

## GNU/Linux Administration - Feature #370

### Setting Up KVM on Debian

04/19/2014 09:09 AM - Daniel Curtis

<b>Status:</b>	Closed	<b>Start date:</b>	04/19/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Daniel Curtis	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	1.00 hour
<b>Target version:</b>		<b>Spent time:</b>	3.00 hours

#### Description

KVM is a full virtualization solution for Linux on x86 (64-bit included) 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`.

In Debian, Xen and VirtualBox are alternatives to KVM.

#### Installation

Install the `qemu-kvm` package with `apt-get` or `aptitude`, e.g. using this command:

```
aptitude install qemu-kvm libvirt-bin
```

The `libvirt-bin` daemon will start automatically at boot time and load the appropriate `kvm` modules, `kvm-amd` or `kvm-intel`, which are shipped with the Linux kernel Debian package. If you intend create VMs from the command-line, install `virtinst`.

In order to be able to manage virtual machines as regular user you should put this user into the `libvirt` group:

```
adduser <youruser> libvirt
```

#### Setting up bridge networking

It can be useful to set up a bridge for the KVM VMs as described here at QEMU page.

- Install the `bridge-utils` package:

```
sudo apt-get install bridge-utils
```

We are going to change the network configuration<sup>1</sup>. To do it properly, you should first stop networking<sup>2</sup>:

```
sudo invoke-rc.d networking stop
```

If you are on a remote connection, and so cannot stop networking, go ahead with the following commands, and use `sudo invoke-rc.d networking restart` at the end. If you make a mistake, though, it won't come back up.

To set up a bridge interface, edit `/etc/network/interfaces` and either comment or replace the existing config with (replace with the values for your network):

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

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
bridge_ports eth0
bridge_stp off
bridge_fd 0
bridge_maxwait 0
```

- Or to use DHCP

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet dhcp
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

This will create a virtual interface br0.

Now restart networking:

```
sudo /etc/init.d/networking restart
```

## Managing VMs from the command-line

You can then use the `virsh(1)` command to start and stop virtual machines. VMs can be generated using `virtinst`. For more details see the `libvirt` page. Virtual machines can also be controlled using the `kvm` command in a similar fashion to QEMU.

### Example installation of a Guest CentOS instance

```
virt-install \
-n myguest \
-r 2048 \
--vcpus=2 \
--os-variant=rhel6 \
-v \
--accelerate \
-c /var/lib/libvirt/images/CentOS-6.2-x86_64-minimal.iso \
-w bridge:br0 \
--vnc --vncport=7601 \
--disk path=/vm/guest.img,size=100
```

The above example uses the following parameters; 100 refers to 100GB for hard drive size, 2048 for MB of RAM, 2 vcpus, and a VNC server setup on port 7601.

### Example installation of a Windows Server 2008 R2

```
# Place WinServ ISO into /storage/local/iso/
# Download VIRTIO driver iso image to use during install
cd /home/user/iso/
wget http://alt.fedoraproject.org/pub/alt/virtio-win/latest/images/bin/virtio-win-0.1-52.iso

# Pre-create VM disk image
qemu-img create -f qcow2 -o preallocation=metadata /home/user/images/winserv1.qcow2 50G
```

```
# Launch installation inside screen
screen
virt-install --connect qemu:///system --name WinServ1 --ram 2048 --vcpus 2 \
--disk path=/home/user/images/winserv1.qcow2,format=qcow2,bus=virtio,cache=none \
--disk path=/home/user/iso/virtio-win-0.1-52.iso,device=cdrom \
--cdrom /home/user/iso/winserv.iso \
--network=bridge:br0,model=e1000 \
--vnc --os-type=windows --os-variant=win2k8 \
--noautoconsole --accelerate --noapic --keymap=en-us
# Detach screen
CTRL+A+D

# Run virt-manager for VNC display (make sure you have X forwarding enabled on your ssh session fo
r remote display)
virt-manager

# OR
virt-viewer -c qemu:///system WinServ1
```

NOTE: Make sure to have the -X option set when connecting via SSH, this is to enable X11 forwarding.

## Managing VMs with a GUI

On the other hand, if you want to use a graphical UI to manage the VMs, you can use the Virtual Machine Manager virt-manager.

## Migrating guests to a Debian host

### Migrating guests from RHEL/CentOS 5.x

There are a few minor things in guest XML configuration files (/etc/libvirt/qemu/\*.xml you need to modify:

- Machine variable in <os> section should say pc, not rhel5.4.0 or similar
- Emulator entry should point to /usr/bin/kvm, not /usr/libexec/qemu-kvm

In other words, the relevant sections should look something like this:

```
<os>
  <type arch='x86_64' machine='pc'>hvm</type>
```

```
--- snip ---
```

```
<devices>
  <emulator>/usr/bin/kvm</emulator>
```

If you had configured a bridge network on the CentOS host, please refer to this wiki article on how to make it work on Debian.

## Troubleshooting

### No network bridge available

virt-manager uses a virtual network for its guests, by default this is routed to 192.168.122.0/24 and you should see this by typing ip route as root.

If this route is not present in the kernel routing table then the guests will fail to connect and you will not be able to complete a guest creation.

Fixing this is simple, open up virt-manager and go to [Edit -> Host details -> Virtual networks](#) tab. From there you may create a virtual network of your own or attempt to fix the default one. Usually the problem exists where the default network is not started.

### cannot create bridge 'virbr0': File exists:

To solve this problem you may remove the virbr0 by running:

```
brctl delbr virbr0
```

Open virt-manager and go to Edit -> Host details -> Virtual networks start the default network.

You can check the netstatus

```
virsh net-list --all
```

Optionally, you can use bridge network BridgeNetworkConnections

## Resources

- <https://wiki.debian.org/KVM>
- <http://www.linux-kvm.org/>
- <http://www.linux-kvm.org/page/HOWTO>
- <http://opennodecloud.com/documentation/howtos/kvm-guests-virt-install-examples/>

---

## History

**#1 - 04/22/2014 06:29 AM - Daniel Curtis**

- Description updated

**#2 - 04/22/2014 11:50 AM - Daniel Curtis**

- Description updated

- Status changed from Resolved to Closed

**#3 - 04/23/2014 09:44 AM - Daniel Curtis**

- Description updated

**#4 - 04/23/2014 09:57 AM - Daniel Curtis**

- Description updated