

Virtuelle Maschinen mit KVM

Bernd Strößenreuther

linux-cafe@stroessenreuther.net

Linux-Cafe 2022-07-06

Lizenz

Sie dürfen die Text-Inhalte dieses Dokument verwenden unter den Bedingungen der Creative Commons Lizenz:

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

Herkunft der verwendeten Bilder, Icons und Logos siehe jeweils direkt an der entsprechenden Stelle im Dokument.

Die Urheberrechte daran liegen beim Autor.

Alle Icons ohne dedizierte Quellenangabe stammen von

<https://freesvg.org/>

Lizenz: Public Domain

Agenda

- 1) Wieso? Weshalb? Warum?
- 2) Live Demo
- 3) Separates Netzwerksegment für VMs
- 4) Fertige Images importieren
- 5) Eure Fragen

Häufige Einsatzzwecke für VMs

Am Desktop

- Einzelne Windows Anwendungen werden auf einem Linux Desktop gebraucht
- Ich will mal schnell was ausprobieren
 - Unter einem anderen Betriebssystem
 - Gleiches Betriebssystem, aber ich will die Testumgebung schnell und rückstandsfrei wieder entfernen
- Ideale Ergänzung für diesen Anwendungsfall: Vagrant (<https://www.vagrantup.com/>)
- Am Entwickler-Notebook als Test-Umgebung

Häufige Einsatzzwecke für VMs

Am Server

- Mandanten-Trennung
- Trennung verschiedener Sicherheitszonen
- Bessere Hardware-Auslastung

Im Vergleich: VirtualBox

- VirtualBox ist stark am Desktop
 - Open-Core-Modell: Basisfunktionalität unter GPL
 - Extension Pack (USB 2.0 + 3.0 devices, VirtualBox RDP, disk encryption, NVMe, PXE boot):
Proprietär, aber kostenlos für „Personal Use and Evaluation“
- Die Oberfläche der VM wird (im Standard-Fall) in einem Fenster angezeigt und kann dort bedient werden
- 2007 von der deutschen Firma InnoTek entwickelt
- 2008 von Sun übernommen / inzwischen Oracle

Im Vergleich: KVM

- Kernel-based Virtual Machine
- Eine Infrastruktur im Linux-Kernel, um die Virtualisierungstechniken der Hardware (Intel-VT, AMD-V, etc.) nutzbar zu machen
- VMs laufen standardmäßig im Hintergrund (als Daemon)
- Wird gerade im Server-Bereich oft bevorzugt
- Lizenz: GPL / LGPL

Welchen nehme ich am besten?

KVM hat Vorteile

- Am Server
- Wenn ich Wert lege auf vollständig freie Software

VirtualBox hat Vorteile

- Am Desktop, wenn ich meistens die grafische Oberfläche des Gastsystems bedienen will

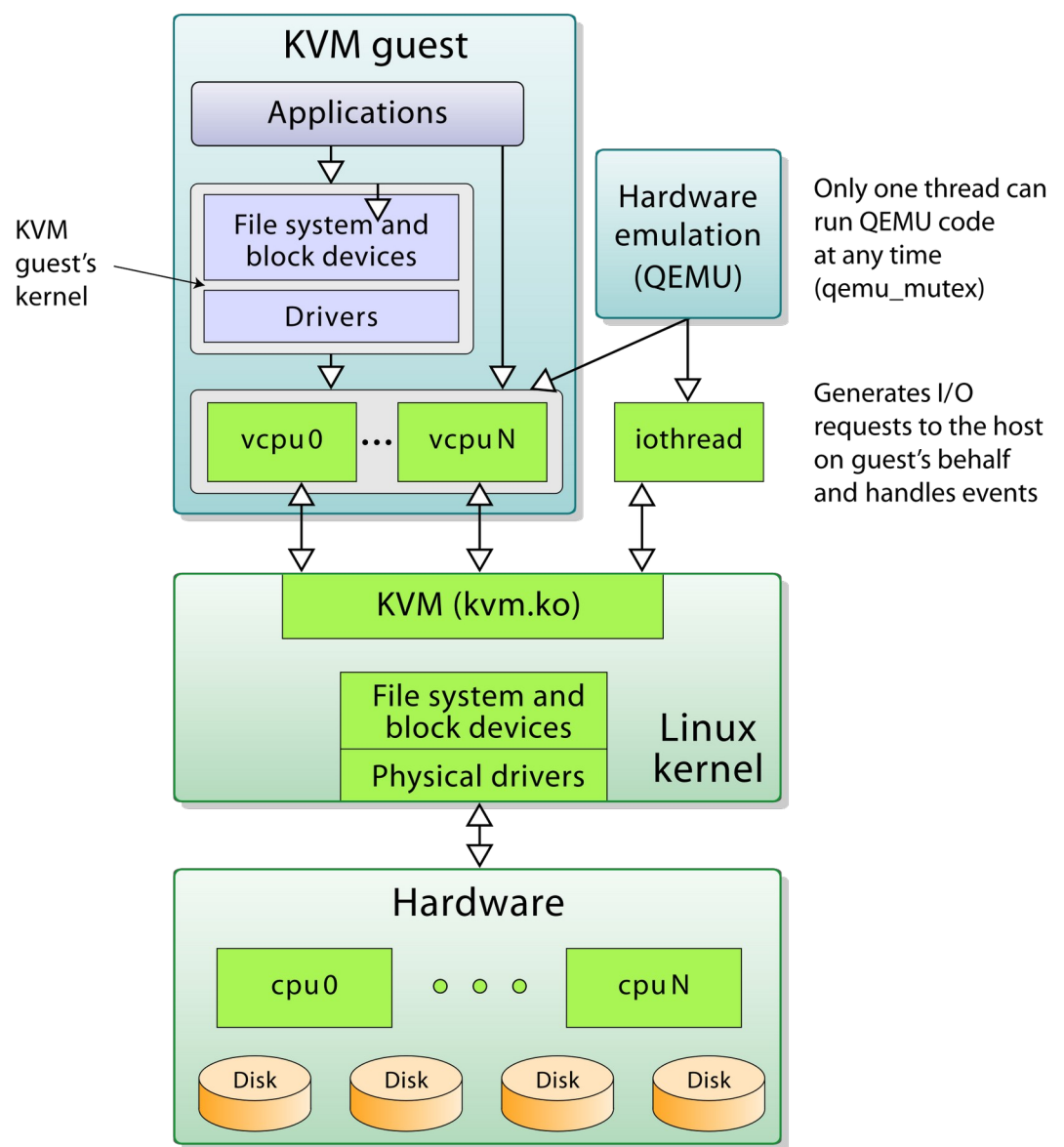
Beide sind mehr oder weniger **gleichauf**

- Wenn ich hauptsächlich Test-Umgebungen (mit verschiedenen Betriebssystemen) brauche
- Insbesondere wenn ich diese mit Vagrant erzeuge

KVM

- Seit Version 2.6.20 im Linux-Kernel enthalten
- QEMU stellt der VM emulierte Geräte zur Verfügung
- Paravirtualisierung mit Virtio:
Festplatten und Netzwerk-Devices

Aufbau KVM



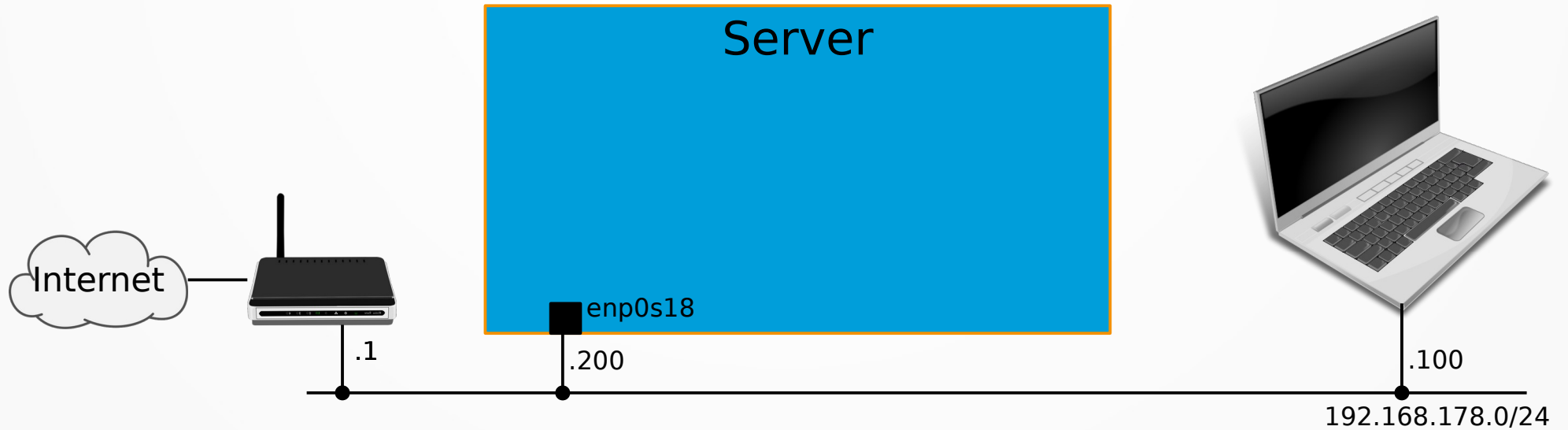
Quelle:

https://commons.wikimedia.org/wiki/File:Kernel-based_Virtual_Machine.svg

Live-Demo

frei nach <https://wiki.debian.org/KVM>

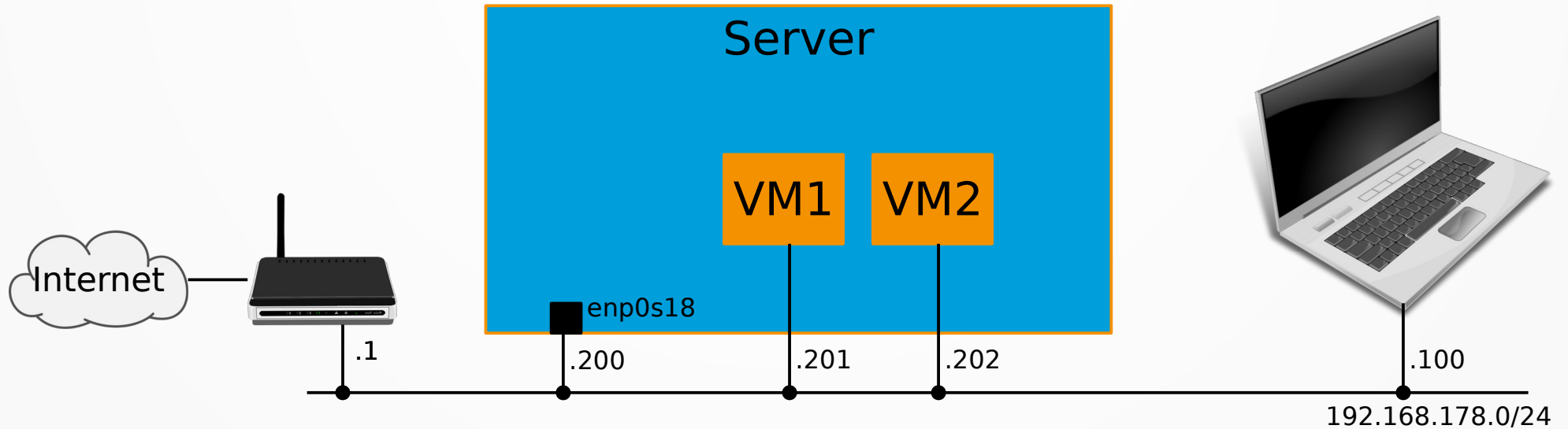
Setup ohne VMs



Live-Demo

- <https://stroessenreuther.info/pub/videos/kvm-livedemo-1920.mp4>
(MP4, 1920x1200, 70 MB)
oder
- <https://stroessenreuther.info/pub/videos/kvm-livedemo-1920.webm>
(webm, 1920x1200, 74 MB)

Setup bisher: flaches Netz

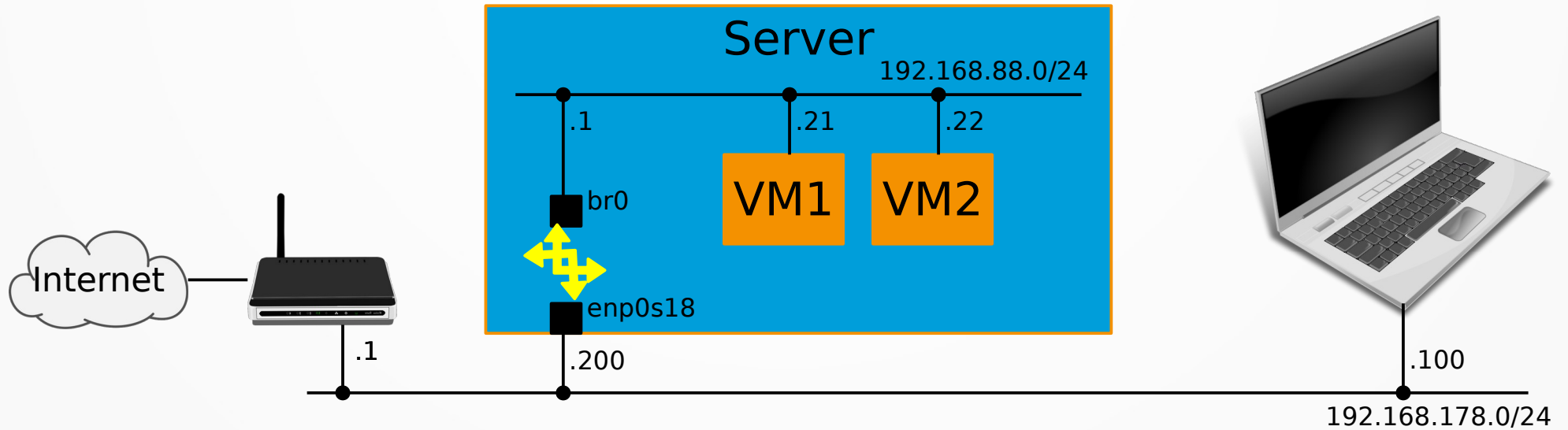


/etc/network/interfaces

```
auto enp0s18
iface enp0s18 inet manual
iface enp0s18 inet6 manual
```

```
auto br0
iface br0 inet static
    address 192.168.178.200
    netmask 255.255.255.0
    network 192.168.178.0
    broadcast 192.168.178.255
    gateway 192.168.178.1
    bridge_ports enp0s18
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
    dns-nameservers 192.168.178.1
iface br0 inet6 static
    address 2001:db8:cafe:178::1
    netmask 64
    accept_ra 0
```

Separates Netz für VMs





/etc/network/interfaces

```
auto enp0s18
iface enp0s18 inet static
    address 192.168.178.200
    netmask 255.255.255.0
    gateway 192.168.178.1
iface enp0s18 inet6 static
    address 2001:db8:cafe:178::1
    netmask 64
    accept_ra 0
```

```
auto br0
iface br0 inet static
    address 192.168.88.1
    netmask 255.255.255.0
    bridge_ports none
    bridge_fd 9
    bridge_hello 2
    bridge_maxage 12
    bridge_stp off
iface br0 inet6 static
    address 2001:db8:cafe:88::1
    netmask 64
    accept_ra 0
```

Routing

- Der Internet-Router muss eine (statische) Route für das Netz 192.168.88.0/24 haben, die auf 192.168.178.200 zeigt

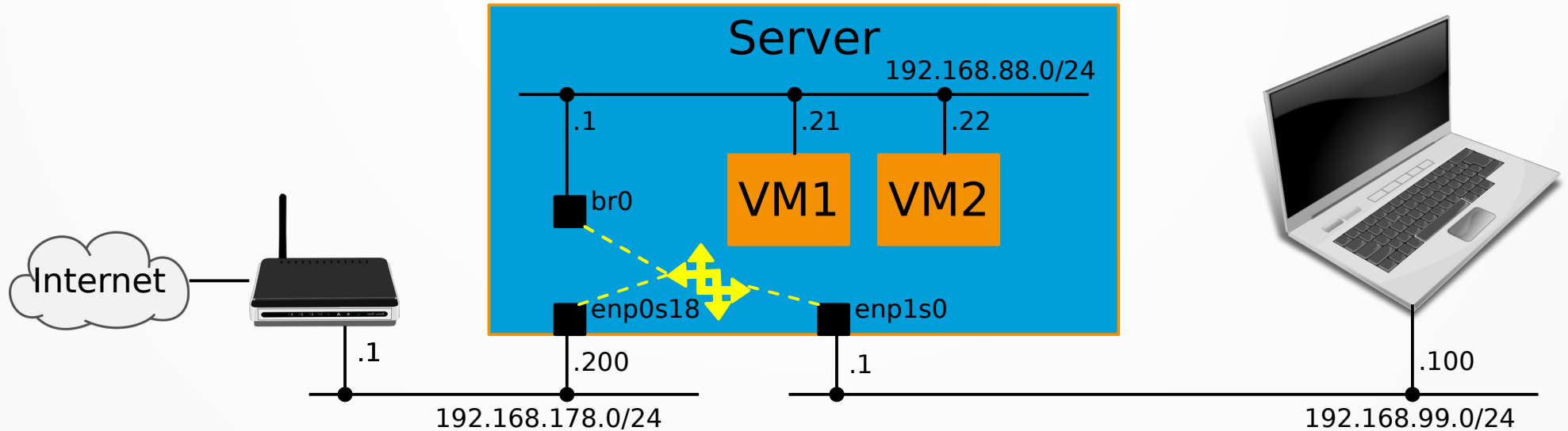
Aktiv	Netzwerk ↕	Subnetzmaske ↕	Gateway ↕	
<input checked="" type="checkbox"/>	192.168.88.0	255.255.255.0	192.168.178.200	 

- Am Server (im Host-Betriebssystem) IP forwarding aktivieren, in /etc/sysctl.conf:

```
net.ipv4.ip_forward=1
```

```
net.ipv6.conf.all.forwarding=1
```

Komplette Netzwerktrennung



Fertiges qcow2 File nutzen

- z. B. beim Umzug einer VM auf einen anderen Virtualisierungshost

```
virt-install --virt-type kvm --name numerobis  
  --os-variant debian10  
  --disk path=/opt/kvm/numerobis.qcow2.img  
  --memory 2048 --network bridge=br0  
  --graphics vnc,port=5922  
  --boot hd --autostart
```

Noch Fragen?

- Jetzt und hier
- Nachher beim Bier
- Bei (fast) jedem Linux-Cafe
- Jederzeit auf der Gluga Users Mailingliste, siehe <https://mailing.gluga.de/>

