**Recovering Access to a Virtual Machine**

A special case of troubleshooting is a situation that involves a virtual machine. If

you have a problem in the virtual machine, and you cannot access it using Virtual

Machine Manager or **virsh console** as described in Chapter 10 , “Working with

Virtual Machines,” you also can’t connect to it using SSH. However, there is still an

option. The technique described here is a bit advanced, but because it might someday

help you reestablish access to a virtual machine that would otherwise have been

lost, it is worth knowing about.

**TIP** The procedure that is described here is relatively complicated, and you

should not have to apply it unless you mess up severely during the exam. So instead

of memorizing this procedure, just be very careful to prevent issues like this from

happening!

**1.** To start, open a root shell on the KVM host. From that host file, make sure

that the virtual machine is stopped by using **virsh destroy** *vmname* . (Use the

name of the virtual machine as listed by the **virsh list** command.).

**2.** Find the disk image file. It normally is stored in the /var/lib/libvirt/images

directory. If you cannot find it there, use **virsh dumpxml vmname | grep**

**“source file=”** to find the name of the source file:

[root@lab ~]# virsh dumpxml sander-vm1 | grep "source file="

< source file= '/home/sander/lab1.img'/>

< source file= '/var/lib/libvirt/images/sander-vm1.img'/>

**3.** Now that you know the name of the virtual machine image file, you can mount

it into the host environment by using the **kpartx -a** command. This command

analyzes the disk layout on the virtual machine and creates storage devices that

allow you to mount devices in the virtual machine in the host file system:

[root@lab ~]# kpartx -av /home/sander/lab1.img

add map loop0p1 (253:5): 0 1024000 linear /dev/loop0 2048

add map loop0p2 (253:6): 0 7362560 linear /dev/loop0 1026048

**4.** The **kpartx** command has created devices that enable you to access the two

partitions in the virtual machine. If this is a typical RHEL 7 setup, the first

partition is the /boot partition, and the second partition is normally used for

LVM logical volumes. You can mount the /boot partition easily by using

**mount /dev/mapper/loop0p1 /mnt** .

**5.** To access the logical volumes that exist within the virtual machine’s second

partition, use the **pvscan** command:

[root@lab mapper]# pvscan /dev/mapper/loop0p2

PV /dev/sda3 VG centos lvm2 [48.83 GiB / 0 free]

PV /dev/sda4 VG centos lvm2 [50.00 GiB / 20.00 GiB

free]

PV /dev/sda5 VG vglvm lvm2 [347.32 GiB / 329.32 GiB

free]

PV /dev/mapper/loop0p2 VG centosvm lvm2 [3.51 GiB / 0

free]

Total: 4 [449.65 GiB] / in use: 4 [449.65 GiB] / in no VG: 0

[0 ]

**6.** You now have activated the LVM setup within the virtual machine. If you type

the **lvs** command, you should show the logical volumes listed as well, but in

an active state. To activate them, use the **vgchange** command on the volume

group that was found in the virtual machine disk image file. So for this exam,

you type **vgchange -a y centosvm** to activate all logical volumes. After doing

so, you can mount them also to directly access all files in the virtual machine

file system and fix all issues that prevented the virtual machine from booting

normally.

**7.** After you have fixed all issues in the virtual machine, you first need to

unmount all file systems currently mounted. Next, you need to remove the

device files that have been created based on the contents of the virtual machine

image file by using the **kpartx -dv /home/sander/lab1.img** command.

(Make sure to use the filename of the image file that is actually used on your

computer.)