KVM

Installation

Install all the required for the installation of Qemu, KVM hypervisor, and Libvirt

```
# apt install qemu-system libvirt-daemon-system qemu-utils virt-manager
```

- qemu-system: is an open source virtualizer that provides hardware emulation for the KVM hypervisor. It acts as a virtual machine monitor together with the KVM kernel modules, and emulates the hardware for a full system such as a PC and its associated peripherals.
- virt-manager: Virt-Manager is a graphical user interface (GUI) tool for managing virtual machines through libvirt-daemon.
- libvirt-daemon-system: provides API libraries that enables GUI apps such as virt-manager to communicate with libvirtd daemon, a system service libvirtd , and a virsh CLI tool
- qemu-utils: Various utilities e.g. for manipulating disk images
- virtinst: Allows to create Virtual Machines (VMs) from the command-line.

Verify that the virtualization daemon, libvritd-daemon, is operating before moving on. Execute the command to achieve this.

```
#  systemctl status libvirtd

Output:

● libvirtd.service - Virtualization daemon
     Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; preset:
enabled)
     Active: active (running) since Sun 2023-08-06 10:57:02 CEST; 1min 30s
ago
TriggeredBy: ● libvirtd-ro.socket
             ● libvirtd-admin.socket
             ● libvirtd.socket
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 7999 (libvirtd)
      Tasks: 19 (limit: 32768)
     Memory: 15.7M
        CPU: 188ms
     CGroup: /system.slice/libvirtd.service
             └─7999 /usr/sbin/libvirtd --timeout 120
Aug 06 10:57:02 pcwerkkamer systemd[1]: Starting libvirtd.service -
Virtualization daemon...
Aug 06 10:57:02 pcwerkkamer systemd[1]: Started libvirtd.service -
Virtualization daemon.
```

Check if libvirtd service will start automatically at boot time.

```
# systemctl is-enabled libvirtd

Output:
enabled
```

qemu-img convert -f vdi -O qcow2 ubuntu.vdi ubuntu.qcow2 If disabled run the following command to have it boot automatically:

```
# systemctl enable --now libvirtd
```

Use the following command to determine whether the KVM modules are loaded:

```
$ lsmod | grep -i kvm
kvm_intel             380928  0
kvm                  1142784  1 kvm_intel
irqbypass              16384  1 kvm
```

Configuration

User priviliges

In order to manage virtual machines as a regular user, that user needs to be added to the libvirt group:

```
# usermod -aG libvirt,kvm USERNAME
```

User-specific and system-wide VMs

By default, if virsh is run as a normal user it will connect to libvirt using qemu:*/session URI string. This URI allows virsh to manage only the set of VMs belonging to this particular user. To manage the system set of VMs (i.e., VMs belonging to root) virsh should be run as root or with qemu:*/system URI:

```
$ virsh --connect qemu:///system list --all
```

To avoid having to use the –connect flag on every command, the URI string can be set in the LIBVIRT_DEFAULT_URI environment variable:

```
$ export LIBVIRT_DEFAULT_URI='qemu:///system'
```

Libvirt default network

If you use libvirt to manage your VMs, libvirt provides a NATed bridged network named “default” that allows the host to communicate with the guests. This network is available only for the system domains (that is VMs created by root or using the qemu:*/system connection URI). VMs using this*

network end up in 192.168.122.1/24 and DHCP is provided to them via dnsmasq. This network is not automatically started. To start it use: virsh –connect=qemu:/system net-start default

To make the default network start automatically use:

```
virsh --connect=qemu:///system net-autostart default
```

Error Access ISO files, etc.

Directory, that libvirt is using for storing/reading ISO and qcow2 files needs to be readable by libvirt-qemu user. In case of an error opening an ISO or image file, the “libvirt-qemu” user does not have access to the file or the directories holding the files. The “libvirt-qemu” user needs **“r+x”** permissions **all the way up the path**. You can check if unix permissions are correct by running a shell under user “libvirt-qemu” and see if you can ls along the path all the way up to the files:

```
$ sudo su -s /bin/bash libvirt-qemu
```

If you can't ls with an account that does have access yields a result with a permission line that contains a dot such as drwxrwxr-x. meaning extended ACL permissions.

Fix 1:

Put the iso or image into one of the pool directories of libvirt manager. E.g. /var/lib/libvirt/images Or any other paths that have full r others rights.

Convert VirtualBox to KVM

```
qemu-img convert -f vdi -O qcow2 virtualbox-name.vdi kvm-name.qcow2
```

Links

- https://wiki.debian.org/KVM#Installation