**Understanding KVM Virtualization**

Different virtualization solutions are available on RHEL 7. The default virtualization

solution, though, is KVM (Kernel Virtual Machine). KVM is included in the

Linux kernel, and the solution offers hypervisor-based virtualization. That means

that you do not have to run a specific program to host VMs; instead, virtualization

support is inside the operating system kernel.

KVM virtualization is not supported by default on every RHEL 7 server; you’ll have

to install it separately on a server that meets the minimal requirements, as described

later in this chapter. KVM virtualization can be used only on 64-bit computer

architecture.

If you are used to using a desktop virtualization solution, such as VMware Workstation

or Oracle Virtual Box, you need to be aware of one important difference

between desktop-based virtualization and hypervisor-based virtualization. In desktop-

based virtualization, the VMs are provided by the virtualization application. As

a result, if you shut down the virtualization application, the VMs running within it

shut down as well.

In KVM hypervisor virtualization, the VMs are running directly on top of the Linux

kernel. As a result, you will not have an application that is running to support VMs

by default, and you might not even notice that the VMs are running on your server.

**Understanding the Role of Libvirtd**

To access VMs that are offered through KVM, you use libvirtd. Libvirtd is a process

that sits between the virtualization layer and the application that an administrator

is using to access the VMs.

**Understanding KVM Host Networking**

After installing the virtualization software on a host computer, networking also

changes significantly. On the host, a virtual bridge is created. This virtual bridge

works like an embedded switch, and it is used to connect one or more of the physical

network interfaces in the host to the different VMs.

While communicating on the network, a VM sends out packets through its internal

(virtual) network interface, which typically has the name eth0. At the hypervisor

level, this network is represented by a vnet interface. The first VM that starts gets

the interface vnet0, the second machine that starts gets vnet1, and so on.

This vnet interfaces on their turn connect to the virtual bridge. The virtual bridge

itself is connected to the physical network interface on your host. To get an overview of the virtual networking configuration, you can use the command **brctl show** . This command for which you can see the output in Listing 10.3 shows the

name of the bridge, the unique ID that has been assigned to the bridge, an indicator

that shows if the bridging protocol STP (Spanning Tree Protocol) has been

enabled, and all the interfaces that are connected to the bridge.