

# Linux 2 DEVOPS 2020

## Lektion 9

# Idag

- Virtuella servrar 2
- KVM
- Bygga upp miljö med virtuella servrar på en egen server

# Virtuella servrar

- Bygga upp ett antal virtuella servrar på en fysisk server
- I princip hela serverparken på en maskin
- En hel testmiljö på en maskin, spridd över olika servrar
- ...eller en av flera fysiska maskiner som används för en uppsättning virtuella servrar

# KVM (Kernel Virtual Machine)

"KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, `kvm.ko`, that provides the core virtualization infrastructure and a processor specific module, `kvm-intel.ko` or `kvm-amd.ko`.

Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

KVM is open source software. The kernel component of KVM is included in mainline Linux, as of 2.6.20. The userspace component of KVM is included in mainline QEMU, as of 1.3."

([www.linux-kvm.org](http://www.linux-kvm.org))

# KVM

- En hypervisor inbyggd i Linux
- Program på Linux
- I de virtuella maskinerna kan olika operativsystem installeras

# KVM

- Kort introduktion på Youtube

<https://www.youtube.com/watch?v=Pgltb5InnLY>

# KVM

- qemu för att sköta kontakten med värdmaskinens hårdvara
- virsh för kommandoraden
- virt-manager för GUI

# KVM

- Kvm check
  - Paket cpu-checker
  - Kommando kvm-ok

- Installera paket

```
sudo apt-get install qemu-kvm libvirt-bin virtinst  
bridge-utils virt-manager
```

- virsh



# Övning 1

- Kör `kvm-ok`
  - Om du inte har själva kommandot `kvm-ok`, behöver du installera paketet `cpu-checker`
- Kan din maskin köra KVM?

# Förebereedelse för kommande övningar

- Om du inte råkar ha någon iso-image liggande på din hårddisk, börja ladda hem en nu så att den finns redo om en stund. (Använd t ex wget.)
- Några relativt små images:
  - CentOS8 minimal  
`http://ftp.lysator.liu.se/pub/CentOS/8.2.2004/isos/x86\_64/CentOS-8.2.2004-x86\_64-minimal.iso`
  - TinyCore  
`http://tinycorelinux.net/11.x/x86/release/Core-current.iso`

# virsh-kommandon

```
virsh list
```

```
virsh list --all
```

```
virsh net-list
```

```
virsh net-info default
```

```
virsh nodeinfo
```

# KVM bibliotek

- `/var/lib/libvirt/`
- ISO images: `/var/lib/libvirt/boot/`
- VM-installationer: `/var/lib/libvirt/images/`
- Libvirt-konfiguration: `/etc/libvirt/`

# Övning 2

- Installera paketen för att använda KVM
- Se att ni får en (tom) lista med `virsh list`
- Se att ni får en (tom) lista med `virsh net-list`
- Se nätverksinformation med `virsh net-info default`

# Övning 2

```
mo@sputnik:~$ kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
mo@sputnik:~$
mo@sputnik:~$ virsh list

```

Id	Name	State
-----		

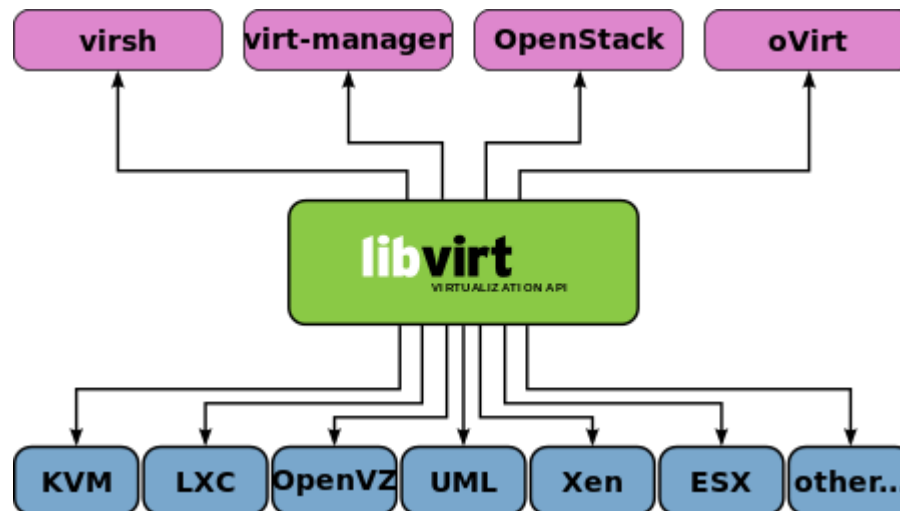
```
mo@sputnik:~$ virsh net-list

```

Name	State	Autostart	Persistent
-----			
default	active	yes	yes

```
mo@sputnik:~$ virsh net-info default
Name:          default
UUID:          2985b4e5-d9cb-49d2-81bb-f927fb89ba51
Active:        yes
Persistent:    yes
Autostart:     yes
Bridge:        virbr0
```

# libvirt finns även i andra miljöer



# KVM: bygga en virtuell maskin

- Börja med att hämta en iso, lämpligen för någon minimerad Linux och kopiera den till `/var/lib/libvirt/boot/`

Exempel:

```
cd /var/lib/libvirt/boot/
```

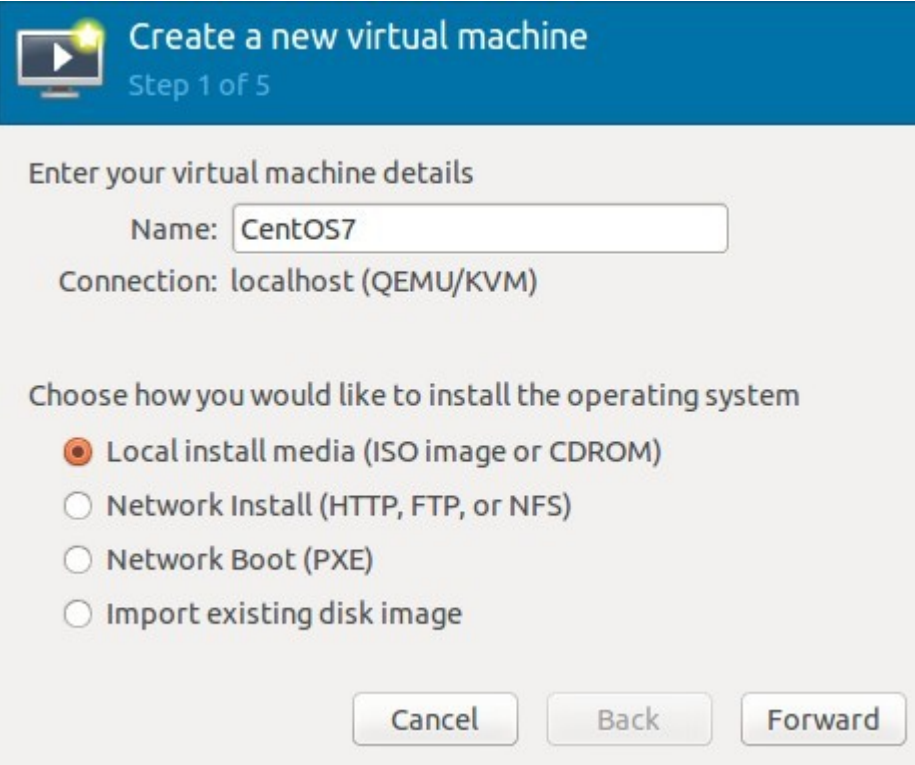
```
sudo wget
```

```
http://ftp.lysator.liu.se/pub/CentOS/8.2.2004/isos/x86_64/CentOS-8.2.2004-x86_64-minimal.iso
```

- Installera med virt-manager (GUI) eller virt-install (CLI)



# virt-manager



The screenshot shows the 'Create a new virtual machine' window in virt-manager. The title bar is blue with a play button icon and the text 'Create a new virtual machine' and 'Step 1 of 5'. The main area is light gray. It has a section 'Enter your virtual machine details' with a text input for 'Name' containing 'CentOS7' and a label 'Connection: localhost (QEMU/KVM)'. Below this is a section 'Choose how you would like to install the operating system' with four radio button options: 'Local install media (ISO image or CDROM)' (selected), 'Network Install (HTTP, FTP, or NFS)', 'Network Boot (PXE)', and 'Import existing disk image'. At the bottom are three buttons: 'Cancel', 'Back', and 'Forward'.

Create a new virtual machine  
Step 1 of 5

Enter your virtual machine details

Name:

Connection: localhost (QEMU/KVM)

Choose how you would like to install the operating system

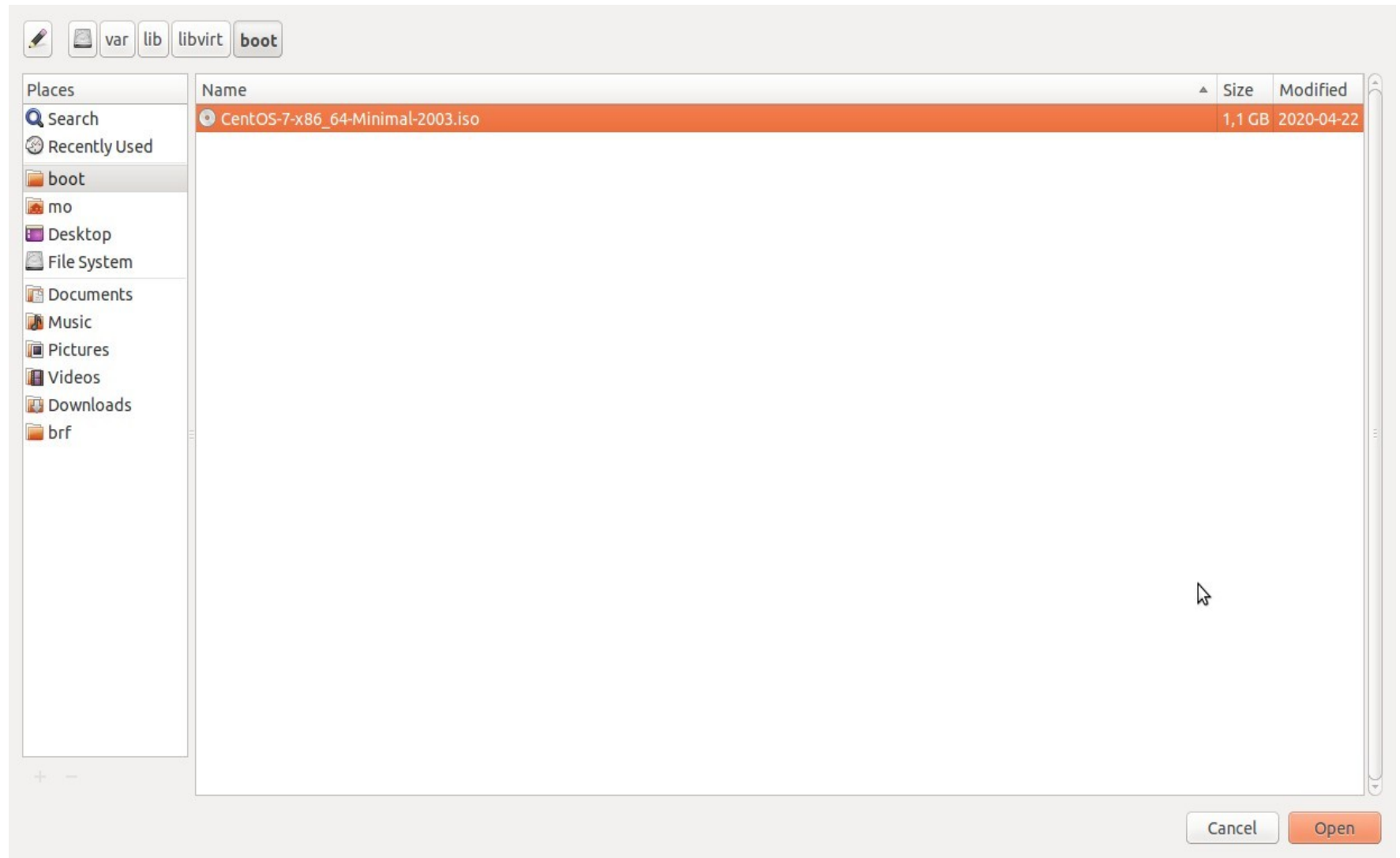
☒ Local install media (ISO image or CDROM)

☐ Network Install (HTTP, FTP, or NFS)

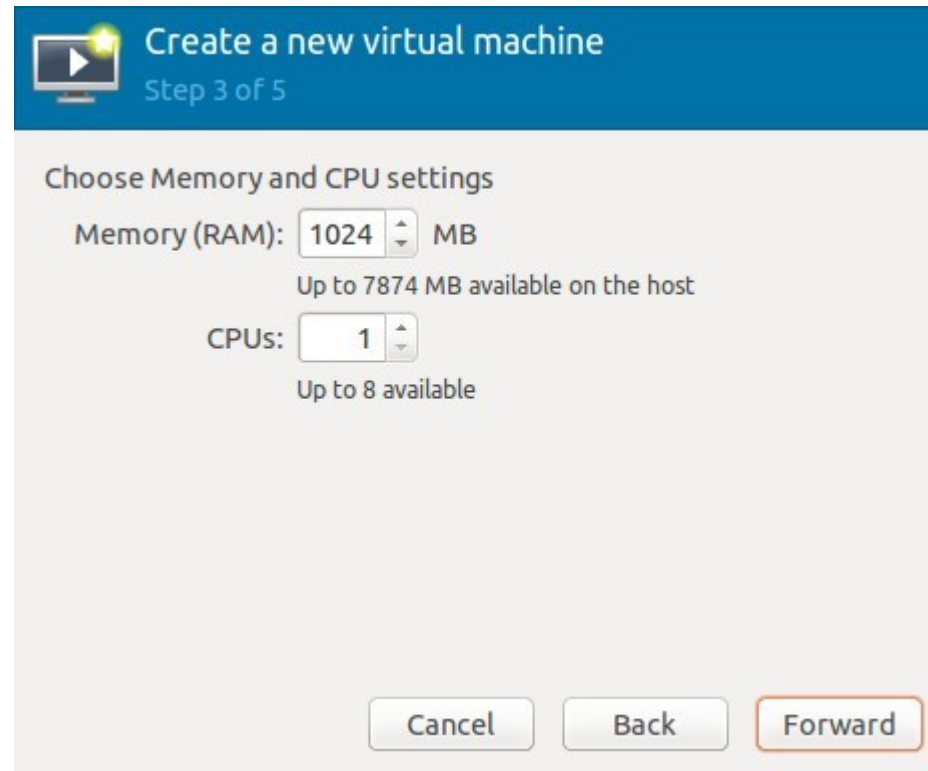
☐ Network Boot (PXE)

☐ Import existing disk image

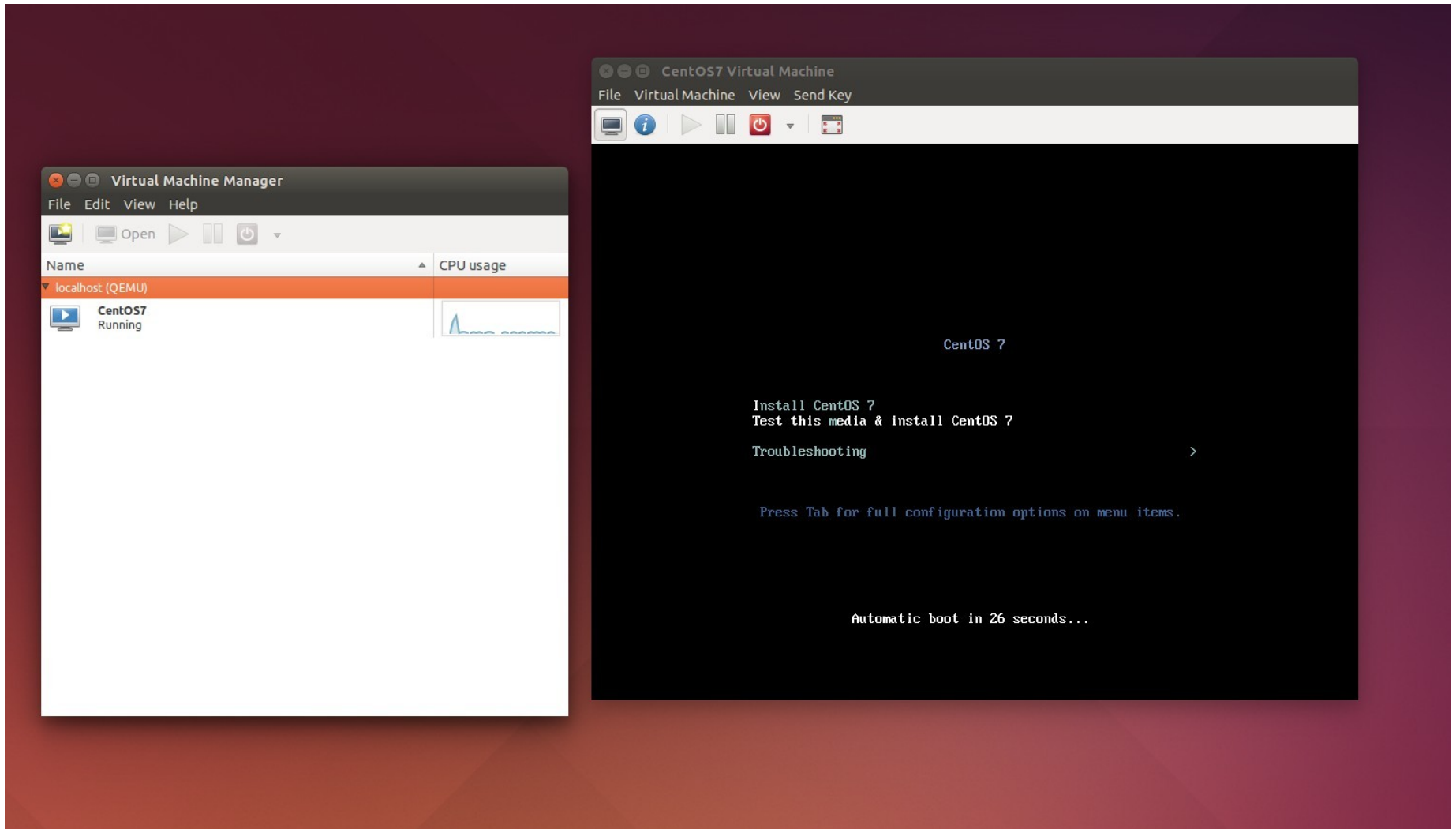
# virt-manager



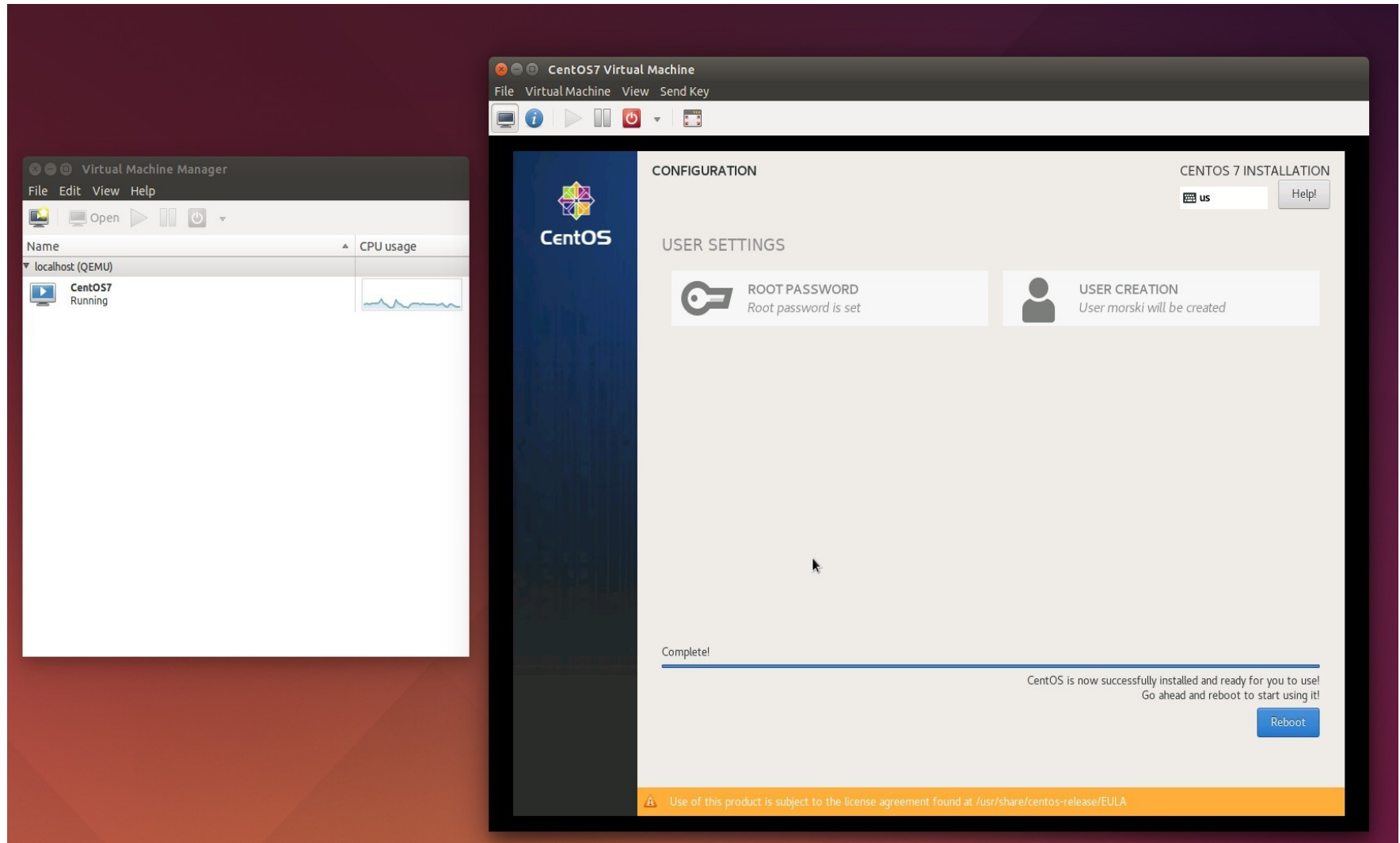
# virt-manager



# virt-manager



# virt-manager



# virt-install, exempel 1

```
sudo virt-install \  
--name centos7 \  
--description "Test VM with CentOS 7" \  
--ram=1024 \  
--vcpus=2 \  
--os-type=Linux \  
--os-variant=rhel7 \  
--disk path=/var/lib/libvirt/images/centos7.qcow2,bus=virtio,size=10 \  
--graphics none \  
--location $HOME/iso/CentOS-7-x86_64-Everything-1611.iso \  
--network bridge:virbr0 \  
--console pty,target_type=serial -x 'console=ttyS0,115200n8 serial'
```

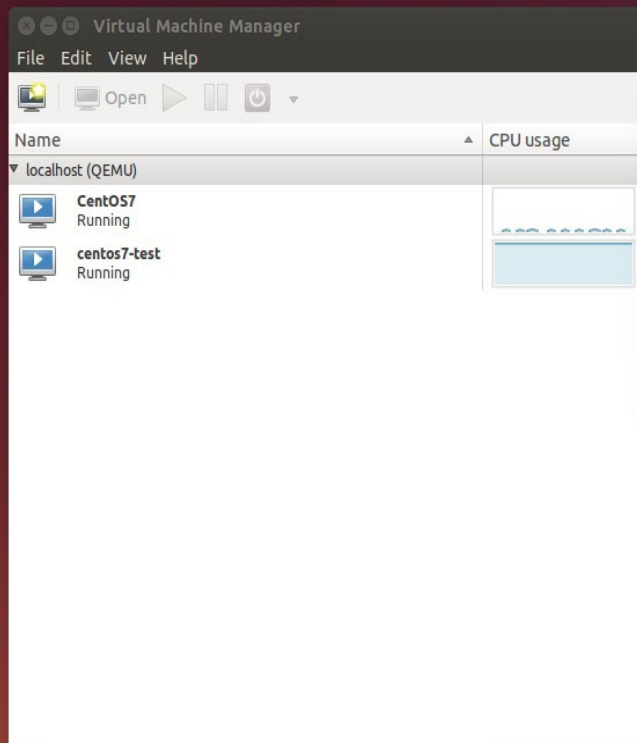
Exempel stulet från

<https://computingforgeeks.com/virsh-commands-cheatsheet/>

# virt-install, exempel 2

```
sudo virt-install \  
--name centos7 \  
--description "Test VM CentOS 7 minimal" \  
--ram=1024 \  
--vcpus=1 \  
--os-type=Linux \  
--os-variant=rhel7 \  
--disk path=/var/lib/libvirt/images/centos7.img,bus=virtio,size=10 \  
--graphics none \  
--cdrom=/var/lib/libvirt/boot/CentOS-7-x86_64-Minimal-2003.iso \  
--network bridge:virbr0 \  
--console pty,target_type=serial -x 'console=ttyS0,115200n8 serial'
```

# virt-install



```
mo@sputnik: /var/lib/libvirt/boot
mo@sputnik:/var/lib/libvirt/boot$ sudo virt-install --name centos7-test --description "Test VM CentOS 7 minimal" --ram=1024 --vcpus=1 --os-type=Linux --os-variant=rhel7 --disk path=/var/lib/libvirt/images/centos7test.img,bus=virtio,size=10 --graphics none --cdrom=/var/lib/libvirt/boot/CentOS-7-x86_64-Minimal-2003.iso --network bridge:virbr0 \
>
[sudo] password for mo:

Starting install...
Allocating 'centos7test.img'          | 10 GB   00:00
Creating domain...                    |  0 B    00:00
Connected to domain centos7-test
Escape character is ^]
```



# Fler virsh-kommandon

```
virsh start <vm>
```

```
virsh shutdown <vm>
```

```
virsh destroy <vm>
```

```
virsh reboot <vm>
```

```
virsh dominfo <vm>
```

```
virsh undefine <vm>
```

# Övning 3

- Installera en virtuell maskin på KVM.
- Kontrollera att du kan starta och starta om din virtuella maskin.

# Övning 3

```
virsh start CentOS7
```

```
virsh shutdown CentOS7
```

```
virsh start CentOS7
```

```
virsh reboot CentOS7
```

```
virsh dominfo CentOS7
```

# virt-clone

- "Clone", skapa en kopia av en virtuell server med `virt-clone`
- Maskinen som skall klonas behöver vara avstängd under kloningen
- Syntax:

```
virt-clone --original {Domain-Vm-Name-Here} --  
name {New-Domain-Vm-Name-Here} --auto-clone
```

# Övning 4

- Klona den virtuella maskin du skapade i den förra övningen

# Övning 4

```
sudo virt-clone --original CentOS7 --name  
CentOS7-2 --auto-clone  
virsh start CentOS7-2
```

# Kopiera till en annan fysisk server

Inte en lika automatisk kloning som lokalt, men med enkla steg:

1. Stäng ned servern som skall kopieras.
2. Kopiera image-filen från `/var/lib/libvirt/images`
3. Plocka ut vm-definitionerna i en XML-fil

```
virsh dumpxml VMNAME > domxml.xml
```

4. Läs in vm-definitionerna på den nya servern

```
virsh define domxml.xml
```

# Nätverk mellan virtuella maskiner

- Default-uppsättning i kvm ger varje virtuell maskin en adress på nätet 192.168.122.0/24
- Värddmaskinen kan nås på 192.168.122.1
- Notera att detta gäller för nätverksuppsättningen "hypervisor default"
- Notera NAT-inställningar som har skapats i iptables



# Övning 5

- Titta på dina virtuella maskiners nätverksinställningar och hitta vilka adresser de har fått
- Testa förbindelsen mellan dina två virtuella maskiner (förslagsvis med ping)
- Testa förbindelsen till 192.168.122.1
- Ta en titt på vad som hänt i iptables (på värdmaskinen)
- Ta en titt på nätverksinterface vibr0 (på värdmaskinen)

# Övning 5

```
mo@sputnik:~/nackademin/Linux2$ sudo iptables -n -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
RETURN     all  --  192.168.122.0/24      224.0.0.0/24
RETURN     all  --  192.168.122.0/24      255.255.255.255
MASQUERADE tcp  --  192.168.122.0/24      !192.168.122.0/24      masq ports: 1024-655
35
MASQUERADE udp  --  192.168.122.0/24      !192.168.122.0/24      masq ports: 1024-655
35
MASQUERADE all  --  192.168.122.0/24      !192.168.122.0/24
mo@sputnik:~/nackademin/Linux2$
```

```
virbr0    Link encap:Ethernet  HWaddr fe:54:00:14:0a:96
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3466 (3.4 KB)  TX bytes:3029 (3.0 KB)
```

# För nätverk utifrån: bridge

- Skapa en brygga (bridge) för att göra virtuella maskiner tillgängliga genom nätverket
- Använder sig av värdmaskinens nätverk
- Fungerar inte med wlan
- Definiera bryggan i `/etc/network/interfaces`
- Starta om networking
- Använd bryggan för att ange nätverkskoppling i nätverkskonfigurationen för virtuella maskiner (istället för "default")

# För nätverk utifrån: bridge

Exempel med dhcp:

```
auto br0
```

```
iface br0 inet dhcp
```

```
    bridge_ports eth0
```

```
    bridge_stp off
```

```
    bridge_fd 0
```

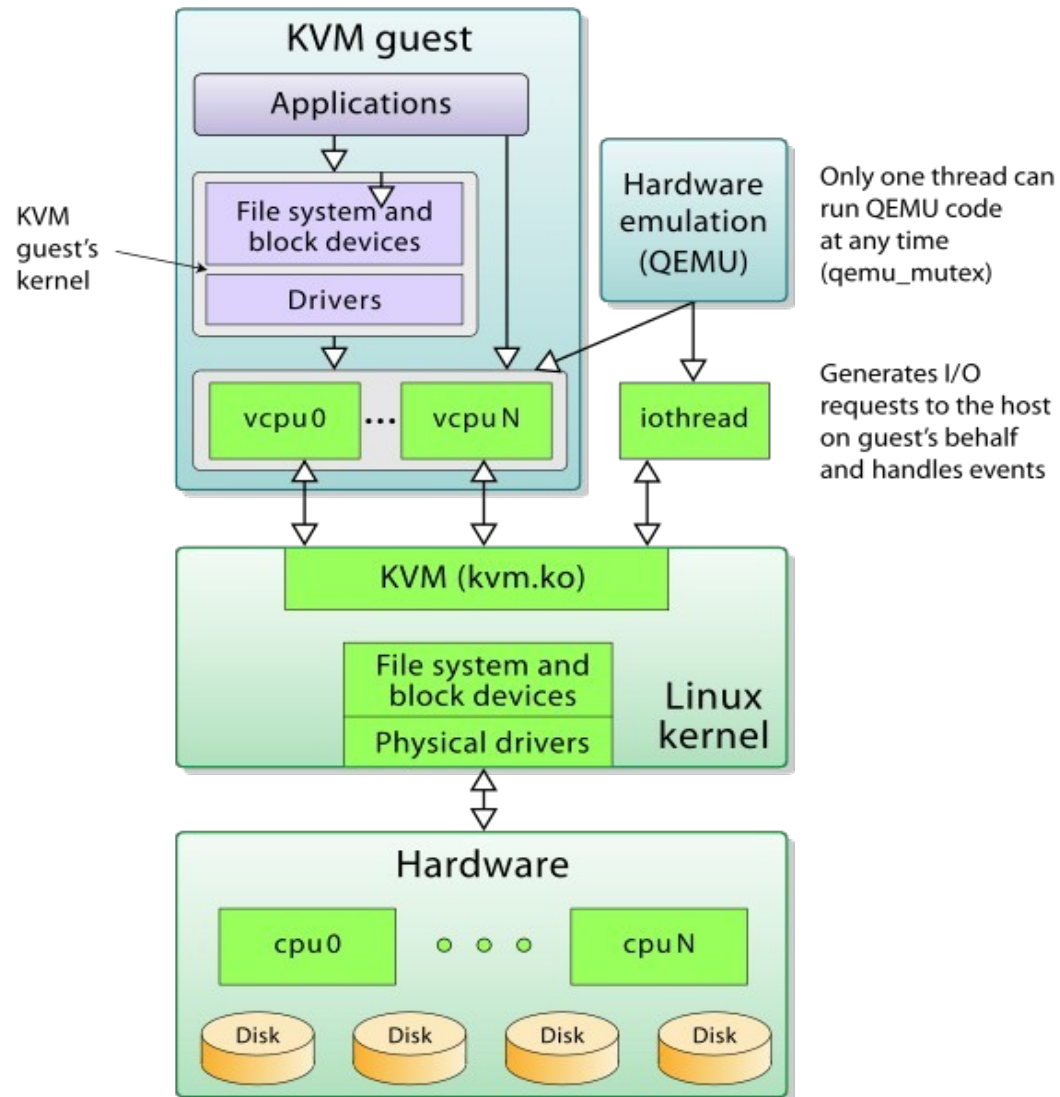
```
    bridge_maxwait 0
```

# För nätverk utifrån: bridge

Exempel med fast adress:

```
auto br0
iface br0 inet static
    address 192.168.0.10
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    dns-nameservers 192.168.0.5 8.8.8.8
    dns-search example.com
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

# KVM



# Tillbakablick, reflektion, kommentarer ...