

Using KVM On Ubuntu 7.10 (Gutsy Gibbon)

By *Mike Weimichkirch*

Published: 2007-11-28 17:38

Using KVM On Ubuntu 7.10 (Gutsy Gibbon)

In this HowTo I'll explain how to install and use KVM for running your services in virtual machines. KVM (Kernel-based Virtual Machine) is a Linux kernel virtualization technique that provides full virtualization by using Intel VT (*Vanderpool*) or AMD-V (*Pacifica*).

I'm working as root on the server. If you are using a normal user, you have to install everything with sudo, e.g.

```
sudo apt-get install kvm
```

Download and install KVM

You can use the package included in Ubuntu 7.10, just execute:

```
apt-get install kvm
```

You can also download the last release of KVM and compile it. But you need following packages for compiling:

```
apt-get install gcc gcc-3.4 gcc-4.1 libstdc++2.9-dev make libz-dev uuid-dev libasound2-dev gcc libc6-dev zlib1g-dev
```

You also need the linux-headers packages for the Kernel you are using. E.g. *linux-headers-2.6.22-14*.

First of all we need the KVM package:

You can find the actual release at <http://sourceforge.net/projects/kvm/>. Download it to your server:

```
wget link/to/file /usr/src/
```

Now unpack the content.

```
cd /usr/src/  
tar -xzf file.tar.gz  
cd folder/to/kvm
```

Before compiling KVM make sure that your CPU supports virtualization and if that option is activated in the BIOS. Without this support, you won't have much performance...

So let's compile KVM.

```
./configure  
make  
make install
```

Now load the KVM modules:

If you have an *Intel* CPU load:

```
modprobe kvm  
modprobe kvm_intel
```

If you have an *AMD* CPU load:

```
modprobe kvm  
modprobe kvm_amd
```

Creating the network bridge

First, we have to install bridge-utils:

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

If you want to connect to your virtual machine later, you need to set up a network bridge.

Edit `/etc/network/interfaces` like this (make sure to insert your public IP Address):

```
auto lo
iface lo inet loopback
auto br0
iface br0 inet static
address xxx.xxx.xxx.xxx
netmask xxx.xxx.xxx.xxx
gateway xxx.xxx.xxx.xxx
bridge_ports eth0
bridge_stp off
bridge_maxwait 5
```

Also edit `/etc/qemu-ifup` like this:

```
#!/bin/sh
/sbin/ifconfig $1 0.0.0.0 promisc up
/usr/sbin/brctl addif br0 $1
sleep 2
```

I recommend restarting your server at this point. If you are able to reconnect, your bridge is working. ***Creating a virtual disk for a virtual machine & install a system into it.***

Create an `.img` file for installation. This file acts as a virtual disk for your vm.

You have 2 possibilities for creating such a file.

Allocate the disk space needed by the .img file directly. In this case the file will have a size of 4 GBytes.

```
dd if=/dev/zero of=disk01.img bs=1G count=4
```

Create an .img file with a maximal size but the disk space will only be allocated when needed. In this case, the maximum size will be 10 GBytes, but the disk space used will be the space required by the files in the vm.

```
qemu-img create -f qcow2 disk01.img 10G
```

Next, we need an ISO file of the system you want to install. Make sure you have downloaded on onto your server.

Let's install the virtual machine. When you don't have physical access to the server, start the installation with the VNC option. So you can connect to the VNC session and you will have the GUI from the installer. In this case we will set the RAM size of the vm to 512 MBytes. At the moment the vm doesn't have access to the network interface. I prefer activating it later, after configuring IP Addresses of the vm.

```
kvm -hda disk01.img -cdrom os.iso -m 512 -boot d -vnc IP:1
```

Now you can connect to IP:1 with a VNC client. You can configure your system and network settings at this point. Don't forget to install a SSH Server. After configuring, shut down the vm.

Now we are going to start the vm with networking support. We assign a MAC-Address to the vm network interfaces. This is important if you use more than 1 vm. Every vm will get its MAC-Address.

```
kvm -hda disk01.img -m 512 -boot c -net nic,vlan=0,macaddr=00:16:3e:00:00:01 -net tap -nographic -daemonize
```

After some seconds you should be able to connect via ssh to the vm. If not, you can add the -vnc option to check if there are problems.

I had the problem that I couldn't reach my vm over the network after some time. I had to add a different network card emulation to the start command. This solved my problem.

```
kvm -hda disk01.img -m 512 -boot c -net nic,vlan=0,macaddr=00:16:3e:00:00:01,model=rtl8139 -net tap -nographic -daemonize
```

If you have the problem that your configured network interface isn't starting, delete the following file (depending on your os):

```
/etc/udev/rules.d/70-persistent-net.rules
```

or

```
/etc/udev/rules.d/z25-persistent-net.rules
```

You also have the possibility to add more virtual disks to one vm. You can add the following parameter to the start command:

```
-hdb disk02.img
```

You can add more virtual machines if you need so. You could also create start scripts for your vms.

Enjoy using KVM! :)