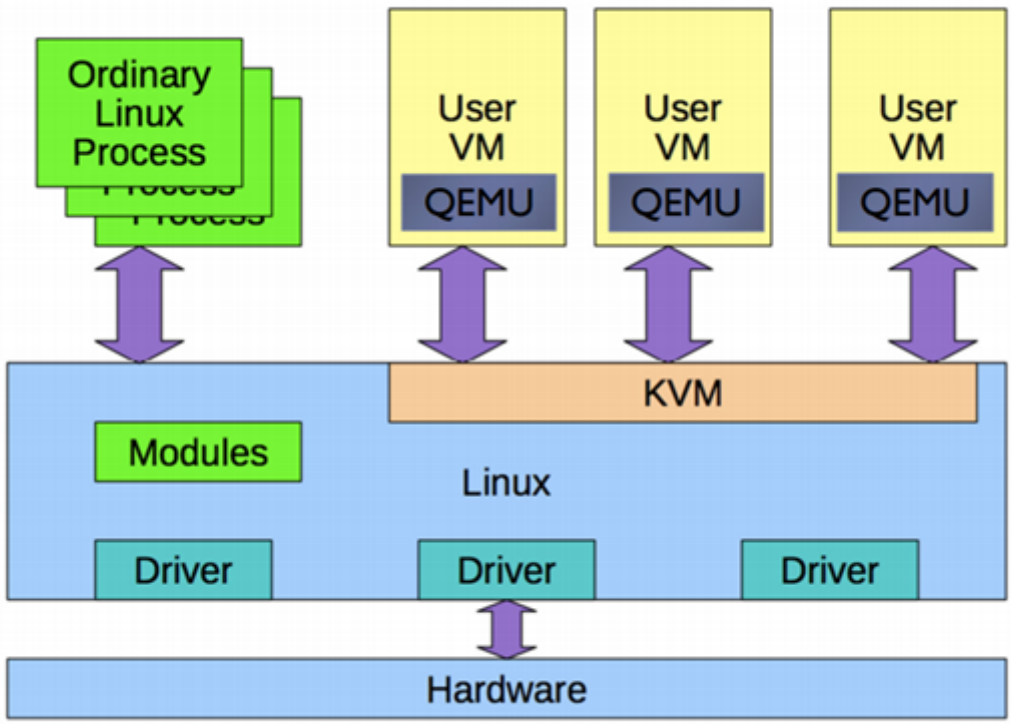
 Light

LEARNNING GNU\LINUX KVM-QEMU VIRTUALIZATION



This project is about learning KVM for virtualization

Install and Configure KVM in Debian

Step:1) Check Whether Virtualization Extension is enabled or not:

```
egrep -c '(vmx|svm)' /proc/cpuinfo
grep -E --color '(vmx|svm)' /proc/cpuinfo
```

Step:2) Install QEMU-KVM & Libvirt packages along with virt-manager

```
#install libvirt packages
sudo apt install -y \
qemu-kvm \
libvirt-clients \
libvirt-daemon-system \
bridge-utils \
virtinst \
libvirt-daemon \
virt-manager
```

```
#osinfo
apt-get install libosinfo-bin

#check status libvirt
sudo systemctl status libvirtd.service
```

Step:3) Start default network and add vhost_net module

```
#show network default and Start
sudo virsh net-list --all

#make it active and auto-restart across the reboot
sudo virsh net-start default
sudo virsh net-autostart default

#add “vhost_net” kernel module
sudo modprobe vhost_net

#add user in  libvirt groups
sudo adduser myuser libvirt
sudo adduser myuser libvirt-qemu

#to refresh or reload group membership run the followings,
newgrp libvirt
libvirt-qemu
```

Step:4) Create Linux Bridge(br0) for KVM VMs

```
sudo vi /etc/network/interfaces



# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens34
auto ens34
iface ens34 inet manual

#Configure bridge and give it a static ip
auto br0
iface br0 inet static
        address 192.168.0.133
        netmask 255.255.255.0
        network 192.168.0.1
```

```
        broadcast 192.168.0.255
        gateway 192.168.0.1
        bridge_ports ens34
        bridge_stp off
        bridge_fd 0
        bridge_maxwait 0
        dns-nameservers 1.1.1.1

# This is an autoconfigured IPv6 interface
iface ens34 inet6 auto



#reboot system
sudo reboot

#check network changes
ip a s br0
```

Install and configure KVM in Rock linux

Install KVM packages

```
# Install the Packages
dnf install -y \
virt-install \
qemu-kvm \
libvirt \
libvirt-python \
libguestfs-tools \
virt-manager

# Enable and Start the Services
systemctl enable libvirtd
systemctl start libvirtd
systemctl status libvirtd

# A Clean System Reboot to ensure everything works after reboot
#init 6

# After the Host reboot, check libvirtd started successfully.
systemctl status libvirtd

# Also need to ensure the kernel modules for KVM are loaded.
modinfo kvm_intel
modinfo kvm
```

Configure bridge network

```
## Variables
BR_NAME="br0"
BR_INT="eth1"
SUBNET_IP="172.36.12.2/24"
GW="172.36.12.1"
DNS1="192.168.0.130"
DNS2="1.1.1.1"

## define the bridge network
nmcli connection add type bridge autoconnect yes con-name ${BR_NAME} ifname ${BR_NAM

## add the IP, gateway, and DNS to the bridge
nmcli connection modify ${BR_NAME} ipv4.addresses ${SUBNET_IP} ipv4.method manual
nmcli connection modify ${BR_NAME} ipv4.gateway ${GW}
nmcli connection modify ${BR_NAME} ipv4.dns ${DNS1} +ipv4.dns ${DNS2}

## Clear old connections
WIRED_NAME=$(nmcli -t -f NAME c show | grep "Wired")
while IFS= read -r NAME; do echo nmcli connection delete "$NAME"; done <<< "$WIRED_N

## Add the identified network device as a slave to the bridge
nmcli connection add type bridge-slave autoconnect yes con-name ${BR_INT} ifname ${B

## Start the network bridge
nmcli connection up br0

## Edit file /etc/qemu-kvm/bridge.conf
# add this line:
allow all

## Restart KVM
systemctl restart libvirtd
```

Storage Pool Configuration

```
# create pv
pvcreate /dev/sdb

# create vg
vgcreate lab_kvm_storage /dev/sdb

# create lv
lvcreate -l +100%FREE -n lab_kvm_lv lab_kvm_storage

# format lv with filesystem xfs
mkfs.xfs /dev/mapper/lab_kvm_storage-lab_kvm_lv

# Edit /etc/fstab and add this lines
#KVM STORAGE -BEGIN
/dev/mapper/lab_kvm_storage-lab_kvm_lv    /var/lib/libvirt/images xfs     defaults
#KVM STORAGE -END
```

```
# mount lv in folder
mount -a
```

Default Paths for VMs

```
$HOME/.local/share/libvirt/images
/var/lib/libvirt/images
```

List of all supported systems

```
osinfo-query os
```

Create the new virtual machine

```
virt-install --name=debian-11-x64 \
--vcpus=1 \
--memory=1024 \
--cdrom=/mnt/isos/Linux/debian-11.0.0-amd64-DVD-1.iso \
--disk size=5 \
--os-variant=debian9
```

Authors

- Marcos Silvestrini
- marcos.silvestrini@gmail.com

License

References

- KVM Docs
- Install and Configure in Debian
- Create and Manage VM's
- Create and Manage VM's
- Rocky Linux Virtualization Techniques
- Create KVM Network Bridge in Rock Linux 9

 Light