(Approx. 1755 words)

## KVM Virtual Machines

By Dick Maybach, Brookdale Computer User Group
[www.bcug.com](http://www.bcug.com)
n2nd (at) att.net

For years, I’ve used VirtualBox to manage virtual machines on my Linux system, but occasionally I’ve had problems with Linux guests. Oracle has said it doesn’t have the resources to resolve this problem, which isn’t surprising as they have far more customers using Windows. An alternative to VirtualBox for Linux users is KVM, which Red Hat developed to manage virtual machines on their Linux servers. (There terminology is confusing. KVM is the core software, but it’s often controlled with the command-line utility QEMU, and you will see it called QEMU/KVM. Most home users use the graphical virt-manager to control KVM. To add to the confusion, virt-manager is titled Virtual Machine Manager in menus.) Linux is Red Hat’s (now a division of IBM) core business, and hosting virtual machines is also important for them, which means KVM is efficient and well-supported. While KVM can run Windows guests, setting them up is somewhat complex, and I still use VirtualBox for these.

While it’s not difficult to install KVM, you can’t just go to the software manager and select a package; it requires some work on the command line. Instead, use the following sequence for Ubuntu and other Debian-related distributions, but substitute your login account for “user” in the third and fourth lines.

sudo apt update

sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager

sudo adduser user libvirt

sudo adduser user kvm

Tutorials on installing on the process are available at

<https://phoenixnap.com/kb/ubuntu-install-kvm> and

<https://www.tecmint.com/install-kvm-on-ubuntu/>.

Read the instructions carefully, as there is more to do than installing most applications.

Although the operations of VirtualBox and KVM are similar, there are significant differences.

* You start a guest machine in VirtualBox by double-clicking on its name. In virt-manager (the graphical interface for KVM), a double-click opens a display window to the virtual machine, which remains off. At this point, you can configure the machine or start it by selecting the start icon in the display window.
* VirtualBox stores the guest files in the user’s home directory, while KVM stores them in the system area at var/lib/libvirt with root as the owner. This may affect your process for backing up guests, for example, if you normally back up only your home directory.
* You must edit the KVM preferences to make connecting USB devices easy. The default settings do this in VirtualBox.
* VirtualBox requires its separately-installed Guest Extensions add-on program to provide several valuable features, such as copy and paste, USB device support, and shared folders. When you upgrade VirtualBox, you must also start each guest and upgrade its Guest Extension driver. KVM includes these features, and upgrading it doesn’t require doing anything in the guests.
* VirtualBox provides better support for Windows. While KVM can run Windows guests, installing them is more complex, and they lack some features, particularly shared folders.
* KVM supports live USB images with persistent storage, while VirtualBox does not; see below.
* The aspect ratio of a KVM guest is determined by its screen resolution, which means that changing the aspect ratio of the display window will result in black borders on either the sides or the top and bottom. The aspect ratio of a VirtualBox guest changes to that of its display window.

Once the virtual machine management software is installed, creating guest machines is similar to VirtualBox. You begin by defining the virtual PC hardware, as shown in Figures 1 through 5. Figure 2 may produce confusion. By default, it tries to detect the operating system you are installing automatically, and this usually fails. The solution is to unselect the automatic operation, type the first letter of the guest (“u” in this case), and choose the closest match. If nothing fits, select “Generic OS.” The program uses this only to estimate what hardware the virtual machine needs. You can change this later; see Figures 4 and 5.



Figure 1. Select a Type of Install Medium.



Figure 2. Select an ISO File.



Figure 3. Set RAM Size and Number of CPUs.



Figure 4. Set Virtual Disk Size.

Think about the disk size, as it’s more difficult to change later than the other parameters.



Figure 5. Name the Guest.

The installation will proceed just as though you were installing on a PC. When it completes, you will have a usable system for running Linux guests with access to the Internet and the ability to copy and paste between host and guest. You will probably want to increase the screen resolution as the default is low, and you do this within the guest.

After creating the virtual machine, you start it from the Virtual Machine Manager, Figure 6. Unlike VirtualBox, double-clicking on the guest’s name only connects the virtual machine to a window; it doesn’t start.



Figure 6. Virtual Machine Manager.

The guest of Figure 7 was started by selecting the small triangle icon (third from the left in the menu bar). Another important menu item is the icon at its extreme right, which starts full-screen mode. To return to window mode, move the cursor to the top of the screen, which will display two icons; the left one is what you want.



Figure 7. Guest Window.

The lightbulb icon (second from left in the menu bar) displays the configuration screen, Figure 8, which allows you to change the virtual machine hardware.



Figure 8. Guest Configuration Screen.

One change you probably want to make is to set up sharing of a directory between the host and the guest.

* Select the *Add Hardware* button.
* Select *Filesystem*
* Enter the information as shown in Figure 9. (Your pathnames will be different). The source path is the host directory to be shared.



Figure 9. Create a Shared Directory.

Reboot the guest. (I’ve found I sometimes must shut down the guest and then start it; a restart command isn’t always effective.) If needed, create a directory to share on the guest; here it’s /home/n2nd/vm\_share. To mount the share, use the following command on the guest terminal.

 sudo mount -t 9p -o trans=virtio, version=9p2000.L hostshare /home/n2nd/vm\_share

This assumes the parameters used above; change them as appropriate. In particular, the target path (hostshare) is an arbitrary label; the only requirement is that what’s shown in Figure 9 is exactly what appears in the mount command. I save the command line in a text file on my desktop, and I paste it into a terminal when I need the share. If you want to mount the share automatically, see <https://ostechnix.com/setup-a-shared-folder-between-kvm-host-and-guest/>. You do need to exercise care, as only one guest at a time should mount the host file. If each guest mounts a share as it boots, you will need a separate host share for each.

Virtual machines make experimenting with live CD-ROMs, and memory sticks quite convenient. First, you download the ISO file and mount it; the time-consuming burn process isn’t needed. Then, in the Virtual Machine Manager, double-click on the virtual machine. Don’t start it but select the light-bulb icon to view the Hardware Details screen. Select the “SATA CDROM 1” item and insert the path to the ISO file, as shown in Figure 10. Then select “Apply.”



Figure 10. Insert an ISO Image in the CDROM Drive.

Then select the Boot Options item, select the “SATA CDROM 1,” and with the arrow keys and move it above VirtIO Disk 1, as shown in Figure 11.



Figure 11. Change the Boot Order.

(This is exactly what you would do on a hardware PC.) The PC will boot from the ISO image. The ISO file of Figure 10 boots PartedMagic, as shown in Figure 12. Note the title of the screen, Pop-20.04 on QEMU/KVM, which identifies the virtual machine. Operations take place as on a hardware PC, except that you can conveniently take screenshots and store them on the host, which can be difficult when operating from a live medium.

When you are finished, go back to the Boot Options screen and uncheck “SATA CDROM 1” (which removes the ISO image) before shutting down the machine. This enables it to boot normally (to Pop-20.04 in this example). But, again, the operation is the same as with a hardware PC.



Figure 12. Booted Live ISO Image.

All the above can also be done with VirtualDisk or VMware Workstation, but only KVM can do the following. Some live memory sticks have persistent storage, a second partition that is available when its OS is operating, and you can run these as virtual machines.

To use persistent storage, you need not an ISO file but an image of a memory stick. For example, Tails, <https://tails.boum.org/>, is available in this form. Tails is a high-security OS, and its persistent storage is encrypted. Having it available on a readily-accessible virtual machine means you are much more likely to use it than if you have to reboot from a memory stick. The following is from <https://tails.boum.org/doc/advanced_topics/virtualization/virt-manager/index.en.html>. First, go to the Tails website and download the USB image file, tails-amd64-4.18.img, in this case. Then, increase the image size to 8 GB with (for example) this (Linux) command. The maximum size of Tails persistent storage is 8 GB, so anything larger than about 10 GB is just wasting space.

 truncate -s 10000M tails-amd64-4.18.img

Follow this procedure.

* Create a new virtual machine, but on the first screen, Figure 1, select “Import existing disk image.”
* On the second screen, Figure 2, browse to the image file, then select “Debian 10” as the operating system you are installing.
* On the third screen, Figure 3, allocate at least 2048 Mbytes of memory (4096 is better if you can) and the number of CPUs.
* On the fourth screen, select a name (probably Tails) and select “Customize configuration” before install; then select “Finish.”
* Select “VirtIO Disk 1” and “Advanced options” and change the “Disk bus” to “USB” (Figure 13); then select “Apply.”
* Finally, on the same screen, select “Removable” and again “Apply.”
* Select “Begin Installation” (at the top left).
* On the Tails welcome screen, select the + button then “Administrative Password” and follow the instructions.



Figure 13. Set the Disk Bus to USB.

Figure 14 shows the resulting virtual machine.



Figure 14. Tails on a Virtual Machine.

There is no additional hard disk space allocated to this virtual machine, as it uses the USB image for storage, just as it would if it were on a memory stick plugged into a hardware PC.

Virtual machines are valuable additions to any operating system, and for Linux users, KVM offers the advantages of higher speed and the ability to use persistent storage.