

Training Services

# Hands-on with KVM Virtualization and Libvirt Workbook

Course ATT1802  
Version 1.0.0

Lab Manual  
September 7, 2011

[www.suse.com](http://www.suse.com)



## Proprietary Statement

Copyright © 2012 Novell, Inc. All rights reserved.

Novell, Inc., has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the [Novell Legal Patents Web page \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.  
404 Wyman Street, Suite 500  
Waltham, MA 02451  
U.S.A.  
[www.novell.com](http://www.novell.com)

## Novell Trademarks

For Novell trademarks, see [the Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

## Third-Party Materials

All third-party trademarks are the property of their respective owners.

## Software Piracy

Throughout the world, unauthorized duplication of software is subject to both criminal and civil penalties.

If you know of illegal copying of software, contact your local Software Antipiracy Hotline. For the Hotline number for your area, access [Novell's World Wide Web page \(http://www.novell.com\)](http://www.novell.com) and look for the piracy page under "Programs." Or, contact Novell's anti-piracy headquarters in the U.S. at 800-PIRATES (747-2837) or 801-861-7101.

## Disclaimer

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

This Novell Training Manual is published solely to instruct students in the use of Novell networking software. Although third-party application software packages are used in Novell training courses, this is for demonstration purposes only and shall not constitute an endorsement of any of these software applications.

Further, Novell, Inc. does not represent itself as having any particular expertise in these application software packages and any use by students of the same shall be done at the student's own risk.

---

# Contents

<b>Section 1</b>	<b>Introduction to Virtualization with KVM.....</b>	<b>7</b>
Exercise 1.1	Install a Virtualization Host Server on SUSE Linux Enterprise.....	8
	Task I: Install Virtualization Software.....	8
<b>Section 2</b>	<b>Introduction to Virtual Machines.....</b>	<b>9</b>
	(No Lab Exercises).....	10
<b>Section 3</b>	<b>Introduction to Open Source Virtualization Management... </b>	<b>11</b>
Exercise 3.1	Install a SLES11 KVM Virtual Machine.....	12
	Task I: Install a KVM Virtual Machine.....	12
<b>Section 4</b>	<b>Manage Virtualization Platforms with Libvirt.....</b>	<b>15</b>
Exercise 4.1	Use Common virsh Commands with KVM.....	16
	Task I: Work with Unmanaged VMs.....	16
	Task II: Work with Managed VMs.....	16
Exercise 4.2	Use virt-manager to Manage Local KVM Virtual Machines.....	19
	Task I: Launch virt-manager.....	19
	Task II: Manage Local VMs with virt-manager.....	19
	Task III: Edit VM Configuration with virt-manager.....	20
Exercise 4.3	Configure Key Based Authentication in OpenSSH.....	21
	Task I: Generate an SSH Key Pair.....	21
	Task II: Upload the Public Key to a Remote Server.....	21
	Task III: Verify Key Based Login.....	22
Exercise 4.4	Use virsh to Connect to a Remote System via SSH.....	23
	Task I: Use virsh to Connect to a Remote System via SSH.....	23
Exercise 4.5	Use virt-viewer to Connect to a Remote KVM VM.....	24
	Task I: Launch a KVM VM on a Remote vHost.....	24
	Task II: Connect to a Remote KVM VM with virt-viewer.....	24
Exercise 4.6	Manage VMs on a Remote vHost with virt-manager.....	26
	Task I: Connect to a Remote vHost with virt-manager.....	26
	Task II: Launch a VM on a Remote vHost.....	27
<b>Section 5</b>	<b>Manage Virtualization Administration and Security.....</b>	<b>29</b>
Exercise 5.1	Use PolicyKit to Delegate VM Administration.....	30
	Task I: Attempt to Use virt-manager as a Non-root User.....	30
	Task II: Delegate the Ability to See Local VMs with PolicyKit.....	30
	Task III: Test The Ability to View VMs in the GUI.....	31
	Task IV: Delegate the Ability to Manage Local VMs with PolicyKit.....	31
	Task V: Test the Ability to Manage VMs in the GUI.....	32
	Task VI: Test the Ability to Manage VMs from the CLI.....	32
Exercise 5.2	Use Unix Group Membership to Delegate Libvirt Administration .....	33

	Task I: Create a Virtualization Admin User.....	33
	Task II: Edit the libvirtd.conf File.....	33
	Task III: Use virsh to Connect to the Local System.....	34
<b>Exercise 5.3</b>	<b>Use SASL+Digest-md5 to Delegate Libvirt Administration.....</b>	<b>35</b>
	Task I: Edit the sasl2/libvirt.conf File.....	35
	Task II: Create an SASL User.....	35
	Task III: Enable SASL for the Libvirt Daemon.....	35
	Task IV: Enable SASL for the QEMU Libvirt Driver.....	36
	Task V: Use virsh to Connect to the Local System.....	36
<b>Section 6</b>	<b>Other Virtualization Management Tasks.....</b>	<b>39</b>
<b>Exercise 6.1</b>	<b>Pause and Save KVM VMs.....</b>	<b>40</b>
	Task I: Pause and Unpause a Running VM with Virt-Manager.....	40
	Task II: Suspend and Resume a Running VM with virsh.....	40
	Task III: Save and Restore a Running VM.....	41
	Task IV: Managesave and Start a Running VM.....	41
<b>Exercise 6.2</b>	<b>Automatically Boot KVM VMs with libvirt-guests.....</b>	<b>43</b>
	Task I: Enabling VM Automatic Booting.....	43
	Task II: Simulate a Shutdown and Restart of the vHost and VMs.....	43
<b>Section 7</b>	<b>Configure Virtual Hardware for KVM Virtualization.....</b>	<b>45</b>
<b>Exercise 7.1</b>	<b>Configure a Virtual Serial Console.....</b>	<b>46</b>
	Task I: Configure a Virtual Serial Device.....	46
	Task II: Configure Access to the Serial Console in the VM.....	46
	Task III: Connect to the Virtual Serial Console.....	47
<b>Exercise 7.2</b>	<b>Configure a Virtual Watchdog Device.....</b>	<b>48</b>
	Task I: Configure a Virtual Watchdog Device.....	48
	Task II: Verify the Watchdog Device in the VM.....	48
<b>Exercise 7.3</b>	<b>Create a Libvirt Virtual Network with Virt-manager.....</b>	<b>50</b>
	Task I: Create a Virtual Network.....	50
<b>Exercise 7.4</b>	<b>Configure a Virtual Network with libvirt.....</b>	<b>52</b>
	Task I: Create a Virtual Network Definition File.....	52
	Task II: Activate a Virtual Network with virsh.....	52
	Task III: Enable the Virtual Network to Start Automatically.....	53
<b>Exercise 7.5</b>	<b>Snapshot a QCOW2 Virtual Disk.....</b>	<b>55</b>
	Task I: Create Snapshots of a QCOW2 Disk.....	55
	Task II: Revert to and Delete Snapshots.....	56

---

# List of Figures



## **Section 1 Introduction to Virtualization with KVM**

This section gives an introduction to the KVM virtualization environment.

## 1.1 Install a Virtualization Host Server on SUSE Linux Enterprise

In this exercise you will install the packages required for SLES to run as a Xen and KVM vHost.

### Objectives:

Task I: Install Virtualization Software

### Special Instructions and Notes:

(none)

### Task I: Install Virtualization Software

1. Launch the YaST Software Management module:  
**YaST > Virtualization > Install Hypervisor and Tools**
2. When prompted to select the virtualization platform to install, select **Xen** and **KVM** and then click **Accept**
3. When prompted that packages need to be installed, click **Continue** to install packages
4. When prompted to create a default network bridge, click **Yes**
5. When prompted to reboot, click **OK**
6. Reboot the machine.

**(End of Exercise)**



## Section 2 Introduction to Virtual Machines

This section provides an introduction to the anatomy and configuration fo virtual machines

**(No Lab Exercises)**

## **Section 3 Introduction to Open Source Virtualization Management**

This section covers utilities used to manage Xen and KVM virtualization platforms.

## 3.1 Install a SLES11 KVM Virtual Machine

In this exercise you use the `vm-install` utility to install a new SLES11 KVM virtual machine.

### Objectives:

Task I: Install a KVM Virtual Machine

### Special Instructions and Notes:

Perform the following Lab on the VM Server (Host Machine).

`INSTALL_URL=_____`

### Task I: Install a KVM Virtual Machine

7. Launch the Virtual Machine Installation utility:  
**YaST > Virtualization > Create Virtual Machines**
8. Begin Installation in the `vm-install` utility by clicking the **Forward** button
9. Click **I need to install an operating system** and then click **Forward**
10. Select **SUSE Linux Enterprise Server 11** from the list of OSes and then click **Forward**
11. Use the following initial settings to configure the VM installation:

---

**NOTE:** To install from a network install server do not add a CDROM drive in the Drives section and add the install URL optionally recorded above as the Install Source in the Operating System Installation section.  
(Hint: Select the blue text title of an entry to change its parameters)

---

#### Name of Virtual Machine

vm1

#### Hardware

Initial Memory: 256 MB  
Maximum Memory: <accept the default>  
Virtual Processors: 1

#### Perhipheral Devices:

Graphics Adapter: Cirrus Logic GD5446 VGA  
Keymap: en-us

Sound Card: None

### Disks

1: 8.0 GB Hard Disk (/var/lib/xen/images/vm1/disk0)

### Network Adapters

1: QEMU Virtualized NIC Card: Randomly generated MAC address

### Operating System Installation

Operating System: SUSE Linux Enterprise Server 11

Installation Source: *INSTALL\_URL*

Automated Installation:

Additional Arguments:

---

**NOTE:** If you wish to install from CD/DVD, add a CD-ROM/DVD as an installation source select the following (DO NOT do this if your are installing form the network):

**DISKS:**

**CD-Rom**

**OK**

**Apply**

You should now see a new CD-ROM entry under Disks:

Under the Operating System Installation section you should also see the Installation Source entry populated with the CD/DVD as the source.

---

12. Click **OK** to start the installation:
13. Continue with the SLES 11 installation as normal in the VNC installation window that appears.

### Notes on install:

Select all defaults for the installation with the following exceptions:

-Server Base Scenario: Physical Machine (or full virtual machine)

-root password = linux

-hostname = vm1.site

-Skip the Internet test

-Do not create a local user

-Do not clone the system for AutoYaST

- You must click inside the installation window to give it focus. To release focus from the window in which the installation is happening you must press **ctrl+alt**
- If you cannot move the mouse cursor to a part of the VMs screen release the cursor, move the cursor to the top left hand corner of the VM and then click inside the VM window. This should resynchronize the mouse cursors.

**(End of Exercise)**

## **Section 4 Manage Virtualization Platforms with Libvirt**

This section covers Libvirt and Libvirt based utilities used to manage Xen and KVM virtualization platforms.

## 4.1 Use Common virsh Commands with KVM

In this exercise you will use common virsh commands to work with KVM VMs.

### Objectives:

Task I: Work with Unmanaged VMs

Task II: Work with Managed VMs

### Special Instructions and Notes:

You must have KVM installed to perform this exercise.

**VM1\_NAME**= \_\_\_\_\_

**VM2\_NAME**= \_\_\_\_\_

### Task I: Work with Unmanaged VMs

In this task you will use common virsh command to work with an unmanaged KVM VM.

1. If you are not already logged in as the root user, open a terminal window and use the **su -** command to become root
2. Enter the following command s to launch the **VM1\_NAME** VM using only it's configuration:

```
cd /vmstore/VM1_NAME
```

```
virsh create VM1_NAME.xml
```

3. View that the VM is running by entering the following:

```
virsh list
```

You should see that the vm is running on your machine

4. Shutdown the VM by entering the following:

```
virsh shutdown VM1_NAME
```

---

**NOTE:** You may need to connect to the VM's virtual framebuffer (GUI console) and enter the root user's password for the VM to shut down.

---

5. See that the VM is no longer running:

```
virsh list
```

### Task II: Work with Managed VMs

In this task you will use common virsh commands to work with a managed KVM VM.

1. List the existing managed and currently running VMs by entering the following:

```
virsh list
```



You should not see information about the *VM1\_NAME* vm.

2. To make the *VM1\_NAME* VM a managed VM, enter the following command:

```
cd /vmstore/VM1_NAME
virsh define VM1_NAME.xml
cd /vmstore/VM2_NAME
virsh create VM2_NAME.xml
```

3. To see that the VM1 and VM2 are now “managed VMs” enter the following command:

```
virsh dominfo VM1_NAME
virsh dominfo VM2_NAME
```

You should see information about the VMs.

4. Launch the VM1 VM by entering the following command:

```
virsh start VM1_NAME
```

5. To see that the VM is running enter the following command:

```
virsh list
```

You should see that the *VM1\_NAME* VM is now running

Record the *VM1\_NAME*'s VM1 ID number here: *VM1\_ID*=\_\_\_\_\_

6. Shut down the *VM1\_NAME* VM using it's VM ID number by entering the following command:

```
virsh shutdown VM1_ID
```

7. Export the *VM1\_NAME*'s configuration out of the managed VM database into a file by entering the following (with no line wraps):

```
virsh dumpxml VM1_NAME > /tmp/VM1_NAME-test.xml
```

8. View the contents of the xml file you just exported by entering the following:

```
less /tmp/VM1_NAME-test.xml
```

9. Remove the *VM1\_NAME* VM's configuration from the managed VM database by entering the following:

```
virsh undefine VM1_NAME
```

10. To see that the *VM1\_NAME*'s configuration is no longer being stored in the managed VM database enter the following command:

```
virsh dominfo VM1_NAME
```

You should not see information about the *VM1\_NAME* VM.

11. To re-import the *VM1\_NAME*'s configuration into the managed VM database using the exported configuration information enter the following command:

```
virsh define /tmp/VM1_NAME-test.xml
```

12. To see that the *VMI\_NAME* VM is now managed again enter the following command:

```
virsh dominfo VMI_NAME
```

You should see information about the VM.

**(End of Exercise)**

## 4.2 Use virt-manager to Manage Local KVM Virtual Machines

In this exercise, you use the virt-manager utility to manage local virtual machines.

### Objectives:

Task I: Launch virt-manager

Task II: Manage Local VMs with virt-manager

Task III: Edit VM Configuration with virt-manager

### Special Instructions and Notes:

You must have at least one virtual machine's configuration uploaded into the manage VM database to perform this exercise. If you do not, use the **virsh define** command to import a VM's configuration into the managed database

**KVM\_VM1\_NAME=**\_\_\_\_\_

**KVM\_VM1\_CONFIG=**\_\_\_\_\_

**KVM\_VM2\_NAME=**\_\_\_\_\_

**KVM\_VM2\_CONFIG=**\_\_\_\_\_

### Task I: Launch virt-manager

1. If not logged in as the root user, enter **su -** to become root.
2. Enter the following commands to make 2 virtual machines managed:

```
virsh define KVM_VM1_CONFIG
```

```
virsh define KVM_VM2_CONFIG
```

3. Launch virt-manager:

```
Alt+F2
```

```
gnomesu virt-manager
```

---

**NOTE:** If you launch virt-manager outside of YaST, you must double-click on the **localhost (QEMU)** entry to connect to the local instance of libvirt

---

4. You should see all running and managed virtual machines on the local system

### Task II: Manage Local VMs with virt-manager

1. To launch a virtual machine, right-click a non-running VM in the list and select

**Run**

The virtual machine should start

---

**NOTE:** If the VM fails to start, an error message will appear describing the reason the VM failed to start.

---

2. To view the virtual frame buffer (GUI console) of the running VM, either double-click on the VM or highlight the VM and click the **Open** button  
The virtual frame buffer or GUI console of the VM should now be visible
3. To stop the VM, either right-click on the VM in the main virt-manager window and select **shutdown** or if the GUI console is open, click on the **Shutdown** button at the top of the window  
The VM should now be shutting down

### Task III: Edit VM Configuration with virt-manager

1. From the main virt-manager window, either right-click on a VM and select **Open** (or just double-click on the VM in the list). On the Virtual Machine window that appears click the **Details** button  
You should now see the virtual machine's details window.
2. Change the virtual machine's hardware configuration by selecting **Memory**, and using the **Change allocation** field to change the amount of memory assigned to the virtual machine.  
You may also start/stop/pause a virtual machine from this window as well
3. Close the virtual machine's details window and close virt-manager

**(End of Exercise)**

## 4.3 Configure Key Based Authentication in OpenSSH

In this exercise, you practice using SSH with public key authentication.

First, you create an ssh-key pair on your vHost. Then you add the public key to the `/root/.ssh/authorized_keys` file on your lab partner's server and note the difference between logging in with and without a public key.

You may wish to perform these same tasks for the `geeko` user as well (using the `geeko` user's home directory and credentials in place of `root`'s)

### Objectives:

Task I: Generate an SSH Key Pair

Task II: Upload the Public Key to a Remote Server

Task III: Verify Key Based Login

### Special Instructions and Notes:

Use the following value(s) in this exercise:

**REMOTE\_IP**=\_\_\_\_\_

### Task I: Generate an SSH Key Pair

1. Log in to your machine as **root**
2. Open a terminal window and enter the following command to generate an SSH key pair:

```
ssh-keygen -t rsa
```

3. Accept the default location for the key (`/root/.ssh/id_rsa`)
4. Press **Enter** twice to set an empty passphrase on the keypair

Information about your key pair, such as the location of your identification and the public key, is displayed

### Task II: Upload the Public Key to a Remote Server

1. On your machine, enter the following command to upload your public key to `root` on your lab partner's machine:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@REMOTE_IP
```

After authentication you should see a message that the key was uploaded successfully

---

**Note:** The longer way to do this would be to `scp` the key to the remote server:

```
scp ~/.ssh/id_rsa.pub root@REMOTE_IP:/root/
```

---

SSH into the remote server:

```
ssh root@REMOTE_IP
```

Append the key to the known\_hosts file:

```
cat ~/id_rsa.pub >> ~/.ssh/known_hosts
```

---

### Task III: Verify Key Based Login

1. Enter the following command to connect to the remote host via ssh as root:

```
ssh root@REMOTE_IP
```

You should be logged into the remote server without being prompted for a password or passphrase

**(End of Exercise)**

## 4.4 Use virsh to Connect to a Remote System via SSH

In this exercise, you use the virsh utility to connect to a remote system via SSH.

### Objectives:

Task I: Use virsh to Connect to a Remote System via SSH

### Special Instructions and Notes:

You must have the server and client certificates installed before you perform this exercise

### Task I: Use virsh to Connect to a Remote System via SSH

1. Log into the machine designated as the **client** as the **root** user with the password **novell**
2. Enter the following command to connect to the remote machine:  

```
virsh -c qemu+ssh://USERNAME@LIBVIRT_SRVR_FQDN/system
```

---

**Note:** If you are using Xen enter the following URI instead:

```
xen+ssh:// USERNAME@LIBVIRT_SRVR_FQDN
```

---

When prompted for the password, enter the password for ***USERNAME***

You should be at a **virsh #** prompt on the remote machine

3. Enter the following command to view information about the vHost:  

```
nodeinfo
```
4. Enter the following command to view a list of the VMs that the vHost knows about:  

```
list --all
```
5. Enter the following command to view a list of the Libvirt virtual networks configured on the vHost:  

```
net-list --all
```
6. Enter the following command to disconnect from the remote machine:  

```
quit
```

You should now be back to the command prompt on your machine

**(End of Exercise)**

## 4.5 Use virt-viewer to Connect to a Remote KVM VM

In this exercise you will use the virt-viewer utility to connect to the GUI console of a running remote KVM VM.

### Objectives:

Task I: Launch a KVM VM on a Remote vHost

Task II: Connect to a Remote KVM VM with virt-viewer

### Special Instructions and Notes:

*REMOTE\_IP*: \_\_\_\_\_  
*VM\_NAME*: \_\_\_\_\_  
*VM\_CONFIG\_FILE*: \_\_\_\_\_

### Task I: Launch a KVM VM on a Remote vHost

In this task you will launch a KVM virtual machine on a remote vHost.

1. On the remote vHost, if one is not already running, launch a KVM virtual machine:

---

**Note:** These commands are each a single line with no line wraps

---

for a managed vm:

```
virsh -c qemu+ssh://root@REMOTE_IP/system start  
VM_NAME
```

for an unmanaged vm:

```
virsh -c qemu+ssh://root@REMOTE_IP/system create  
VM_CONFIG_FILE
```

If prompted for the root user's password, enter **novell**

### Task II: Connect to a Remote KVM VM with virt-viewer

In this section you will use virt-viewer to connect to the GUI console of a running remote virtual machine.

1. To connect to the running virtual machine with virt-viewer, on a different machine enter the following command:

---

**Note:** These 2 commands are each on single lines with no line wraps

---

**Alt+F2**

```
virt-viewer -c qemu+ssh://root@REMOTE_IP/system  
VM_NAME
```

2. If prompted for a password, enter the root users password on the machine where



the VMs are running

You should now see the GUI console of the virtual machine.

3. To switch to virtual terminal 1 select the following:

**Send Key > Ctrl+Alt+F1**

You should now see the login prompt at virtual terminal 1

4. To switch back to the GUI select the following:

**Send Key > Ctrl+Alt+F7**

You should now see the GUI

5. To lock the pointer in the VM's virt-viewer window select the window with the mouse pointer and then enter the following:

**Ctrl+Alt**

You should no longer be able to move the mouse pointer out of the VM's virt-viewer window. All input will be grabbed only by the VM.

---

**TIP:** The title bar of the VM's virt-viewer window will show you when the mouse pointer has been grabbed by that windows and the key stroke used to release the pointer.

---

6. To release the pointer form the VM's virt-viewer window, enter the following:

**Ctrl+Alt**

You should now be able to move the mouse pointer out of the VM's virt-viewer window.

7. Close the virt-viewer window to disconnect from the remote VM

**(End of Exercise)**

## 4.6 Manage VMs on a Remote vHost with virt-manager

In this exercise you will use the virt-manager utility to connect to and manage a remote vHost.

### Objectives:

Task I: Connect to a Remote vHost with virt-manager

Task II: Launch a VM on a Remote vHost

### Special Instructions and Notes:

**VMHOST\_IP:** \_\_\_\_\_

This exercise was written based on virt-manager 0.9.x. If you are using a different version, the exercise will still work but the steps might be slightly different.

### Task I: Connect to a Remote vHost with virt-manager

In this section you will use virt-manager to connect to a remote VM server..

1. While logged in as the gecko user, launch the virt-manager utility as follows:

**Alt+F2**

**virt-manager**

2. Select from the menu bar: **File > Add Connection**
3. Select **Connect to remote host** and then select the following:

Method: **SSH**

Username: **root**

Hostname: **VHOST\_IP**

Selecting Autoconnect is optional

4. Click the **Connect** button
5. If you are prompted for a ssh password, enter the password of the root user on the remote machine

---

**TIP:** The “accept ssh host key” and “ssh password” prompts may appear behind the virt-viewer window. You must type “yes” and then click **OK** to accept the host key. If you do not see them after clicking **Connect**, move the windows around to find them.

If the connection fails and you don't see a ssh askpass prompt, you will need to generate a ssh key and then copy it to the remote machine:

```
ssh-keygen -t rsa
```

---

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@VHOST_IP
```

---

You should now see a list of VMs that are on your teammates VM server.

## Task II: Launch a VM on a Remote vHost

In this task you will launch a Xen Virtual machine on a remote vHost.

1. Double-click one of the non-running VMs on the remote vHost
2. In the Virtual Machine Console window that appears, click **Run**
3. If you are prompted for an ssh password, enter the password of the root user on the remote machine

You should now see the console of the VM running on the remote VM server

4. In the Virtual Machine Console window, click **Shutdown**  
You should see the VM running on the remote VM server shut down
5. If you are prompted for an ssh password, enter the password of the root user on the remote machine

**(End of Exercise)**



## Section 5 Manage Virtualization Administration and Security

This section covers the delegation of virtualization administration and configuration of virtual machine security with Libvirt.

## 5.1 Use PolicyKit to Delegate VM Administration

In this exercise, you use PolicyKit to grant authorization to a non-root user to be able to use libvirt based tools to administer virtual machines.

### Objectives:

- Task I: Attempt to Use virt-manager as a Non-root User
- Task II: Delegate the Ability to See Local VMs with PolicyKit
- Task III: Test The Ability to View VMs in the GUI
- Task IV: Delegate the Ability to Manage Local VMs with PolicyKit
- Task V: Test the Ability to Manage VMs in the GUI
- Task VI: Test the Ability to Manage VMs from the CLI

### Special Instructions and Notes:

You must be booted into the Xen virtualization environment or have KVM enabled to perform this exercise

**VM\_NAME=**\_\_\_\_\_

The VM listed above must already be running to perform this exercise. If it is not running, you must launch it.

### Task I: Attempt to Use virt-manager as a Non-root User

**IMPORTANT:** If you will be using Xen virtual machines, ensure that you are booted into the Xen virtualization environment by running the following command:

```
uname -r
```

If you do not see “xen” in the kernel name, you must reboot into the Xen environment.

---

1. Log in as the user geeko
2. Launch Virt-Viewer:

**Alt+F2**

**virt-viewer VM\_NAME**

You should be denied access because of failed authentication (you are not root)

### Task II: Delegate the Ability to See Local VMs with PolicyKit

1. Launch the GNOME PolicyKit Authorizations utility:  
**Computer > More Applications > Tools > Authorizations**
2. To allow the geeko user to see VMs on the local VM Server, select:  
**org > libvirt > unix > Monitor local virtualized systems**
3. To see the users that have been granted or denied authorizations, select:

**Show authorizations from all users**

You should see no users in the list

4. On the “Authentication is required to read authorizations of other users” pop-up window enter the root password and click **Authenticate**
5. Next to the Explicit Authorizations area, click **Grant**
6. From the Beneficiary drop down list, select the **geeko** user
7. Under Constraints select **none** and click **Grant**
8. Back in the Authorizations application, on the “Authentication is required to grant authorizations of other users” pop-up window, enter the root password and click **Authenticate**
9. Do not close the Authorizations application window

---

**Note:** you can also do this from the command line using the following command (single command with no line wraps):

```
polkit-auth --user geeko --grant  
org.libvirt.unix.monitor
```

---

**Task III: Test The Ability to View VMs in the GUI**

1. While still logged in as the geeko user, launch Virt-Viewer again:  
**Alt+F2**  
**virt-viewer VM\_NAME**
2. You should now be connected to the VM
3. Close Virt-Viewer

**Task IV: Delegate the Ability to Manage Local VMs with PolicyKit**

1. Switch back in the Authorizations application
2. To allow the geeko user to manage vms on the VM Server, select:  
**org > libvirt > unix > Manage local virtualized systems**
3. Next to the Explicit Authorizations area, click **Grant**
4. From the Beneficiary drop down list, select the **geeko** user
5. Under Constraints select **none** and click **Grant**
6. On the “Authentication is required to grant authorizations of other users” pop-up window enter the root password and click **Authenticate**
7. Click **Close** to close the GNOME PolicyKit Authorization utility

---

**Note:** you can also do this from the command line using the following command (single command with no line wraps):

```
polkit-auth --user geeko --grant
```

---

---

`org.libvirt.unix.manage`

---

## Task V: Test the Ability to Manage VMs in the GUI

1. While still logged in as the `geeko` user, launch Virt-Manager:  
**Alt+F2**  
**virt-manager**
2. Double-click on the **localhost** entry
3. You should still be able to see the VMs on this VM Server
4. Right-click on one of the non-running VMs and you should see the **Run** option enabled

---

**NOTE:** You may launch the VM if you wish but it is not required. Seeing the Run option enabled in the menu is enough to see that the delegation of permissions was successful.

---

5. Close Virt-Manager

## Task VI: Test the Ability to Manage VMs from the CLI

1. Switch to virtual terminal 2 (VT2):  
**Ctrl+Alt+F2**
2. Login as **geeko**
3. Enter the following command to launch one of the non-running VMs:  
**virsh start VM\_NAME**
4. The VM should launch
5. Enter the following to see that the VM launched successfully:  
**virsh list**  
You should see the VM running
6. Wait a minute or two and shut down the VM:  
**virsh shutdown VM\_NAME**
7. Logout by entering the following:  
**exit**
8. Switch back to the GUI by pressing **Ctrl+Alt+F7**

**(End of Exercise)**



## 5.2 Use Unix Group Membership to Delegate Libvirt Administration

In this exercise you configure the Libvirt daemon to allow administration by anyone that is a member of the libvirt UNIX group. You then create a virtualization admin user that is a member of that group and test it.

### Objectives:

Task I: Create a Virtualization Admin User

Task II: Edit the libvirtd.conf File

Task III: Use virsh to Connect to the Local System

### Special Instructions and Notes:

The libvirt group must exist in the /etc/group file before this exercise can be performed.

### Task I: Create a Virtualization Admin User

1. In a terminal, enter the following commands create a group and a user for virtualization administration:  

```
groupadd -r virtadmins  
useradd -m -c "Virt Admin" -g virtadmins virtadmin  
passwd virtadmin
```

Enter **linux** for the password
2. Log out and then back in as the virtadmin user

### Task II: Edit the libvirtd.conf File

1. While logged in as the root user, in the text editor of your choice, open the `/etc/libvirt/libvirtd.conf` file to be edited
2. In the **UNIX socket access controls** section uncomment and edit the following line(s) to match:  

```
unix_sock_group = "virtadmins"  
unix_sock_ro_perms = "0777"  
unix_sock_rw_perms = "0770"
```
3. To disable local authentication by PolicyKit so the authorization can be handled by local UNIX group membership:  
In the **Authentication** section, uncomment and edit the following lines to match:  

```
auth_unix_ro = "none"
```

```
auth_unix_rw = "none"
```

4. Save the file and close the text editor
5. Enter the following command to restart the libvirt daemon:

```
rclibvirtd restart
```

### Task III: Use virsh to Connect to the Local System

1. Enter the following command to connect to the local machine:

for Xen:

```
virsh
```

For KVM:

```
virsh -c qemu:///system
```

You should be at a **virsh #** prompt on the local machine

2. Enter the following command to view information about the vHost:

```
nodeinfo
```

3. Enter the following command to view a list of the VMs that the vHost knows about:

```
list --all
```

4. Enter the following command to view a list of the Libvirt virtual networks configured on the vHost:

```
net-list --all
```

5. Enter the following command to disconnect from the **virsh #** prompt:

```
quit
```

You should now be back to the command prompt on your machine

**(End of Exercise)**

## 5.3 Use SASL+Digest-md5 to Delegate Libvirt Administration

In this exercise you configure the Libvirt daemon to only allow users authenticated via SASL with digest-md5 to gain access.

### Objectives:

- Task I: Edit the `sasl2/libvirt.conf` File
- Task II: Create an SASL User
- Task III: Enable SASL for the Libvirt Daemon
- Task IV: Enable SASL for the QEMU Libvirt Driver
- Task V: Use `virsh` to Connect to the Local System

### Special Instructions and Notes:

(none)

#### Task I: Edit the `sasl2/libvirt.conf` File

1. While logged in as the **root** user, in the text editor of your choice, open the `/etc/sasl2/libvirt.conf` file to be edited
2. Locate and uncomment the **`mech_list: digest-md5`** line as follows:  
**`mech_list: digest-md5`**
3. Save the file

#### Task II: Create an SASL User

1. In a terminal, enter the following command to create a SASL user for Libvirt  
**`saslpasswd2 -a libvirt libvirtadmin`**  
Enter **linux** for the password
2. Enter the following command to verify the user was added:  
**`sasldblistusers2 -f /etc/libvirt/passwd.db`**  
You should see the `virtadmin` user listed

#### Task III: Enable SASL for the Libvirt Daemon

In this task you enable Libvirt itself to require SASL authentication.

1. In the text editor of your choice, open the `/etc/libvirt/libvirtd.conf` file to be edited
2. In the **Authentication** section uncomment and edit the following line(s) to match:  
**`auth_unix_ro = "sasl"`**

```
auth_unix_rw = "sas1"
```

3. Save the file and close the text editor
4. Enter the following command to restart the libvirt daemon:

```
rc libvirtd restart
```

## Task IV: Enable SASL for the QEMU Libvirt Driver

In this task you enable VNC via TCP+SASL for the QEMU Libvirt driver.

1. In the text editor of your choice, open the `/etc/libvirt/qemu.conf` file to be edited
2. Locate and uncomment the following lines:

```
vnc_listen = "0.0.0.0"
```

```
vnc_sasl=1
```

3. Save the file

---

**Note:** You must stop and then restart any running VMs for this to take effect

---

## Task V: Use virsh to Connect to the Local System

1. Log out and then back in as the `virtadmin` user
2. Enter the following command to connect to the local machine:

For Xen:

```
virsh
```

For KVM:

```
virsh -c qemu:///system
```

You should be at a `virsh #` prompt on the local machine

3. Enter the following command to view information about the vHost:

```
nodeinfo
```

You should be prompted to authenticate. Enter the username and password for the `libvirtadmin` user created above

4. Enter the following command to view a list of the VMs that the vHost knows about:

```
list --all
```

You should not be prompted to authenticate again because your previous authentication is sufficient

5. Enter the following command to view a list of the Libvirt virtual networks configured on the vHost:

```
net-list --all
```

6. Enter the following command to disconnect from the `virsh #` prompt:

**quit**

You should now be back to the command prompt on your machine

7. Log out and then back in as the **geeko** user

8. Enter the following command to connect to the local machine:

For Xen:

**virsh**

For KVM:

**virsh -c qemu:///system**

9. Enter the following command to view information about the vHost:

**nodeinfo**

You should be rejected because the **geeko** user is not part of the **virtadmins** UNIX group that was previously configured to grant authorization to administer Libvirt.

10. Enter the following command to add the **geeko** user to the **virtadmins** group:

**su -**

(enter root password)

**groupmod -A geeko virtadmins**

**exit**

11. Enter the following command to reload the **geeko** user's groups memberships:

**newgrp -l**

12. Enter the following command to verify membership of the **virtadmins** group:

**groups**

13. Now enter the following command to connect to the local machine:

For Xen:

**virsh**

For KVM:

**virsh -c qemu:///system**

14. Enter the following command to view information about the vHost:

**nodeinfo**

You should now be prompted to authenticate. Enter the username and password for the **libvirtadmin** user created above

**(End of Exercise)**



## Section 6 Other Virtualization Management Tasks

This section covers other virtualization administration tasks such as pausing, saving and restoring virtual machines.

## 6.1 Pause and Save KVM VMs

In this exercise, you pause/unpause and save/restore a KVM VM.

### Objectives:

Task I: Pause and Unpause a Running VM with Virt-Manager

Task II: Suspend and Resume a Running VM with virsh

Task III: Save and Restore a Running VM

Task IV: Managedsave and Start a Running VM

### Special Instructions and Notes:

`VM_NAME=` \_\_\_\_\_

`VM_DIR=` \_\_\_\_\_

### Task I: Pause and Unpause a Running VM with Virt-Manager

1. If you are not already logged in as the root user, enter `su -` to become root.
2. Using either `virt-manager` or the `virsh` command, launch a virtual machine
3. If not already running, launch `virt-manager` and open the graphical console of the virtual machine (using built-in `virt-manager` VNC viewer).
4. Open a terminal window. In the terminal window run the following command:  

```
watch virsh list --all
```

Arrange the windows so that you can see the terminal window, the graphical console and the `virt-manager` window at the same time.
5. In the VM's graphical console, click the **Pause** button to pause the virtual machine. You should see that the virtual machine is paused in the `virt-manager` window.
6. In the terminal window running the `virsh list` command, you should see the state of the virtual machine is paused
7. To unpause the VM, click the **Restore** button in the `virt-manager` graphical console. You should see that the virtual machine is running again

### Task II: Suspend and Resume a Running VM with virsh

1. If it is not already running launch the virtual machine
2. If not already running launch `virt-manager` and open the graphical console (built-in `virt-manager` VNC viewer) of the virtual machine
3. Open a terminal window. In the terminal window run the following command:



```
watch virsh list --all
```

4. At the command line of the VM server run the following command to suspend (pause) the virtual machine:

```
virsh suspend VM_NAME
```

5. You should see in both the virt-manager window and the terminal window running **virsh list** that the virtual machine is paused

6. To resume (unpause) the saved virtual machine enter the following command at the command line of the VM server:

```
virsh resume VM_NAME
```

You should see that the virtual machine is now running again

### Task III: Save and Restore a Running VM

1. If it is not already running launch the virtual machine
2. If not already running launch virt-manager and open the graphical console (built-in virt-manager VNC viewer) of the virtual machine

3. Open a terminal window. In the terminal window run the following command:

```
watch virsh list --all
```

4. At the command line of the VM server run the following command to save the virtual machine:

```
virsh save VM_NAME VM_DIR/checkpoint.chk
```

5. You should see in both the virt-manager window and the terminal window running **virsh list** that the virtual machine is no longer running

6. To view the saved memory image “checkpoint” file enter the following command at the command line of the VM server:

```
ls -l VM_DIR
```

You should see that a new memory image “checkpoint” file named checkpoint.chk exists in that directory

7. To restore the saved virtual machine enter the following command at the command line of the VM server:

```
virsh restore VM_DIR/checkpoint.chk
```

8. You should see that the virtual machine is now running again and in the same state it was in when it was saved

### Task IV: Managedsave and Start a Running VM

1. If it is not already running launch the virtual machine
2. If not already running launch virt-manager and open the graphical console (built-in

- virt-manager VNC viewer) of the virtual machine
3. Open a terminal window. In the terminal window run the following command:  
**watch virsh list --all**
  4. At the command line of the VM server run the following command to save the virtual machine:  
**virsh managedsave *VM\_NAME***
  5. You should see in both the virt-manager window and the terminal window running **virsh list** that the virtual machine is no longer running
  6. To view the saved memory image “checkpoint” file enter the following command at the command line of the VM server:  
**ls -l /var/lib/libvirt/qemu/save/**  
You should see that a new memory image file named ***VM\_NAME.save*** exists in that directory
  7. To restore the saved virtual machine enter the following command at the command line of the VM server:  
**virsh start *VM\_NAME***
  8. You should see that the virtual machine is now running again and in the same state it was in when it was saved

**(End of Exercise)**

## 6.2 Automatically Boot KVM VMs with libvirt-guests

In this exercise, you use the libvirt-guests script to automatically start and stop/save KVM VMs when the vHost starts and stops.

### Objectives:

Task I: Enabling VM Automatic Booting

Task II: Simulate a Shutdown and Restart of the vHost and VMs

### Special Instructions and Notes:

Before performing this exercise ensure that all Xen VMs on VM Server are not running.

**VM\_NAME=** \_\_\_\_\_

**VM\_CONFIG=** \_\_\_\_\_

### Task I: Enabling VM Automatic Booting

1. Verify that the libvirt-guests script starts automatically by entering the following at the command line:

```
insserv libvirt-guests
```

```
chkconfig -l libvirt-guests
```

You should see that the libvirt-guests script is set to start in runlevels 3 and 5

2. Enter the following command to make a VM “managed” by Libvirt:

```
virsh define VM_CONFIG
```

3. Enter the following command to start the VM:

```
virsh start VM_NAME
```

### Task II: Simulate a Shutdown and Restart of the vHost and VMs

1. To simulate a shutdown of the vHost, enter the following command:

```
rclibvirt-guests stop
```

2. Verify that the VM(s) are not running:

```
virsh list --all
```

3. View the memory image checkpoint file for the VM(s):

```
ls -l /var/lib/libvirt/qemu/save/
```

4. To simulate a startup of the vHost, enter the following command:

```
rclibvirt-guests start
```

5. After the libvirt-guests script finishes, check to see if the VM(s) are started by

either using virt-manager or the **virh list** command.

6. Shutdown the VM and remove it from being managed by Libvirt:

```
virsh shutdown VM_NAME
```

```
virsh undefine VM_NAME
```

**(End of Exercise)**

## **Section 7 Configure Virtual Hardware for KVM Virtualization**

This section covers configuring virtual hardware for KVM virtualization.

## 7.1 Configure a Virtual Serial Console

In this exercise you configure a virtual serial console for a full virtual machine

### Objectives:

Task I: Configure a Virtual Serial Device

Task II: Configure Access to the Serial Console in the VM

Task III: Connect to the Virtual Serial Console

### Special Instructions and Notes:

Use the following value(s) in this exercise:

**VM\_NAME**= \_\_\_\_\_

### Task I: Configure a Virtual Serial Device

1. Launch Virt-manager:  
**Alt+F2**  
**virt-manager**
2. Connect to the **localhost**, select the **VM\_NAME** virtual machine and click **Open**:
3. On the virtual machine window click Show virtual hardware details (blue circle with an i)
4. Click **Add Hardware**
5. On the New Virtual Hardware screen, in the left pane, select **Serial**
6. In the right pane, from the **Device Type** drop-down list select **Pseudo TTY (pty)** and then click **Finish**

You should see a new serial device listed in the left pane

### Task II: Configure Access to the Serial Console in the VM

1. Launch the **VM\_NAME** virtual machine
2. When it is booted up log in as the root user
3. In the text editor of your choice open the **/etc/inittab** file
4. Locate the virtual terminal section and uncomment the **S0** line as follows:  
**S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102**
5. Save the file
6. Open the **/etc/securetty** file
7. Add the following to the end of the file to allow the root user to log in to the serial

console on ttyS0:

**ttyS0**

8. Save the file and close the text editor
9. Enter the following command to have init reread its configuration file and enable the serial console:

```
init q
```

### **Task III: Connect to the Virtual Serial Console**

1. Back in the vHost, open a terminal and if not already logged in as the root user use **su -** to become root
2. Enter the following command to connect to the virtual serial console of the virtual machine:  

```
virsh console VM_NAME
```
3. Press **Enter** to have the serial console present you with a login prompt:
4. Log in to the VM as the root user
5. Press the following key combination to disconnect from the virtual serial console:

**Ctrl+]** (Ctrl + right square bracket)

You should now be back at a terminal prompt in the vHost

**(End of Exercise)**

## 7.2 Configure a Virtual Watchdog Device

In this exercise you configure a virtual watchdog device in a KVM virtual machine.

### Objectives:

Task I: Configure a Virtual Watchdog Device

Task II: Verify the Watchdog Device in the VM

### Special Instructions and Notes:

Use the following value(s) in this exercise:

**VM\_NAME**=\_\_\_\_\_

### Task I: Configure a Virtual Watchdog Device

1. Launch Virt-manager:  
**Alt+F2**  
**virt-manager**
2. Connect t to the **localhost**, select the **VM\_NAME** virtual machine and click **Open**:
3. On the virtual machine window click Show virtual hardware details (blue circle with an i)
4. Click **Add Hardware**
5. On the New Virtual Hardware screen, in the left pane, select **Watchdog**
6. In the right pane, select the following values from their respective drop-down lists and then click **Finish** :  
Model: **default**  
Action: **Forcefully power off the guest**  
You should see a new watchdog device listed in the left pane

### Task II: Verify the Watchdog Device in the VM

1. Power on the VM and log in as the root user
2. open a terminal and enter the following command to see that the watchdog device kernel module is loaded:  
**lsmod | grep i6300esb**  
You should see that the kernel module for the watchdog device is loaded

**(End of Exercise)**





## 7.3 Create a Libvirt Virtual Network with Virt-manager

In this exercise, you use virt-manager to create a virtual network.

### Objectives:

Task I: Create a Virtual Network

### Special Instructions and Notes:

(none)

### Task I: Create a Virtual Network

1. As the root user, launch the virt-manager utility:  
**Alt+F2**  
**gnomesu virt-manager**
2. If not connected to the localhost, double-click on the localhost entry for the currently running hypervisor
3. From the menu bar, select **Edit > Host Details**
4. On the Host Details window select the **Virtual Networks** tab
5. On the **Virtual Networks** tab, click the **plus sign** in the bottom left corner to add a new virtual network
6. On the Creating a New Virtual Network screen click **Forward**
7. On the Naming a Virtual Network screen, in the **Name** field, enter **vnet1** and then click **Forward**
8. On the Choosing an Ipv4 Address Space screen, enter the following in the Network field and then click **Forward**:  
**192.168.11.0/24**
9. On the Selecting the DHCP Range screen, enter or select the following and then click **Forward**:  
Enable DHCP: **(checked)**  
Start: **192.168.11.128**  
End: **192.168.11.254**
10. On the Connecting to Physical Network screen select the following and then click **Forward**:  
Forwarding to physical network: **(selected)**  
Destination: **Any physical device**

Mode: NAT

11. On the Ready to Create Network screen, review the settings and click **Finish**

12. Open a terminal window and enter the following command:

```
virsh net-list --all
```

You should see the new virtual network listed

13. Enter the following command to view the XML definition of the virtual network:

```
virsh net-dumpxml vnet1
```

**(End of Exercise)**

## 7.4 Configure a Virtual Network with libvirt

In this exercise, you use libvirt to configure a virtual network to start automatically when the libvirt daemon starts.

### Objectives:

Task I: Create a Virtual Network Definition File

Task II: Activate a Virtual Network with virsh

Task III: Enable the Virtual Network to Start Automatically

### Special Instructions and Notes:

(none)

### Task I: Create a Virtual Network Definition File

1. If not already logged in as root, open a terminal window and enter `su -` to become root.
2. In the text editor of your choice (as root) open or create the `/tmp/vnet2` file to be edited.
3. Type the following into the file:

```
<network>
  <name>vnet2</name>
  <bridge name='virbr2' stp='off' forwardDelay='0' />
  <forward mode='nat' />
  <ip address='192.168.12.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.12.128' end='192.168.12.254'
    />
    </dhcp>
  </ip>
</network>
```

4. Save the file and close the text editor

### Task II: Activate a Virtual Network with virsh

1. In a terminal window, while logged in as root, enter the following command to view the existing virtual networks:

```
virsh net-list --all
```

You should not see the network defined in the configuration file

2. Enter the following command to view the active network bridges:

```
brctl show
```

You should not see the virtual network here either.

3. Enter the following command to define the virtual network in Libvirt:

```
virsh net-define /tmp/vnet2.xml
```

4. List the virtual networks again:

```
virsh net-list --all
```

You should now see the virtual network defined but not activated

5. Enter the following command in the terminal windows to activate the virtual network:

```
virsh net-start vnet2
```

You should see a message stating that the virtual network named **vnet2** was started

6. Enter the following command to verify that the virtual network was created:

```
virsh net-list
```

You should see a network named **vnet2** in the list

7. Enter the following command to view the active network bridges:

```
brctl show
```

You should see a new bridge named **virbr2** in the list

8. Enter the following command to deactivate the virtual network:

```
virsh net-destroy vnet2
```

9. Enter the following commands to see that the network is gone:

```
virsh net-list
```

```
brctl show
```

You should no longer see the virtual network or bridge

### Task III: Enable the Virtual Network to Start Automatically

1. View the contents of the `/etc/libvirt/qemu/networks/autostart` by entering the following command:

```
ls -l /etc/libvirt/qemu/networks/autostart
```

You should see that the directory is empty

2. Enter the following command at the command line to enable the network named `vnet2` to start automatically:

```
virsh net-autostart vnet2
```

3. Again, view the contents of the `/etc/libvirt/qemu/networks/autostart` directory:

```
ls -l /etc/libvirt/qemu/networks/autostart
```

You should see a symbolic link to the **vnet2.xml** file in the parent directory. When the libvirt daemon starts, this network will be activated automatically.

---

**NOTE:** You could have also manually created the symbolic link as well.

---

4. Restart the libvirt daemon:

```
rclibvirtd restart
```

5. Enter the following commands to see that the network was activated:

```
virsh net-list
```

```
brctl show
```

You should see that the virtual network and bridge are now active.

**(End of Exercise)**

## 7.5 Snapshot a QCOW2 Virtual Disk

In this exercise you use the `vm-snapshot-disk` command to create a copy on write (COW) snapshot of a QCOW2 disk image.

### Objectives:

Task I: Create Snapshots of a QCOW2 Disk

Task II: Revert to and Delete Snapshots

### Special Instructions and Notes:

(none)

### Task I: Create Snapshots of a QCOW2 Disk

1. While logged in as the root user, open a terminal and `cd` into the virtual machine directory

2. Enter the `ls -l` command to view the current files in the virtual machine directory

You should see some configuration files and a single disk image name

**disk0.qcow2**

3. Enter the following command to create a snapshot of the disk image:

```
vm-snapshot-disk create disk=disk0.qcow2
```

4. Enter the `ls -l` command again to see the new current list of files in the virtual machine directory

You should see the original disk images now named **disk0.qcow2.base**, a hard link to that disk named **disk0.qcow2.base.snap1** and the new COW snapshot disk

named **disk0.qcow2.base.snap1\_working**

5. Power on the VM and create a file named **/root/file1**

6. Power off the VM

7. Enter the following command to create a new snapshot of the disk:

```
vm-snapshot-disk create disk=disk0.qcow2
```

8. Enter the `ls -l` command again

You should see the previous `_working` snapshot is now named

**disk0.qcow2.base.snap1** and a new working snapshot file was created

9. Enter the following command to view the relationship of the new working snapshot:

```
qemu-img info disk0.qcow2.base.snap1.snap2._working
```

You should see that its backing disk is the **disk0.qcow2.base.snap1.snap2** file

10. Power on the VM and create another file named **/root/file2**

11. Power off the VM

12. Enter the following command to create a third snapshot:

```
vm-snapshot-disk create disk=disk0.qcow2
```

13. Power on the VM and create a new file named **/root/file3**

14. Power off the VM

15. Enter the following command to create a new snapshot branch:

```
vm-snapshot-disk branch disk=disk0.qcow2 snapname=snap1
```

16. Enter the **ls -l** command again to see the new current list of files in the virtual machine directory

You should see a new snapshot file with **snap1-1** in the name

17. Power on the VM and create a new file named **/root/file4**

Note which of the previously created files exist

18. Power off the VM

## Task II: Revert to and Delete Snapshots

1. Enter the following command to revert a previous snapshot:

```
vm-snapshot-disk revert disk=disk0.qcow2 snapname=snap1
```

2. Enter **ls -l** command to see the new current list of files in the virtual machine directory:

You should see the previous snapshot files that were based on snapshot1 are now gone

3. Enter the following command to remove the snapshots from the disk0.qcow2

```
vm-snapshot-disk remove disk=disk0.qcow2
```

4. Enter **ls -l** command to see the new current list of files in the virtual machine directory:

You should see all of the snapshot files are gone

**(End of Exercise)**



