

# Virtualisierung unter Linux (mit Bordmitteln)

PC-Treff-BB

Roland Egeler

# Agenda

- Fokus
- Bordmittel
- Einrichtung
- Grafische Oberfläche
- Dateiformat (qcow2)
- Konvertierung von anderen Dateiformaten
- Ausblick

## Fokus

- Kenntnisse über Virtualisierung werden vorausgesetzt
- Nicht betrachtet werden andere frei verwendbare bzw. käufliche Produkte wie
  - VMware
  - VirtualBox
  - ...
- Es werden nur Programme benutzt, die in der Distribution mitgeliefert werden
- Es geht nur um Voll-/Paravirtualisierung
  - Es werden komplette Betriebssysteme virtualisiert
  - Fremde Prozessoren können emuliert werden
  - Keine Container

# Bordmittel

- Themen sind
  - KVM: [KVM], [KVM2]
  - QEMU: [QEMU], [QEMU2]
  - [libvirt]
  - [libvirtd]
  - virt-manager: [virt-manager], [virt-manager2]
- Siehe Anhang „Quellen“

## Bordmittel KVM

- KVM bedeutet „Kernel-based Virtual Machines“
- Ist ein Kernelmodul (seit 2007, ab Kernel 2.6.20)
- Ist ein Hypervisor
- Ob Typ-1 oder Typ-2 ist sich das Netz nicht einig
- Läuft innerhalb eines anderen Betriebssystems (Linux)
- Benutzt Ressourcen, die Linux bereitstellt
- Schnittstelle zur Ausführung von virtuellen Maschinen
- Stellt VMs Ressourcen bereit
  - CPU
  - RAM
  - Plattenplatz

# Bordmittel QEMU

- Heißt „Quick Emulator“
- Simuliert einen virtuellen Computer
  - CPU
  - Mainboard
  - BIOS
  - Controller
    - Netzwerk
    - Platte
    - USB
    - Seriell
  - Steckkarten
  - ...

# Bordmittel QEMU

- Kann andere CPU-Architekturen emulieren
  - ARM
  - SPARC
  - Power
  - RISC-V...
- Sonderfall Emulation x86 auf x86
- Kann Prozessorerweiterungen benutzen
  - Intel VT
  - AMD-V
- Werden von KVM zur Verfügung gestellt
- Läuft daher nahezu gleich schnell wie das Original

## Bordmittel QEMU

- Unterstützt Paravirtualisierung
- Schnittstelle „virtio“
- Treiber für „virtio“ müssen im Betriebssystem der virtuellen Maschine geladen sein
- Kann die Dateiformate von anderen VMs benutzen
  - VMware (.vmdk)
  - VirtualBox (.vdi)
  - Microsoft HyperV (.vhd)
  - Raw (.img), mit „dd“ erzeugt
  - ...



## Bordmittel libvirt

- Sammlung von Funktionen zur Virtualisierung
- Enthält
  - API
  - Daemon „libvirtd“
  - Kommandozeilenbefehl „virsh“
  - Einheitliche Schnittstelle für Hypervisoren

## Bordmittel libvirt

- Ein Hintergrundprozess („daemon“)
- Ermöglicht die Kommunikation mit den VMs
- Wird zur Steuerung der VMs benutzt
- Ermöglicht anderen Programmen die Anzeige der Benutzeroberfläche
- Ermöglicht den Betrieb der VMs im Hintergrund
- Ohne libvirt keine VMs

## Bordmittel virt-manager

- Grafische Oberfläche zur Verwaltung von VMs
- Kommuniziert mit libvirtd
- Benutzt XML-Dateien zur Speicherung der Konfigurationen der VMs
- Siehe auch „Grafische Oberfläche“

# Einrichtung

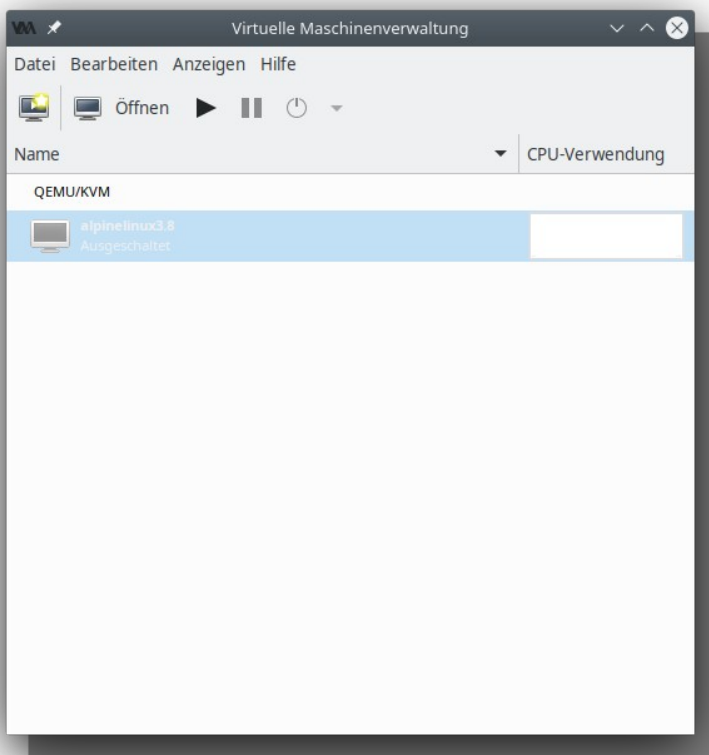
- Hier werden nur „debianoide“ Linuxe betrachtet
  - `sudo apt -y install bridge-utils cpu-checker libvirt-clients libvirt-daemon qemu qemu-kvm`
- Schauen, ob KVM da ist:
  - `kvm-ok`
  - INFO: `/dev/kvm` exists
  - KVM acceleration can be used
- Installation virt-manager
  - `sudo apt install virt-manager`

# Einrichtung

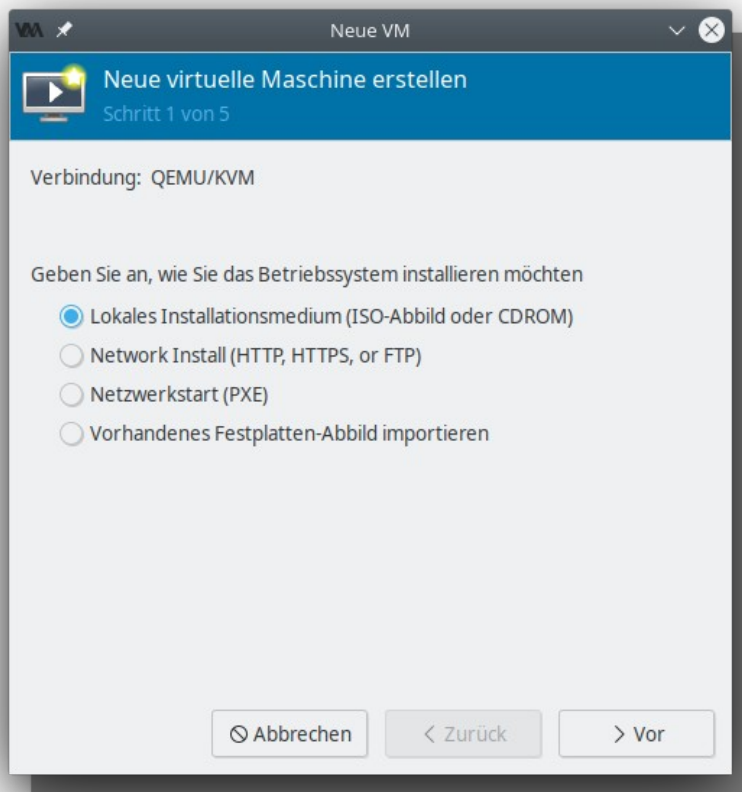
- virt-manager darf VM-Dateien nur benutzen, wenn der Benutzer in der richtigen Gruppe ist
- Man ist versucht, ihn als „root“ zu starten
- „root“ darf aber normalerweise kein X-Windows benutzen
- Das kann man aber einrichten
- Siehe [X11]
- Alternativ als normaler Benutzer arbeiten
- Darf dann auch X-Windows
- Normaler Benutzer muss in Gruppe „libvirt“
  - `adduser <youruser> libvirt`
- Siehe auch [arch], [ubuntu], [redhat], [debian]

# Grafische Oberfläche

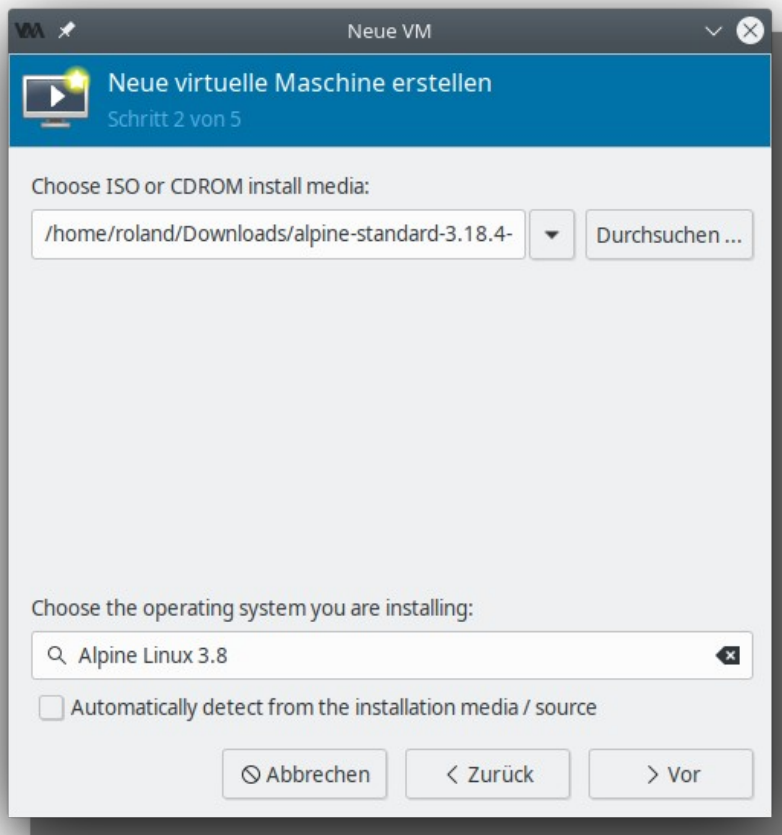
- virt-manager zeigt eine Oberfläche, die den kommerziellen Produkten sehr ähnlich ist



- Neue Maschine erstellen:

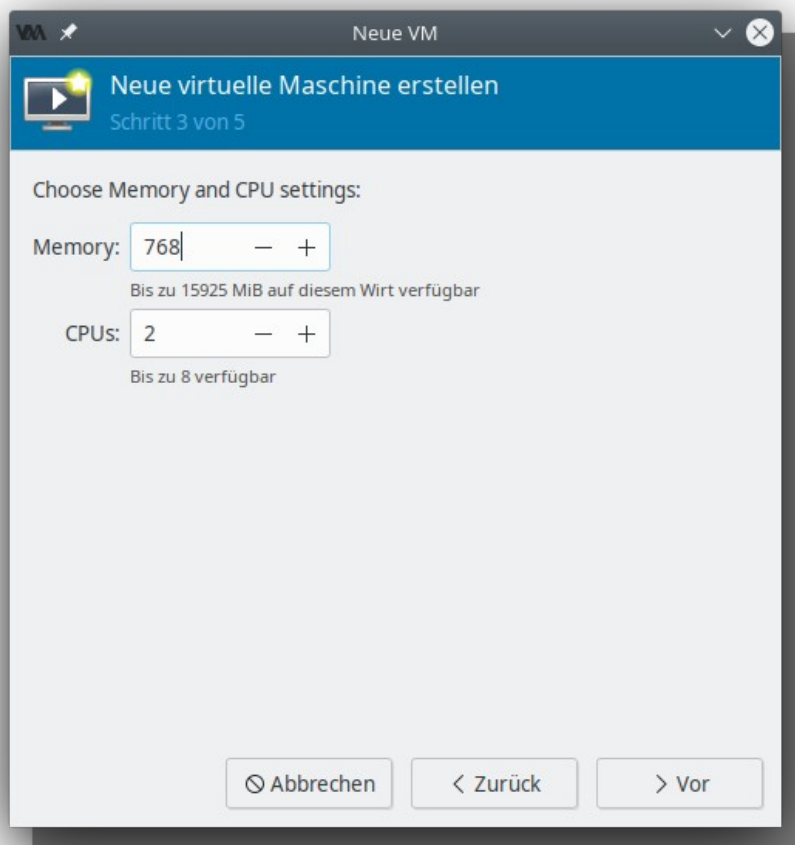


- Neue Maschine erstellen:



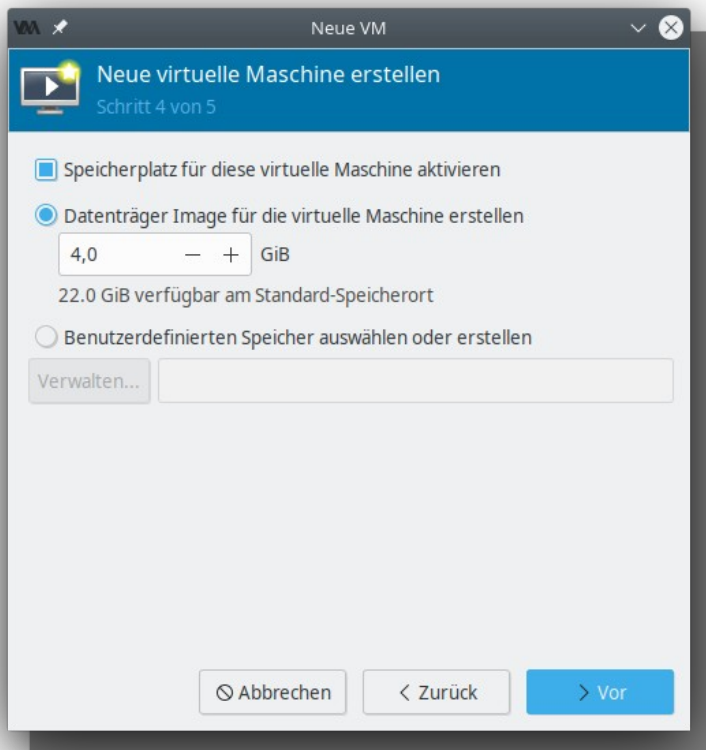


- Neue Maschine erstellen:

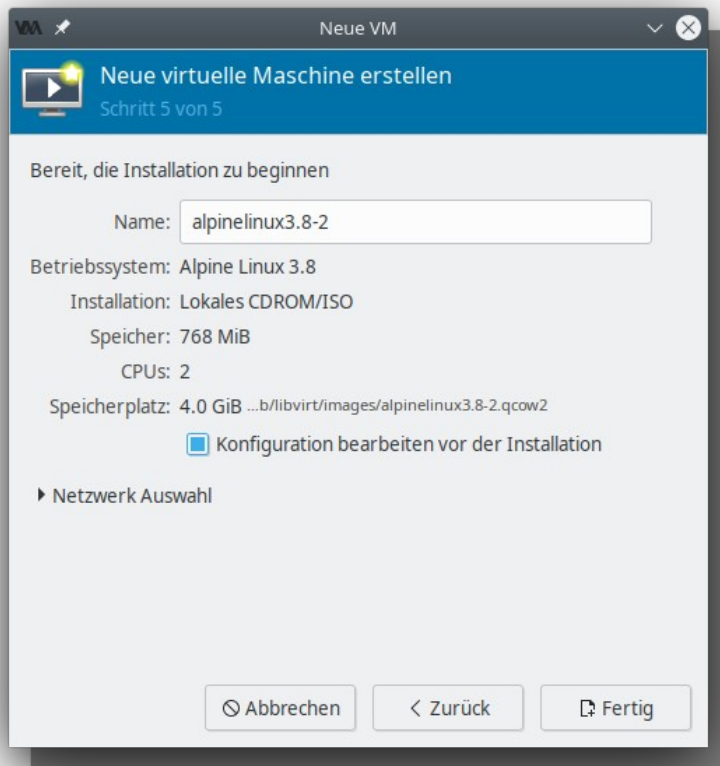


# Grafische Oberfläche

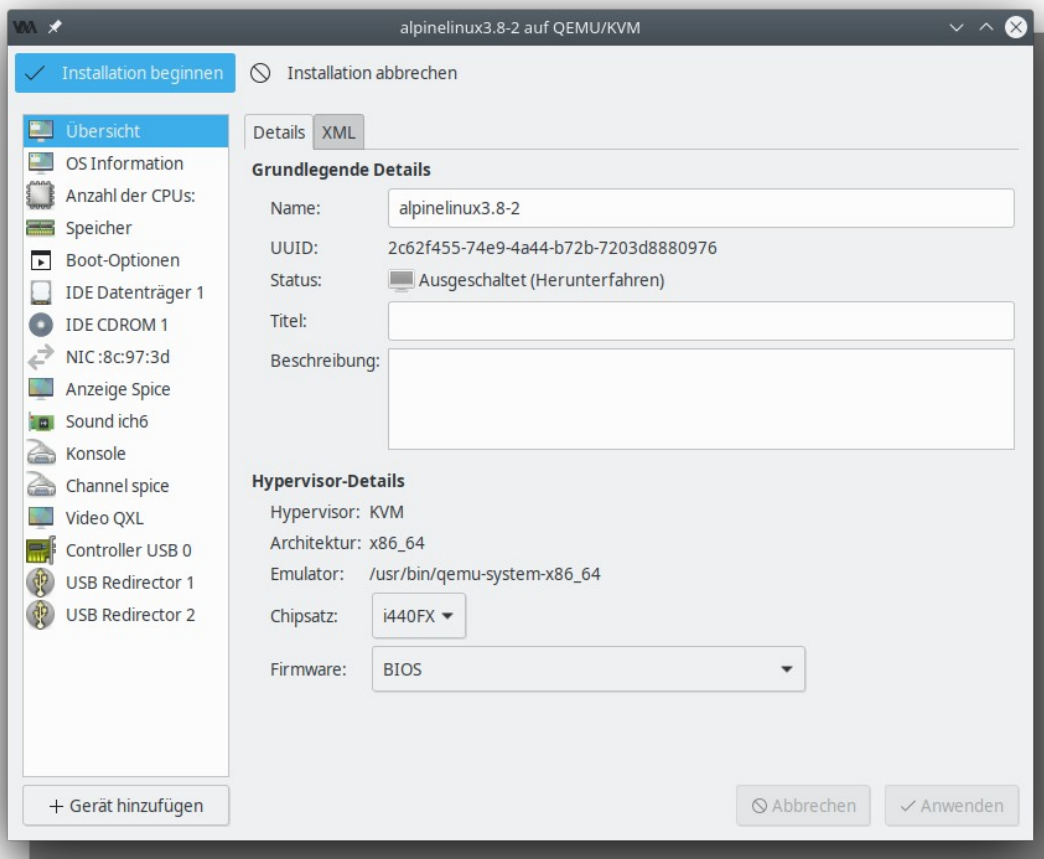
- Neue Maschine erstellen:



- Neue Maschine erstellen:

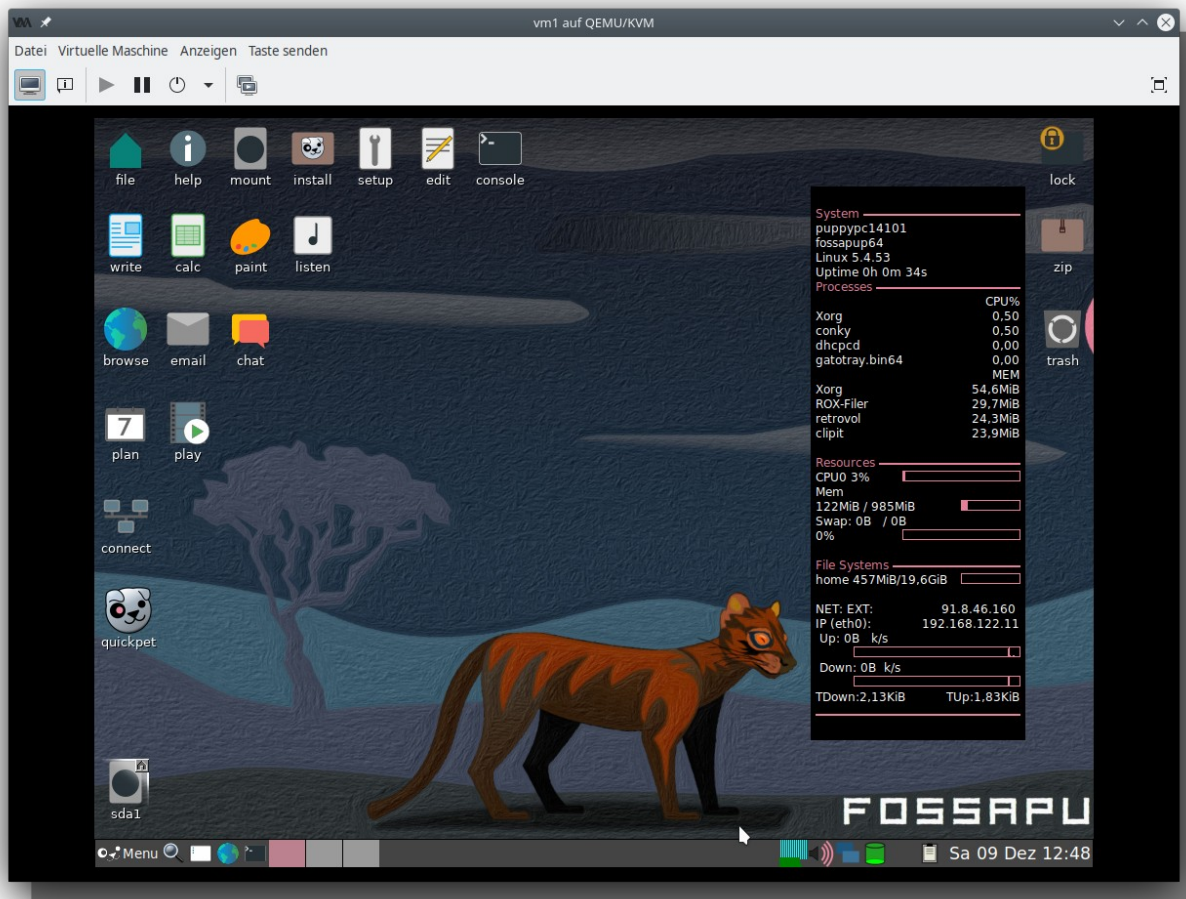


- Neue Maschine erstellen:



# Grafische Oberfläche

- Neue Maschine erstellen:

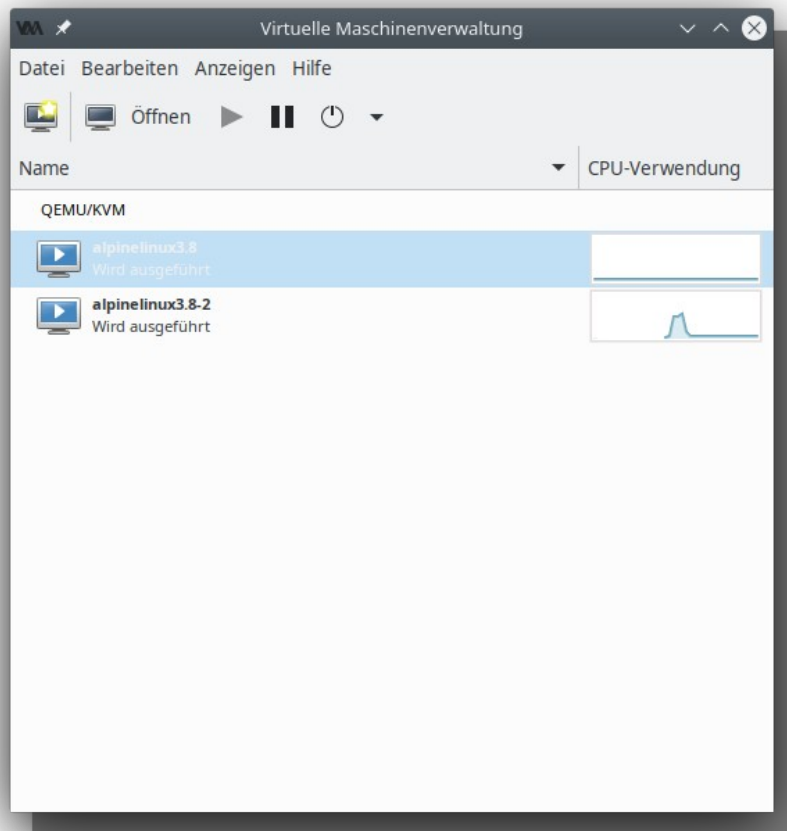


## Grafische Oberfläche

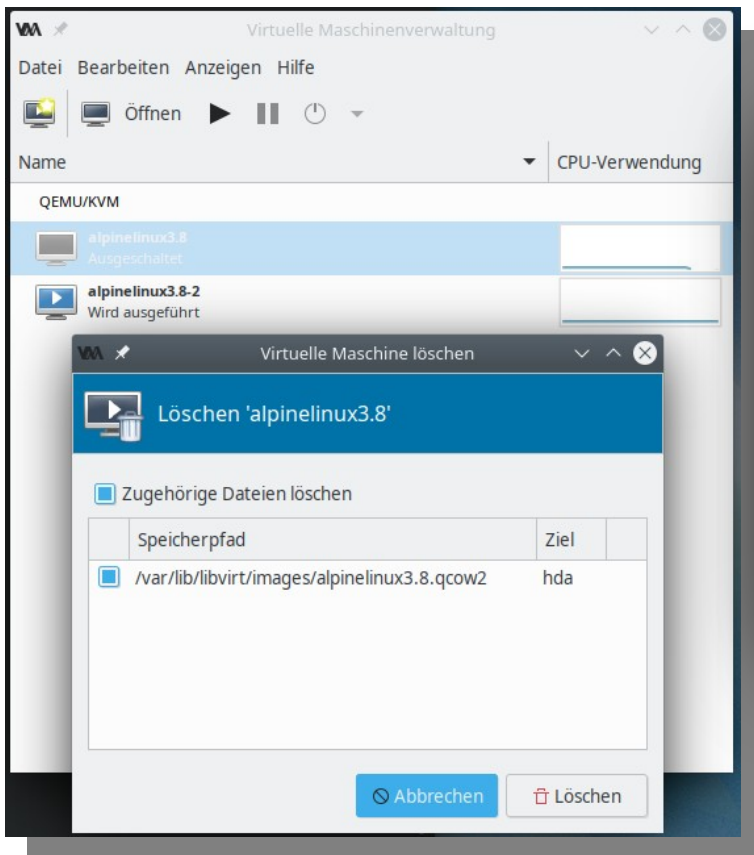
- Beim Klick in das Fenster wird der Mauszeiger „gefangen“
- Man bekommt ihn durch gleichzeitiges Drücken von „Strg“ und „Alt“ (links der Leertaste) wieder

# Grafische Oberfläche

- CPU-Verwendung wird angezeigt



- Maschine löschen:





# Dateiformat

- QEMU benutzt „qcow2“
- „cow“ heißt „copy on write“
- Erlaubt die Allokation von Speicherplatz erst wenn er gebraucht wird
- Siehe [sparse]
- Auch auf Dateisystemen, die das nicht selbst mitbringen
- Kann komprimieren
- Trivialerweise ist „qcow2“ eine erweiterte Version von „qcow“
- Kann Verschlüsselung
- Kann „Snapshots“

## Dateiformat

- Nach Löschung keine Änderung des Plattenplatzes?
- Qcow2 belegt initial „quasi keinen“ Plattenplatz
- Daher auch „quasi keine“ Änderung beim Löschen
- Echter Bedarf ist im Beispiel  $844 \cdot 1024$  Bytes
- Siehe Screenshot (nächste Folie):

- Screenshot:

```

root@roland-Lenovo-Y50-70:~# df -h /
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/sda9_crypt  59G   37G   19G   67% /
root@roland-Lenovo-Y50-70:~# ls -l /var/lib/libvirt/images/
insgesamt 1688
-rw----- 1 libvirt-qemu kvm 4295884800 Dez  8 16:22 alpinelinux3.8-2.qcow2
-rw----- 1 libvirt-qemu kvm 4295884800 Dez  8 15:41 alpinelinux3.8.qcow2
root@roland-Lenovo-Y50-70:~# ls -l /var/lib/libvirt/images/
insgesamt 844
-rw----- 1 libvirt-qemu kvm 4295884800 Dez  8 16:22 alpinelinux3.8-2.qcow2
root@roland-Lenovo-Y50-70:~# df -h /
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/sda9_crypt  59G   37G   19G   67% /
root@roland-Lenovo-Y50-70:~# df -h /
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/sda9_crypt  59G   37G   19G   67% /
root@roland-Lenovo-Y50-70:~# ls -lsk /var/lib/libvirt/images/
insgesamt 844
844 -rw----- 1 libvirt-qemu kvm 4295884800 Dez  8 16:22 alpinelinux3.8-2.qcow2
root@roland-Lenovo-Y50-70:~# stat /var/lib/libvirt/images/alpinelinux3.8-2.qcow2
  Datei: /var/lib/libvirt/images/alpinelinux3.8-2.qcow2
  Größe: 4295884800   Blöcke: 1688      EA Block: 4096   Normale Datei
Gerät: fd00h/64768d   Inode: 1572953   Verknüpfungen: 1
Zugriff: (0600/-rw-----)  Uid: (64055/libvirt-qemu)  Gid: ( 132/   kvm)
Zugriff: 2023-12-08 16:22:25.753740055 +0100
Modifiziert: 2023-12-08 16:22:24.713735912 +0100
Geändert: 2023-12-08 16:22:25.717739912 +0100
Geburt: -
root@roland-Lenovo-Y50-70:~#

```

- Anderes Beispiel:

```
root@roland-Lenovo-Y50-70:~# df -h /
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/sda9_crypt  59G   39G   18G   69% /
root@roland-Lenovo-Y50-70:~# cd /var/lib/libvirt/images/
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# ls -ls
insgesamt 898996
898996 -rw----- 1 root root 21478375760 Dez  8 18:25 vm1.qcow2
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# ls -lsh
insgesamt 878M
878M -rw----- 1 root root 21G Dez  8 18:25 vm1.qcow2
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# ls -lsk
insgesamt 898996
898996 -rw----- 1 root root 21478375760 Dez  8 18:25 vm1.qcow2
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# stat vm1.qcow2
  Datei: vm1.qcow2
  Größe: 21478375760      Blöcke: 1797992      EA Block: 4096      Normale
Gerät: fd00h/64768d      Inode: 1583317      Verknüpfungen: 1
Zugriff: (0600/-rw-----) Uid: (  0/   root)  Gid: (  0/
Zugriff: 2023-12-09 10:00:59.309578617 +0100
Modifiziert: 2023-12-08 18:25:53.536368631 +0100
Geändert: 2023-12-08 19:26:19.499831481 +0100
Geburt: -
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images#
```

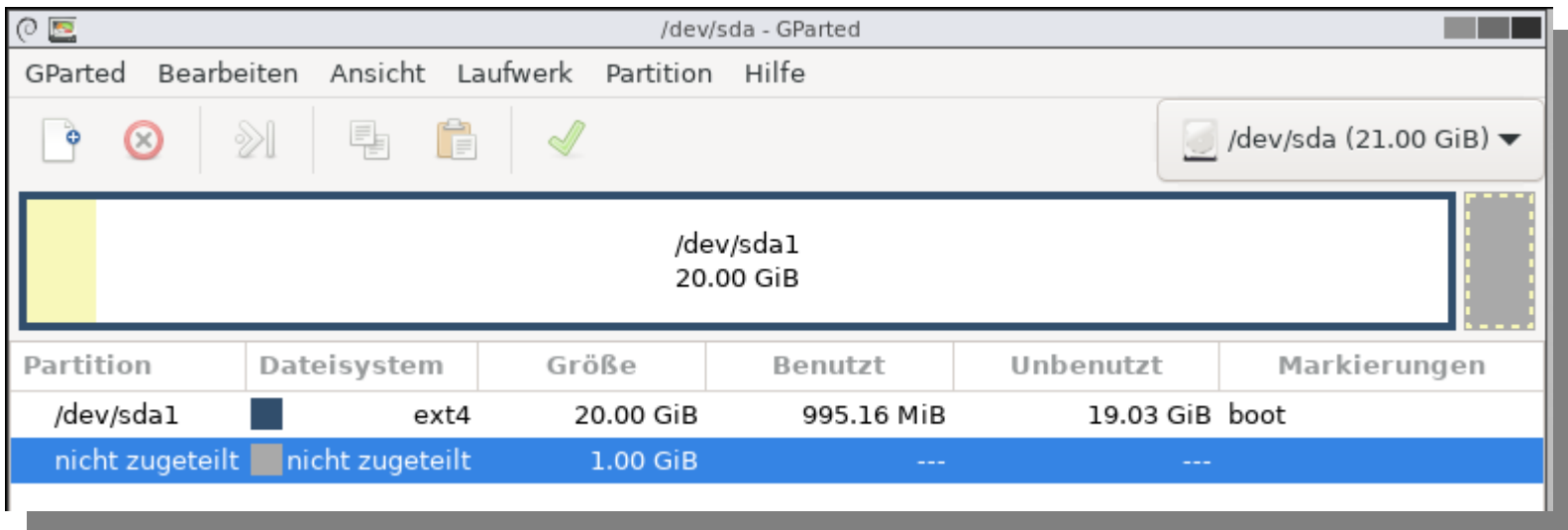
# Dateiformat

- Kann recht einfach erweitert werden
  - `qemu-img resize alpinelinux3.8-2.qcow2 +1G`
- Braucht erst mal nicht mehr Platz

```
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# df -h /
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/sda9_crypt  59G   39G   18G   69% /
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# ls -lsk
insgesamt 898992
898992 -rw----- 1 root root 21478375424 Dez  8 18:24 vm1.qcow2
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# qemu-img resize vm1.qcow2 +1G
Image resized.
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# ls -lsk
insgesamt 898996
898996 -rw----- 1 root root 21478375760 Dez  8 18:25 vm1.qcow2
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# qemu-img resize vm1.qcow2 +1G
qemu-img: Could not open 'vm1.qcow2': Failed to get "write" lock
Is another process using the image [vm1.qcow2]?
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images# stat vm1.qcow2
  Datei: vm1.qcow2
  Größe: 21478375760   Blöcke: 1798000   EA Block: 4096   Normale Datei
Gerät: fd00h/64768d   Inode: 1583317   Verknüpfungen: 1
Zugriff: (0600/-rw-----)  Uid: (64055/libvirt-qemu)  Gid: ( 132/   kvm)
Zugriff: 2023-12-08 18:26:15.564509111 +0100
Modifiziert: 2023-12-08 18:25:53.536368631 +0100
Geändert: 2023-12-08 18:26:15.540508958 +0100
Geburt: -
root@roland-Lenovo-Y50-70:/var/lib/libvirt/images#
```

# Dateiformat

- Man sieht die Erweiterung, wenn man von einem „gparted“ ISO-Image startet



Partition	Dateisystem	Größe	Benutzt	Unbenutzt	Markierungen
/dev/sda1	ext4	20.00 GiB	995.16 MiB	19.03 GiB	boot
nicht zugeteilt	nicht zugeteilt	1.00 GiB	---	---	

## Dateiformat

- Man kann eine qcow2-Datei nach längerem Gebrauch wieder kleiner machen
- Siehe [shrink]

# Konvertierung aus anderen Dateiformaten

- Es existieren eine Fülle von Kommandozeilenbefehlen
- Beispiel (vom Vmware-Format nach „raw“):
  - `qemu-img convert -f vmdk source.vmdk -O raw destination.img`
- Geht auch umgekehrt:
  - `qemu-img convert -f raw -O vmdk vm1.qcow2 vm1.vmdk`
- Beispiele aus dem Netz
- Nicht selbst ausprobiert
- Siehe [convert], [convert2] und [convert3]



## Ausblick

- Virtuelle Maschine kann auf einem Server laufen
- Grafische Oberfläche auf dem Client
- Kann „Thin Client“ sein
- Optimierung der Performance durch Benutzung von „virtio“-Treibern (Paravirtualisierung)

Vielen Dank!

# Quellen

- [KVM]: [https://linux-kvm.org/page/Main\\_Page](https://linux-kvm.org/page/Main_Page)
- [QEMU]: <https://www.qemu.org/>
- [libvirt]: <https://libvirt.org/>
- [libvirtd]: <https://libvirt.org/manpages/libvirtd.html>
- [virt-manager]: <https://virt-manager.org/>
- [KVM2]: [https://de.wikipedia.org/wiki/Kernel-based\\_Virtual\\_Machine](https://de.wikipedia.org/wiki/Kernel-based_Virtual_Machine)
- [QEMU2]: <https://de.wikipedia.org/wiki/QEMU>
- [virt-manager2]:  
[https://de.wikipedia.org/wiki/Virtual\\_Machine\\_Manager](https://de.wikipedia.org/wiki/Virtual_Machine_Manager)
- [sparse]: <https://de.wikipedia.org/wiki/Sparse-Datei>

# Quellen

- [shrink]: <https://kofler.info/wie-ich-ein-qcow2-image-auf-ein-drittel-geschrumpft-habe/>
- [shrink2]: [https://pve.proxmox.com/wiki/Shrink\\_Qcow2\\_Disk\\_Files](https://pve.proxmox.com/wiki/Shrink_Qcow2_Disk_Files)
- [convert]: <https://docs.openstack.org/de/image-guide/convert-images.html>
- [convert2]: <https://askubuntu.com/questions/4398/how-do-i-convert-a-multiple-part-vmdk-disk-image-to-qcow2>
- [convert3]: <https://serverfault.com/questions/815224/converting-multiple-vmware-disk-image-to-single-disk-image>

# Quellen

- [arch]: <https://forum.endeavouros.com/t/tutorial-how-to-install-virt-manager-correctly-on-eos-arch-base-systems/17136>
- [ubuntu]: <https://ubuntu.com/blog/kvm-hypervisor>
- [redhat]:  
[https://access.redhat.com/documentation/de-de/red\\_hat\\_enterprise\\_linux/6/html/virtualization\\_host\\_configuration\\_and\\_guest\\_installation\\_guide/chap-virtualization\\_host\\_configuration\\_and\\_guest\\_installation\\_guide-guest\\_installation\\_virt\\_manager-creating\\_guests\\_with\\_virt\\_manager](https://access.redhat.com/documentation/de-de/red_hat_enterprise_linux/6/html/virtualization_host_configuration_and_guest_installation_guide/chap-virtualization_host_configuration_and_guest_installation_guide-guest_installation_virt_manager-creating_guests_with_virt_manager)

## Quellen

- [debian]: <https://wiki.debian.org/KVM>
- [X11]: <https://www.simplified.guide/ssh/x11-forwarding-as-root>