

## KVM\_Setup\_on\_Debian\_Squeeze

KVM on Debian Squeeze is much better and smoother than it was on Lenny. In Squeeze, you can specify virtio with virt-install amongst a lot of other enhancements. All the following was done on a base install of Debian Squeeze with a kernel of 2.6.32-5-amd64. LVM was already setup on the KVM Host.

This how-to documents my kvm setup, in case I need it in future for bare metal recovery. Note: I do not use the default libvirt dhcp private network. I have a mixture of private and public ips and need all vms to be able to see each other. While Libvirt's default network does dhcp and nat, it creates its own iptable rules which are a little difficult to work around, and also while it is possible to fix an IP with dhcp and the mac address, it's a little extra work. As for the disk setup, I do quite a bit of resizing of LVs with LVM, growing and shrinking LVs. While typically only 1 LV would be assigned to a VM and then inside this there would be a /boot, swap and root partitions, it can get a little messy resizing and using fdisk to increase cylinder counts etc. As a result I create 3 LVs for each VM: 1 lv for /boot and grub partitions, 1 lv for swap and 1 lv for root. This makes resizing the LV for root much easier as I don't have to worry about resizing partitions using fdisk. This comes close to the flexibility I had previously with Xen (as it doesn't typically have grub or boot partitions).

```
cat /etc/apt/sources.list
deb http://ftp.de.debian.org/debian/ stable main non-free
# Debian.org security updates
deb http://security.debian.org/ stable/updates main contrib non-free
```

## Contents

- 1 Installation of KVM on Host running Squeeze
- 2 Virsh Shell
  - ◆ 2.1 Virsh Commands for KVM Host, Guests & Default Network
  - ◆ 2.2 Common virsh commands
- 3 Network Setup with KVM
  - ◆ 3.1 Manual Firewall Config
- 4 Create Virtual Machine
  - ◆ 4.1 Underlying xml config for VM
- 5 Other
  - ◆ 5.1 Test disk speed using DD
  - ◆ 5.2 Virt-install from CDROM
  - ◆ 5.3 Change CD ROM
  - ◆ 5.4 virsh console
  - ◆ 5.5 Mount Guest Filesystem on host
  - ◆ 5.6 Install munin node
  - ◆ 5.7 Add cdrom to virsh domain XML config
- 6 Misc Notes

## Installation of KVM on Host running Squeeze

```
apt-get install qemu-kvm libvirt-bin virtinst virt-top
#virtinst is for virt-install tools etc.
#qemu-kvm is the new name for the kvm package in squeeze
#libvirt-bin is what will control kvm and start guests on boot etc.
#virt-top is a 'top'-like utility for virtualization stats
```

## Virsh Shell

### Virsh Commands for KVM Host, Guests & Default Network

```
virsh //You are now in the virsh console
virsh # nodeinfo //Shows info on the Host
virsh # list --all //Shows all VMs. (Same as going ~# ls /etc/libvirt/qemu/). It'll sho
virsh # net-list --all //Shows all the network config files. (Same as going ~# ls /etc/libv
virsh # net-edit default //It is BEST to edit network config files using vish.
//If you edit /etc/libvirt/qemu/networks/default.xml manually, it wo
virsh # net-undefine default //Delete the default libvirt network. (Same as going rm /etc/libvirt/
virsh # quit //Quit virsh
```

### Common virsh commands

Once you have VMs installed and setup, You can enter the virsh shell and specify commands there, or you can prefix command with virsh. E.g.:

```
root@host:~# virsh start vm01 //Boot the VM
root@host:~# virsh destroy vm01 //Turn the power off in VM
#or
virsh #
virsh # start vm01 //Boot the VM
virsh # shutdown vm01 //Shutdown the VM. Requires acpid running on the guest. This com
virsh # edit vm01 //Edit the xml file for the host. (Same as: vi /etc/libvirt/qemu
//Its BEST to edit the host config file using virsh.
//If you edit /etc/libvirt/qemu/vm01 manually, it won't update w
```

## Network Setup with KVM

Some of the config for this comes from Hetzner's wiki. See: <http://wiki.hetzner.de>

A bridged network setup is used. br0 is a bridge for the external ips. br1 is an internal 192.168 network.

```
vi /etc/network/interfaces
auto lo
iface lo inet loopback

# device: eth0
auto eth0
iface eth0 inet static
    address 188.111.111.111
    broadcast 188.111.111.63
    netmask 255.255.255.192
    gateway 188.111.111.1

auto br0
iface br0 inet static
    address 188.111.111.111
    netmask 255.255.255.255
    bridge_stp off
```

## KVM\_Setup\_on\_Debian\_Squeeze

```
bridge_fd 0
pre-up brctl addbr br0
#pre-up brctl addif br0 vnetX //If VMs are running and init.d/networking restart d
up ip route add 188.111.111.122 dev br0
up ip route add 188.111.111.123 dev br0
up ip route add 188.111.111.133 dev br0
post-down brctl delbr br0

auto br1
iface br1 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    bridge_stp off
    bridge_fd 0
    pre-up brctl addbr br1
    pre-up echo 1 > /proc/sys/net/ipv4/ip_forward
    post-down brctl delbr br1

/etc/init.d/networking restart
ifconfig
brctl show
```

## Manual Firewall Config

```
vi /etc/init.d/firewall
#!/bin/sh
IPTABLES=/sbin/iptables

EXTBR=br0
INTBR=br1

PRIVATE=192.168.1.0/24

$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -F POSTROUTING -t nat
$IPTABLES -F PREROUTING -t nat

#####
# FORWARDS
#####
$IPTABLES -A FORWARD -d $PRIVATE -o $INTBR -m state --state RELATED,ESTABLISHED -j ACCEPT
$IPTABLES -A FORWARD -s $PRIVATE -i $INTBR -j ACCEPT
$IPTABLES -A FORWARD -i $INTBR -o $INTBR -j ACCEPT
$IPTABLES -A FORWARD -i $EXTBR -o $EXTBR -j ACCEPT

#####
# NATTING
#####
$IPTABLES -t nat -A POSTROUTING ! -d $PRIVATE -s $PRIVATE -j MASQUERADE
$IPTABLES -t nat -A POSTROUTING ! -s $PRIVATE -d $PRIVATE -j MASQUERADE

#####
# PORT FORWARDING (Remote Desktop)
#####
$IPTABLES -t nat -A PREROUTING -p tcp --dport 8883 -j DNAT --to 192.168.1.2:3389

#####
# BLOCKING
```

## KVM\_Setup\_on\_Debian\_Squeeze

```
#####  
#IPTABLES -A FORWARD -j REJECT --reject-with icmp-port-unreachable  
  
chmod 755 /etc/init.d/firewall  
/etc/init.d/firewall  
iptables -L  
vi /etc/rc.local  
# By default this script does nothing.  
/etc/init.d/firewall  
exit 0
```

## Create Virtual Machine

```
lvcreate -n lin01-boot --size 250m vg0  
lvcreate -n lin01-swap --size 1g vg0  
lvcreate -n lin01-root --size 19g vg0  
mkfs.ext3 /dev/vg0/lin01-root  
mkswap /dev/vg0/lin01-swap
```

```
virt-install -d --name=lin01-quake --ram 512 --disk path=/dev/vg0/lin01-boot,bus=virtio,cache=none -
```

```
#After virt-install, you will see "Cannot open display:"  
#Domain installation still in progress. You can reconnect to the console to complete the installatio  
#Connect with vncviewer. You will have to ssh to the server and port forward port 5900.  
#netstat -tap //This will show the ports open and if they are bound to 127.0.0.1  
#virsh list //will show the vm running  
#As this VM is using br0, manually configure the network with the VM IP and KVM host IP for the Gate  
#Choose Manual disk setup. Assign /boot to the 265MB and set the boot flag. Assign / to the 19GB and  
#Under Software selection, untick Graphical desktop environment and standard system utils for a mini  
#Complete the install. Debian will shutdown the VM after installation
```

```
virsh list --all  
virsh start lin01-quake  
#Connect with VNCviewer and apt-get install ssh
```

## Underlying xml config for VM

```
#vi /etc/libvirt/qemu/lin01-quake.xml  
#It's better to use virsh to edit and save and change the xml files:  
virsh edit lin01-quake  
#Some exerpts are below  
  <disk type='block' device='disk'>  
    <driver name='qemu' type='raw' cache='none'></>  
    <source dev='/dev/vg0/lin01-boot'></>  
    <target dev='vda' bus='virtio'></>  
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'></>  
  </disk>  
  
  <interface type='bridge'>  
    <mac address='xx:xx:xx:xx:xx:xx'></>  
    <source bridge='br0'></>  
    <target dev='vnet0'></>  
    <model type='virtio'></>  
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'></>  
  </interface>
```

## Other

### Test disk speed using DD

```
dd if=/dev/zero of=test bs=1M count=10000
dd if=/dev/zero of=test bs=2M count=5000
dd if=/dev/zero of=test bs=4M count=2500
```

### Virt-install from CDRROM

```
virt-install -d --name=vm02 --ram 512 --disk path=/dev/vg0/vm01,bus=virtio,cache=none --network brid
```

### Change CD ROM

```
virsh attach-disk --type cdrom --mode readonly vm01 /srv/cdimage.iso hdc
#Ref: http://wiki.libvirt.org/page/OEMUSwitchToLibvirt
```

### virsh console

In order to be able to go: virsh console vm, there are a few tweaks that must be done to the guest to get a serial console to which virsh can attach.

```
On guest:
vi /etc/inittab
//uncomment or add in the line:
T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
```

```
//Debian Squeeze with grub2:
vi /etc/default/grub
//ammendn the line:
GRUB_CMDLINE_LINUX="console=tty0"
:wq
update-grub
```

Reboot the guest. Then on the host, you can go virsh console vm, and hit return a few times. To escape the shell you can go:

```
CTRL + ]
```

### Mount Guest Filesystem on host

KVM guests have a virtual disk with partitions for root, swap and also boot sectors. To mount a partition inside the virtual disk, you need to do:

```
apt-get install kpartx
kpartx -av /dev/vg0/vm01disk
```

## KVM\_Setup\_on\_Debian\_Squeeze

```
//this will then list all the partitions and their map points
mount /dev/mapper/vg0-vm01diskp1 /mnt
//job done. To remove:
umount /mnt
kpartx -dv /dev/vg0/vm03
```

## Install munin node

```
apt-get install munin-node munin-libvirt-plugins
vi /etc/munin/munin-node.conf
#Add in munin host ip for allow
/etc/init.d/munin-node restart
```

## Add cdrom to virsh domain XML config

```
virsh edit VMname
//Change:
<boot dev='hd' />
//to:
<boot dev='cdrom' />

    <disk type='file' device='cdrom'>
      <source file='/debian-6.0.1a-amd64-netinst.iso' />
      <target dev='hdc' bus='ide' />
      <readonly />
    </disk>
```

## Misc Notes

[\[Host Setup Config\]](#) [\[Host Interfaces Config\]](#)