Virtualisierung mit qemu

Stand: 05.03.2022 Quelle: https://www.freiesoftwareog.org

Inhaltsverzeichnis

Vorbereiten bzw. Installieren der nötigen Pakete

Vor dem „Virtualisierungs-Vergnügen“ müssen zunächst noch einige Vorbereitungen getroffen werden.

Systemtest, ob Virtualisierung überhaupt möglich, bzw. aktiviert ist

Normalerweise sollte man stand Heute davon ausgehen, dass halbwegs aktuelle Systeme alle mit integrierter Virtualisierungstechnik ausgestattet sind. Um sicher zu gehen, kann man jedoch folgenden einfachen Test in einem Terminal ausführen:

```
egrep -c ‘(vmx |svm)‘ /proc/cpuinfo
```

Dies sollte eine einzige Zahl ausgeben. Ist diese größer als „0“, gibt es einen der beiden Begriffe (vmx für Intel, svm für AMD) und das System ermöglicht Virtualisierung.

Sollte es „0“ sein, kann eine letzte Chance auch noch ein Blick ins BIOS des PCs helfen.

Oftmals ist die Virtualisierung dort einfach nicht aktiviert.

Installation der benötigten Pakete

Mit der folgenden Terminal-Zeile werden alle notwendigen Pakete für die Virtualisierung mit qemu installiert:

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

Danach kann/muss noch ein Symbolischer Link gesetzt werden:

```
sudo ln -s /usr/bin/qemu-system-x86_64 /usr/bin/qemu
```

Dann kann man noch testen, ob der libvirt daemon läuft:

```
sudo systemctl is-active libvirtd
```

Falls das nicht automatisch geschehen ist, kann man noch den eigenen Benutzer zu den notwendigen Gruppen hinzufügen:

```
sudo usermod -aG libvirt $USER

sudo usermod -aG kvm $USER
```

Verwendung von qemu im Terminal

Man kann qemu direkt ohne grafische Oberfläche verwenden.
Die Grundfunktionen sind recht einfach mit Parametern festzulegen.

Booten eines Live-Images (ISO oder USB)

Wenn man nur mal schnell eine neue Distribution ausprobieren möchte, kann man diese direkt vom heruntergeladenen Abbild (iso oder img) tun.
Dazu setzt man den folgenden Befehl im Verzeichnis des Downloads ab:

```
qemu-system-x86_64 -cdrom dateiname.iso
```

Wobei „dateiname.iso“ die Bezeichnung des Abbildes sein sollte.
Hier wird mit dem Parameter „-cdrom“ angewiesen, dass von „CD“ gestartet werden soll und das Argument für den Parameter ist der Name des Abbildes.
Hat man z.B. einen Live USB-Stick, von dem man virtuell starten möchte, verhält es sich (fast) genauso, nur mit anderem Parameter und Argument:

```
qemu-system-x86_64 -hda /dev/sdx
```

Hier wird mit dem Parameter „-hda“ festgelegt, dass das nachfolgende Argument eine bootfähige Festplatte ist, in diesem Fall der eingesteckte USB-Stick. Der Name des Gerätes (sdx) muss natürlich an die Gegebenheiten angepasst werden.

Installieren eines Betriebssystems

Möchte man eine virtuelle Installation haben, werden folgende Schritte erforderlich.

Erzeugen einer virtuellen Festplatte

Damit man eine „Heimat“ für das virtualisierte Gastsystem hat, muss man zunächst eine Festplatte erzeugen:

```
qemu-img create dateiname.img -f qcow2 20G
```

Die Option -f legt das Format der virtuellen Festplatte fest, in diesem Fall qcow2.
Dieses Format ist praktisch, da es zur Laufzeit nach Bedarf „wächst“, d.h. es wird immer nur soviel Platz reserviert, wie gerade notwendig ist. Die Grenze ist natürlich die absolute Größe, welche mit dem Parameter „20G“ angegeben wird.
Qemu/KVM bieten einige weitere Formate, die je nach Bedarf eingesetzt werden können.

Installation des Betriebssystems

```
qemu-system-x86_64 -enable-kvm -cdrom ubuntu-20.10-desktop-amd64.iso -boot d →
dateiname.img -m 2048
```

Die Parameter und ihre Argumente kurz erklärt:

- **-enable-kvm** initialisiert die Nutzung von KVM
- **-cdrom** bestimmt das zu verwendende optische Laufwerk. Dies kann ebenfalls ein Abbild sein, oder das tatsächliche physische Laufwerk des Hostsystems
- **-boot** bestimmt die virtuelle Festplatte, auf die das Gastsystem installiert werden soll, sowie den Ort des Bootloaders. Die Argumente hierfür sind:
  - c – boote die erste virtuelle Festplatte
  - d – boote das erste virtuelle optische Laufwerk
  - n – boote von einem virtuellen Netzwerk
- **-m** teilt qemu mit, wie viel des Arbeitsspeichers des Hostsystems verwendet werden soll

Alternativ:

```
kvm -enable-kvm -cdrom ubuntu-20.10-desktop-amd64.iso -boot d →
dateiname.img -m 2048
```

Starten des virtuellen Rechners

```
qemu-system-x86_64 -enable-kvm -vga std -usb -smp 2 -hda dateiname.qcow2 →
-cdrom /dev/sr0 -m 3072
```

Die Parameter und ihre Argumente kurz erklärt:

- **-enable-kvm** initialisiert die Nutzung von KVM
- **-vga** legt die Grafikkarte fest. „std“ ist der im Kernel eingebaute Standard und bietet Auflösungen >= 1280x1024
  Für bessere Performance und andere Auflösungen gibt es das Argument „qxl“. Dazu müssen „SPICES“ genutzt werden
- **-usb** schleift die USB-Schnittstellen des Hostsystems durch
- **-smp** legt mit seinem Argument die Anzahl der zu verwendenden Prozessorkerne des Hostsystems fest
- **-hda** bestimmt die zu verwendende Festplatte bzw. das Image
- **-cdrom** bestimmt das zu verwendende optische Laufwerk. Dies kann ebenfalls ein Abbild sein, oder das tatsächliche physische Laufwerk des Hostsystems
- **-m** teilt qemu mit, wie viel des Arbeitsspeichers des Hostsystems verwendet werden soll

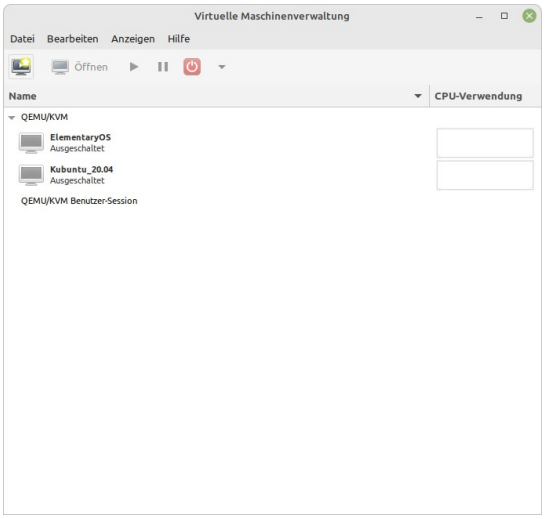
Alternativ:

```
kvm -enable-kvm -vga std -usb -smp 2 -hda dateiname.qcow2 -cdrom /dev/sr0 -m 3072
```

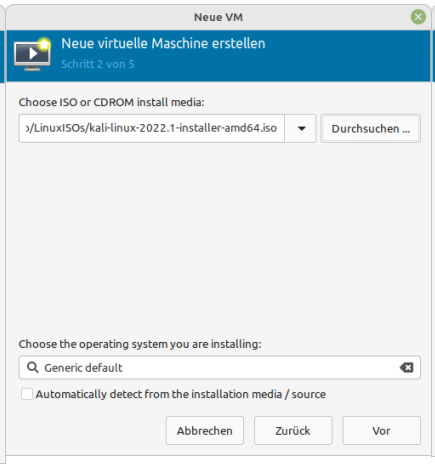
Verwendung von qemu mit GUI „virt-manager“ aka „Virtuelle Maschinenverwaltung“

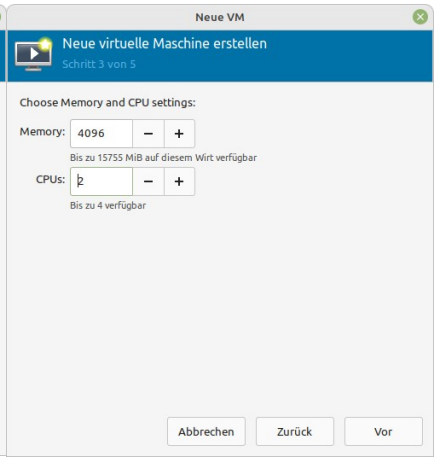
Es gibt genau wie bei VirtualBox auch ein grafisches Frontend für qemu.
Dieses wurde im Kapitel „Installation der benötigten Pakete“ bereits mitinstalliert.
Aufruf über das Startmenü „Systemverwaltung – Virtuelle Maschinenverwaltung“ (Linux Mint).

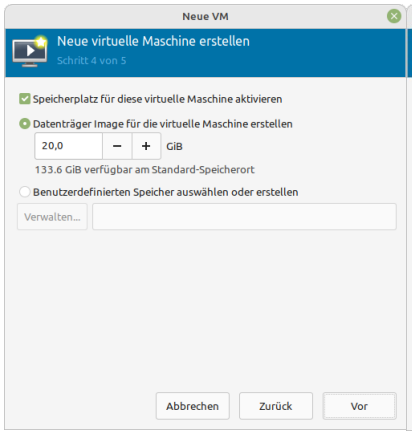
Übersicht

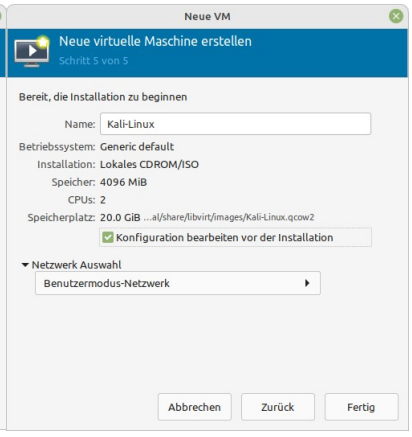


Erstellen einer Virtuellen Maschine

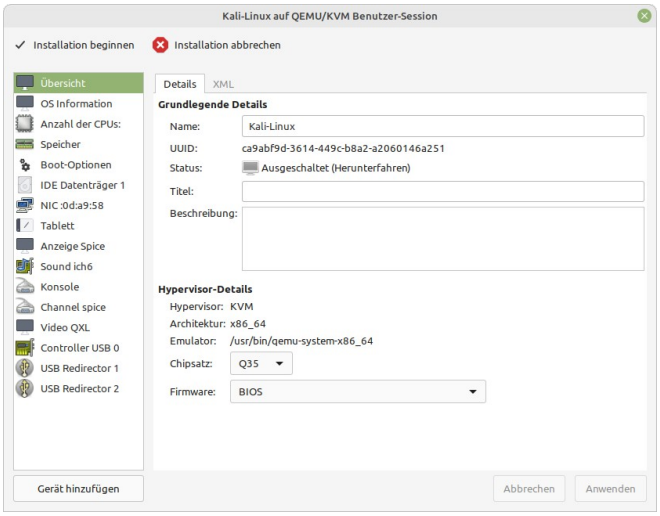








Konfigurationsdialog



Die Möglichkeiten des grafischen Frontends sind nicht Teil dieses Artikels.
Stattdessen verweise ich auf die Fülle an How-Tos im Internet.

Snapshots von virtuellen Maschinen erzeugen

Manchmal möchte man eine Momentaufnahme einer virtuellen Maschine erstellen, um bei Problemen wieder auf einen funktionierenden Zustand zurückkehren zu können.
Beispielsweise vor einem Update oder riskanten Experimenten. Dazu gibt es die Möglichkeit mit dem Tool qemu-img. Dies funktioniert allerdings nur bei cqow2-Abbildern.

Bestehende Snapshots auflisten

Wenn man sich eine Übersicht von bestehenden Snapshots einer virtuellen Maschine machen will, verwendet man folgenden Befehl:

```
qemu-img snapshot -l Ubuntu.qcow2
```

Erstellte Snapshots erscheinen auch in der Ausgabe des Befehls:

```
quemu-img info Ubuntu.qcow2
```

Einen Snapshot erzeugen

Zur Erzeugung eines Snapshots verwendet man:

```
qemu-img snapshot -c Ubuntu_VorUpdate_05.03.22 Ubuntu.qcow2
```

Einen Snapshot zurückspeichern

Um einen erzeugten Snapshot auf eine virtuelle Maschine zurückzuspielen verwendet man:

```
qemu-img snapshot -a Ubuntu_VorUpdate_05.03.22 Ubuntu.qcow2
```

Einen Snapshot löschen

Wenn ein Snapshot obsolet geworden ist, kann man diesen auch wieder löschen:

```
qemu-img snapshot -d Ubuntu_VorUpdate_05.03.22 Ubuntu.qcow2
```

Sonstige Tipps

Konvertierung virtueller Maschinen aus anderen Formaten

Es ist möglich, virtuelle Maschinen aus anderen Programmen (VMware, VirtualBox, ...) so umzuwandeln, dass diese mit qemu verwendet werden können.

Dies wird mit dem Hilfsprogramm qemu-img und dessen Konvertierfunktion erledigt.

Verfügbare Formate:

Abbildformat	Argument für qemu-img
Qcow2 (KVM, Xen)	qcow2
QED (KVM)	qed
raw	raw
VDI (VirtualBox)	vdi
VHD (Hyper-V)	vpc
VMDK (VMware)	vmdk

.vmdk (VMware)

```
qemu-img convert -p -f vmdk -O qcow2 original.vmdk original.qcow2
```

.vdi (VirtualBox)

Für die Umwandlung eines VirtualBox Abbildes werden zwei Schritte benötigt.

Zunächst muss die .vdi Datei mit einem VirtualBox-Werkzeug in raw umgewandelt werden:

```
VBoxManage clonehd ~/VirtualBox\ VMs/dateiname.vdi dateiname.img --format raw
```

Danach kann mit qemu-img dieses .raw in .qcow2 umgewandelt werden. Siehe nächster Abschnitt.

.raw (einfaches Festplattenabbild)

```
qemu-img convert -p -f raw -O qcow2 dateiname.img dateiname.qcow2
```

Die Parameter und ihre Argumente kurz erklärt:

- **-p** zeigt eine Fortschrittsanzeige an
- **-f** Gibt das Quellformat an (optional, wird normalerweise automatisch erkannt)
- **-O** Nach diesem Parameter stehen das Zielformat, der Quelldateiname und der Zieldateiname

Virtuelle Festplatte vergrößern

Man kann eine einmal angelegte virtuelle Festplatte auch nachträglich vergrößern.

Dies geschieht einfach über folgenden Befehl (z.B. hier: 10 Gigabyte mehr):

```
sudo qemu-img resize -p dateiname.qcow2 +10G
```

Weitere qemu Parameter

Hier noch einige weitere Parameter, welche evtl. gebraucht werden können:

Parameter	Beschreibung	Mögliche Werte für x (Auswahl)
-M <x>	Angabe des Maschinentyps	Q35,accel=kvm
-cpu <x>	Angabe des Prozessortyps	Opteron_G5 (AMD Opteron 63xx CPU) Nehalem (Intel Core i7 9xx) Penryn (Intel Core 2 Duo P9xxx) host (empfohlen)
-smp <x>	Prozessor-Topologie, Anzahl Kerne	4 $(nproc) verwendet alle verfügbaren
-k <x>	Tastatur-Layout	de
-boot <x>	Angabe des Bootgeräts	c (1. virtuelle Festplatte) d (1. virtuelles CD-ROM) n (Virtuelles Netzwerk)
-vga <x>	Angabe der emulierten Grafikkarte	Cirrus (einfache Grafikkarte) std (Auflösung bis 1280x1024) virtio (??) qxl (Starke Grafikkarte mit SPICE)
-usbdevice <x>		Tablet (Tablet statt PS/2 Maus, empfohlen)
-usb	USB aktivieren	
-netdev	Angaben zur Netzwerkkarte und -verbindung	-netdev user,id=n1 -device virtio,netdev=n1