

# Converting a virtual machine from VirtualBox to KVM

Submitted by nickaubert on 2009, October 6 - 23:00

- [kvm](#)
- [linux](#)
- [tutorial](#)
- [virtualbox](#)

VirtualBox has been great for me, but I thought I'd give KVM/qemu a try. I'd tried running Bochs around 2003 and it was exciting to see that it actually worked, but the poor performance made it more of a novelty than a useful tool. It's nice to see how far Qemu it's come in this time. It's not only usable but quite handy, with simple CLI management and excellent performance.

Some people report better performance using KVM. I haven't benchmarked KVM against VirtualBox, but I found they both performed adequately on my AMD Athlon. VirtualBox is certainly the slicker of the two: if you run the free-as-in-beer guest extensions, desktop integration is fantastic. I will say I prefer the KVM pure CLI management interface. VirtualBox has notions of tracking virtual machines in a registry while in KVM you specify the hardware specs on invocation. The only thing that's preserved on VM reboots is the hard drive. So converting a VM from VirtualBox to KVM really means converting the hard drive image.

The VM I converted was my Oracle test host which runs CentOS. The first thing to do is to deflate the compressed VDI image to a raw disk image. As of VirtualBox version 3.0.4 this is done using "VBoxManage clonehd":

```
VBoxManage clonehd /path/to/hard_drive_image/guesthd.vdi /path/to/hard_drive_image.img --format raw
```

Be sure to use full path names if you don't want your disk images to end up in ~/.VirtualBox/HardDisks/ Typically in Unix output filenames are going to drop out in the current directory if the full path isn't given. VirtualBox commands don't follow this convention (which is a pretty minor gripe in the scheme of things). The output image is uncompressed, so if this is an image of a 20g drive that's half full, the output file will be the full 20g.

KVM can read raw files, but it can also read Qemu's qcow2 compressed format, so you can save disk space by converting the raw image using qemu-img:

```
qemu-img convert -f raw hard_drive_image.img -O qcow2 hard_drive_image.qcow2
```

Just because you have KVM in the kernel doesn't mean you have the qemu utilities installed, so install them using your distro's application management system. Also note that if you're in the target directory you don't have to bother with full path names. Qemu and KVM tend to behave more like typical Unix commands.

Once the image is ready you can fire up KVM using whatever options suit your needs. This is what mine looks like:

```
kvm -hda /path/to/VCentOracle.qcow -hdb /path/to/Oracle20G.qcow -m 2048 -vga none -usb -usbdevice tablet -net nic -net user,hostfwd=tcp:192.168.1.2:2228-:22
```