtuelle Maschine auf neuem System oder geändertem Pfad aktivieren.

```
sudo virsh define /tmp/VMNAME.xml
```

Registrierte Images ansehen

```
sudo virsh list --all
```

Starten einer virtuellen Maschine

```
sudo virsh start VMNAME
```

## Netzwerkbrücke

Software „bridge-utils“ bereits installiert - s.o.

Ist die Netzverwaltung auf dem Server mit Netplan realisiert: siehe [Bridges mit Netplan](#).

```
sudo nano /etc/network/interfaces
```

Beispiel

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto br0
iface br0 inet dhcp
    bridge_ports eth0
```

br0 = Name der Netzwerkbrücke

eth0 oder eno1 oder ... = Name der Netzwerkkarte

Ggf. Netzwerk neu starten

```
sudo service networking restart
```

## Download Betriebssysteme

Debian 11 - ISO download in das Verzeichnis für OS-Images

```
cd /home/kvm/
```

```
wget
https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-11.6.0-amd64-netinst.iso
```

Images für Windows finden sich unter: <https://www.microsoft.com/de-de/software-download>  
Sollte man die Seite unter einem Windows-OS aufrufen, dann sind die Download-Optionen dadurch ggf. eingeschränkt. Unter einem Linux-System kann man problemlos Images aller gängigen Win-Systeme downloaden. **Natürlich ohne Lizenz!**

Installation dann über das GUI des Virt-Manager

## Administration

Virt-Manager installieren, kann auch auf einem anderen System erfolgen.

```
sudo apt install virt-manager
```

Starten über die GUI

## Festplattenspeicher erweitern

Das virtuelle System muss vor der Vergrößerung heruntergefahren werden.  
Auf dem Server in das Verzeichnis des Images wechseln.  
Anzeigen der aktuellen Größe:

```
sudo qemu-img info win10.qcow2
```

win10.qcow2 = Image

Vergrößern (hier um 10 GB):

```
sudo qemu-img resize win10.qcow2 +10G
```

Dann mit Hilfe der Windows-Datenträgerverwaltung die Festplatte erweitern. Erweitern funktioniert allerdings nur, wenn der (neue) freie Speicher direkt neben der Partition liegt, die erweitert werden soll. Anderfalls ist das „Erweitern“ ausgegraut und es wird eine externes Tool benötigt. Z.B. die Freeware-Version von [Minitool](#).

From:

<https://wiki.bluegnu.de/> - **wiki**

Permanent link:

<https://wiki.bluegnu.de/doku.php/open:it:kvm?rev=1724079213>

Last update: **2024/08/19 16:53**

