Oracle Linux Advanced Administration

Activity Guide

D84498GC10
Edition 1.0
April 2014
D86119

ORACLE

Author

Craig McBride

Technical Contributors and Reviewers

Michele Dady, Avi Miller, Elena Zannoni, Wim Coekaerts, Al Flournoy, Harald Van Breederode, Joel Goodman, Manish Kapur, Yasar Akthar, Ozgur Yuksel, Antoinette O'Sullivan, Gavin Bowe, Nick Alcock, Dwight Engen, Wayne Lewis, Herbert Van Den Bergh, Tim Hill, Kris Van Hees, John Haxby

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Course Introduction 1-1
Practices for Lesson 1: Course Introduction 1-2
Practice 1-1: Exploring the dom0 Environment 1-3
Practice 1-2: Starting, Stopping, and Listing Guests 1-10
Practice 1-3: Exploring the host01 VM 1-12
Practice 1-4: Exploring the host02 VM 1-15
Practice 1-5: Exploring the host03 VM 1-17
Practice 1-6: Logging Off from Your Student PC 1-19
Practices for Lesson 2: Network Addressing and Name Services 2-1
Practices for Lesson 2: Network Addressing and Name Services 2-2
Practice 2-1: Configuring a DHCP Server 2-3
Practice 2-2: Configuring a DHCP Client 2-5
Practice 2-3: DNS Configuration 2-8
Practices for Lesson 3: Authentication and Directory Services 3-1
Practices for Lesson 3: Authentication and Directory Services 3-2
Practice 3-1: Configuring an NIS Server 3-3
Practice 3-2: Configuring an NIS Client 3-9
Practice 3-3: Implementing NIS Authentication 3-11
Practice 3-4: Testing NIS Authentication 3-14
Practice 3-5: Auto-mounting a User Home Directory 3-16
Practice 3-6: Configuring an OpenLDAP Server 3-19
Practice 3-7: Implementing OpenLDAP Authentication 3-30
Practice 3-8: Authenticating from an OpenLDAP Client 3-33
Practices for Lesson 4: Web and Email Services 4-1
Practices for Lesson 4: Web and Email Services 4-2
Practice 4-1: Configuring Apache Web Server 4-3
Practices for Lesson 5: Installing Oracle Linux by Using Kickstart 5-1
Practices for Lesson 5: Installing Oracle Linux Using Kickstart 5-2
Practice 5-1: Using the Kickstart Configurator 5-3
Practice 5-2: Performing a Kickstart Installation 5-8
Practice 5-3: Using Rescue Mode 5-20
Practices for Lesson 6: Samba Services 6-1
Practices for Lesson 6 6-2
Practice 6-1: Installing the Samba Packages 6-3
Practice 6-2: Configuring a Samba Server 6-4
Practice 6-3: Accessing Samba Shares from a Client Host 6-9
Practice 6-4: Accessing a Linux Samba Share from a Windows System 6-13
Practices for Lesson 7: Advanced Software Package Management 7-1
Practices for Lesson 7: Advanced Software Package Management 7-2
Practice 7-1: Managing Yum Plug-Ins 7-3
Practice 7-2: Creating an RPM Package 7-16
Practice 7-3: Managing Software Updates with PackageKit 7-25
Practice 7-4: Working with Yum History and Yum Cache 7-34
Practices for Lesson 8: Advanced Storage Administration 8-1
Practices for Lesson 8: Advanced Storage Administration 8-2

Practice 8-1: Creating and Mounting a File System 8-3
Practice 8-2: Implementing Access Control Lists 8-6
Practice 8-3: Setting Disk Quotas 8-8
Practice 8-4: Encrypting a File System 8-12
Practice 8-5: Using kpartx 8-16
Practice 8-6: Exploring and Configuring Udev 8-20

Practices for Lesson 9: OCFS2 and Oracle Clusterware 9-1
Practices for Lesson 9: Clustering with OCFS2 9-2
Practice 9-1: Preparing for an OCFS2 Configuration 9-3
Practice 9-2: Verifying That the Required Software Is Installed 9-9
Practice 9-3: Configuring the Cluster Layout 9-11
Practice 9-4: Configuring and Starting the O2CB Cluster Stack Service 9-15
Practice 9-5: Creating an OCFS2 Volume 9-17
Practice 9-6: Mounting an OCFS2 Volume 9-21
Practice 9-7: Tuning and Debugging OCFS2 9-26

Practices for Lesson 10: iSCSI and Multipathing 10-1
Practices for Lesson 10: iSCSI and Multipathing 10-2
Practice 10-1: Exploring a Configured iSCSI Server 10-3
Practice 10-2: Modifying the iSCSI Server (Target) Configuration 10-7
Practice 10-3: Configuring an iSCSI Client (Initiator) 10-12
Practice 10-4: Configuring iSCSI Multipathing 10-21

Practices for Lesson 11: XFS File System 11-1
Practices for Lesson 11: XFS File System 11-2
Practice 11-1: Creating an XFS File System 11-3
Practice 11-2: Setting Quotas on an XFS File System 11-8
Practice 11-3: Backing Up and Restoring XFS File Systems 11-13

Practices for Lesson 12: Btrfs File System 12-1
Practices for Lesson 12: Btrfs File System 12-2
Practice 12-1: Creating a Btrfs File System 12-3
Practice 12-2: Working with Subvolumes and Snapshots 12-12
Practice 12-3: Recovering from Data Corruption 12-19
Practice 12-4: Creating a Btrfs root File System by Installing Oracle Linux from the UEK Boot ISO 12-23

Practices for Lesson 13: Control Groups (Cgroups) 13-1
Practices for Lesson 13: Managing Resources with Control Groups (Cgroups) 13-2
Practice 13-1: Exploring the Default Cgroup Hierarchy 13-3
Practice 13-2: Creating Cgroups in the Default Hierarchy 13-7
Practice 13-3: Creating a Custom Cgroup Hierarchy 13-11
Practice 13-4: Using Cgroups to Allocate CPU Resources 13-15

Practices for Lesson 14: Virtualization with Linux 14-1
Practices for Lesson 14: Virtualization with Linux 14-2
Practice 14-1: Preparing the Virtualization Host for KVM 14-3
Practice 14-2: Starting the Virtual Machine Manager and Preparing to Create a Virtual Machine 14-9
Practice 14-3: Creating a Virtual Machine 14-16
Practice 14-4: Managing Your Virtual Machine 14-28

Practices for Lesson 15: Linux Containers (LXC) 15-1
Practices for Lesson 15: Linux Containers 15-2
Practice 15-1: Completing Linux Container Prerequisites 15-3
Practice 15-2: Creating an Oracle Linux Container 15-7

Practice 15-3: Using lxc Commands 15-11
Practice 15-4: Creating a Container from an Oracle VM Template 15-18

Practices for Lesson 16: Security Enhanced Linux (SELinux) 16-1
Practices for Lesson 16: Security Enhanced Linux (SELinux) 16-2
Practice 16-1: Exploring SELinux........ 16-3
Practice 16-2: Configuring an SELinux Boolean 16-13
Practice 16-3: Configuring SELinux Context........ 16-17

Practices for Lesson 17: Core Dump Analysis........ 17-1
Practices for Lesson 17: Core Dump Analysis........ 17-2
Practice 17-1: Configuring Kdump 17-3
Practice 17-2: Creating a Core Dump File........ 17-14
Practice 17-3: Preparing Your System to Analyze the vmcore........ 17-16
Practice 17-4: Using the crash Utility........ 17-20

Practices for Lesson 18: Dynamic Tracing with DTrace 18-1
Practices for Lesson 18: Dynamic Tracing with DTrace 18-2
Practice 18-1: Using sftp to Upload DTrace Packages........ 18-3
Practice 18-2: Installing the DTrace Packages 18-8
Practice 18-3: Using DTrace from the Command Line 18-12
Practice 18-4: Creating and Running D Scripts........ 18-20

Appendixes: Remote Access Options........ 19-1
Appendix 19-2
Appendix A: Using NX Client to Connect to dom0 19-3
Appendix B: Using NX Player to Connect to dom0 19-7
Appendix C: Using VNC (TightVNC) to Connect Directly to VM Guests........ 19-13
Appendix D: Using NoMachine Version 4 to Connect to dom0........ 19-16

Oracle Systems Learning Stream

A New Innovative Continuous Learning Solution

Ideal for supplementing what you learned in the classroom

Oracle Learning Streams

Videos from Oracle Experts

Continuously Updated Content

Connect Live with Instructors

ORACLE

Oracle Systems Learning Stream

Keep Your Skills Current through Continuous Learning

Expert Delivered

Access to hundreds of instructional videos delivered by Oracle subject matter experts

Training Across Your Infrastructure

For technical and business professionals looking to regularly broaden and deepen their knowledge

Continuously Refreshed Content

Covers technical, new features, how-to information and more…. on Oracle Hardware, Software, Operating System and Virtualization Solutions

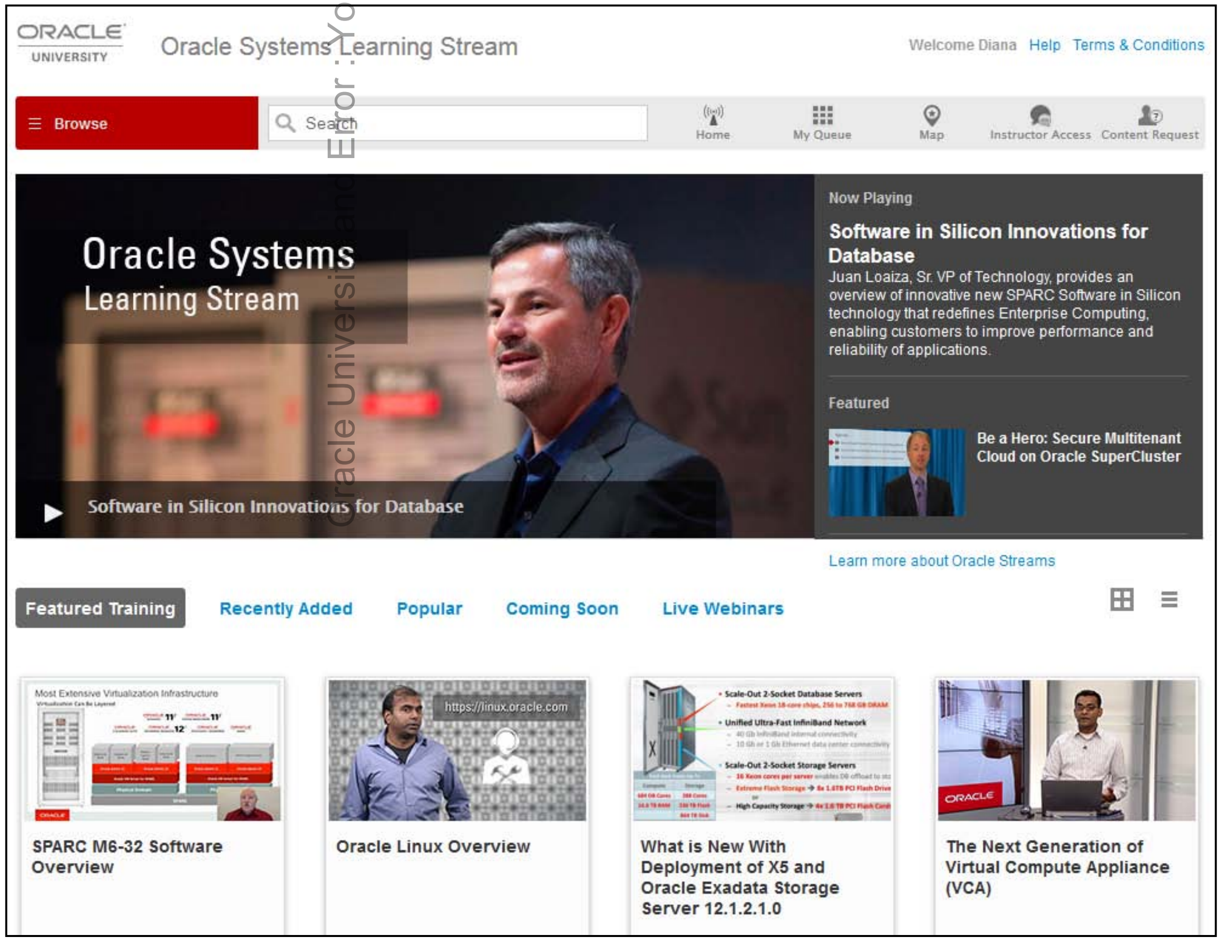
Request Topics that Interest You

Subscription Service

Preview the Oracle Systems Learning Stream

NOW!

education.oracle.com/streams/systems



ORACLE UNIVERSITY

ORACLE

Practices for Lesson 1: Course Introduction

Chapter 1

Practices for Lesson 1: Course Introduction

Practices Overview

In these practices, you will:

- Log in to your classroom PC and become familiar with the Oracle VM Server for x86 environment installed on your classroom PC
- Connect to the virtual machines used for the hands-on practices and become familiar with the VM guest configurations

Practice 1-1: Exploring the dom0 Environment

Overview

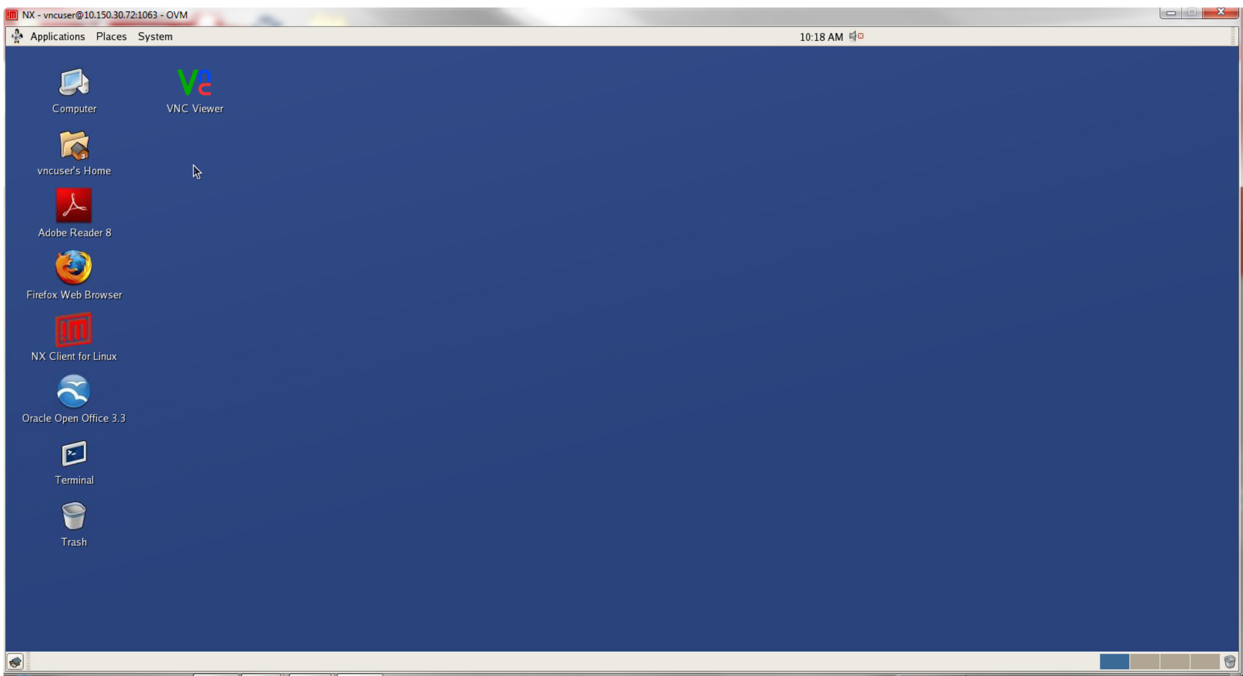
In this practice, you explore the **dom0** configuration and directory structure.

Assumptions

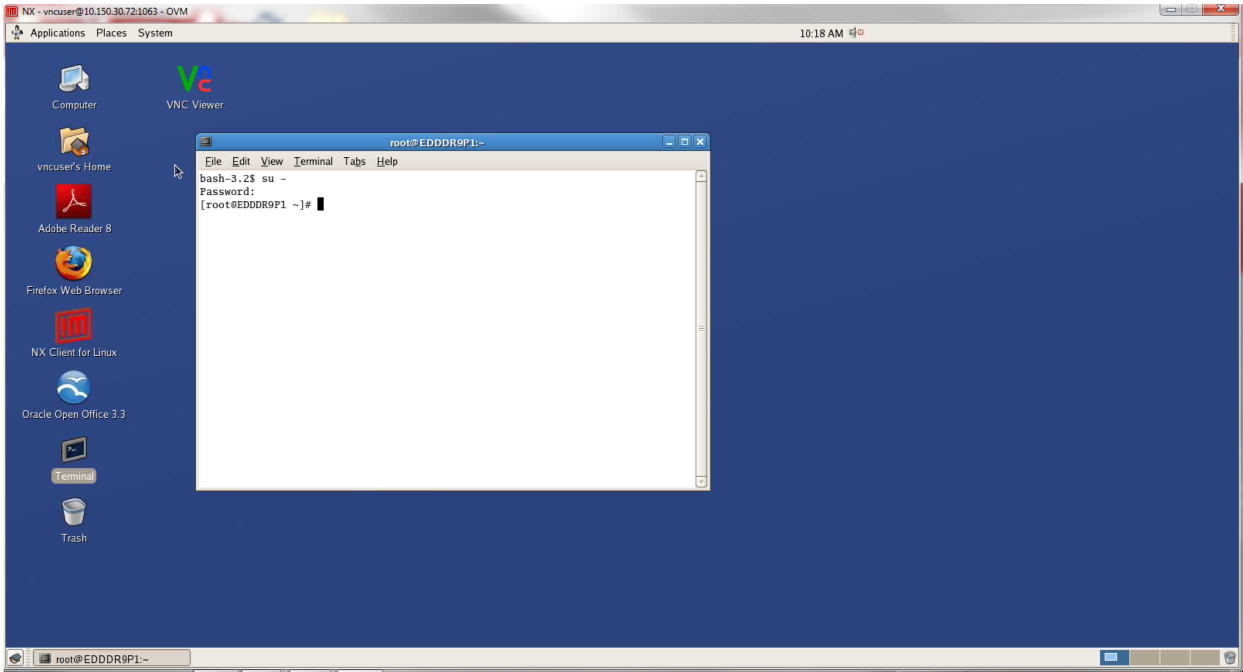
- Your instructor has assigned a student PC to you.
- Your student PC is running Oracle VM Server for x86 version 2.2.1.
- You are logged in to your student PC as `vncuser` with the password `vnctech`.
- The GNOME desktop is installed on **dom0**.
- There are three guests (virtual machines): **host01**, **host02**, and **host03**.
- All guest VMs have Oracle Linux 6.5 installed.

Tasks

1. Open a terminal window.
    - Begin this task from the **dom0** GNOME virtual desktop window as shown in the following screenshot:



   Double-click the **Terminal** icon on the GNOME desktop.
    - A terminal window opens.
2. Become the `root` user.
    - Enter the commands from an open terminal window as shown in the following screenshot:



Become the `root` user by using the `su -` command. The `root` password is `oracle`. Confirm that you are `root` by printing the user identity with the `whoami` command:

```
$ su -
Password: oracle
# whoami
root
```

3. Determine the operating system that is running on **dom0**.

   Use the `uname -a` command to display the operating system version.

```
# uname -a
Linux EDDDR9P1 2.6.18-128.2.1.4.25.el5xen #1 SMP Tue Mar 23
12:43:27 EDT 2010 i686 i686 i383 GNU/Linux
```

- In this example, the operating system is Linux.
- The Linux kernel is `2.6.18-128.2.1.4.25.el5xen`.
- The host name is `EDDDR9P1` (your host name is different).

4. Determine the network configuration of **dom0**.

   Use the `ifconfig -a` command to display the network configuration. Only partial output is shown.

```
# ifconfig -a
...
eth0      Link encap:Ethernet
          inet addr:10.150.30.72
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1
...
```

```
vif...    Link encap:Ethernet
...
virbr0    Link encap:Ethernet
          inet addr:192.0.2.1
...
virbr1    Link encap:Ethernet
          inet addr:192.168.1.1
...
xenbr0    Link encap:Ethernet
          inet addr:10.150.30.72
...
```

- In this example, the network interface for **dom0** is `eth0` and is assigned an IP address of `10.150.30.72`. The IP address of your system is different.
- The `lo` interface is a software loopback interface that identifies the `localhost`. It is always assigned an IP address of `127.0.0.1`.
- The `virbr0` interface is a `xen` bridge interface used by VM guests. It is assigned an IP address of `192.0.2.1`.
- The `virbr1` interface is a second `xen` bridge interface used by VM guests. It is assigned an IP address of `192.168.1.1`.
- Communication to the classroom network is through `xenbr0`, which has the same IP address as `eth0`.
- You also notice `vif<#>.<#>` entries. These are virtual interfaces that are tied to the VM/domU IDs. You can get the VM/domU IDs from the `xm list` command, which you run later in this practice.

5. Explore the `/OVS` directory structure on **dom0**.
   a. Explore the top level of the `/OVS` directory. (Only partial output is shown.)

```
# ls -l /OVS
lrwxrwxrwx /OVS -> /var/ovs/mount/...
# cd /OVS
# ls -l
drwxrwxrwx ... iso_pool
drwxr-xr-x ... lost+found
drwxr-xr-x ... OL6U5REPO
drwxrwxrwx ... publish_pool
drwxrwxrwx ... running_pool
drwxrwxrwx ... seed_pool
drwxrwxrwx ... sharedDisk
```

   - Note that `/OVS` is a symbolic link to the `/var/ovs/mount/...` directory.
   - There are seven directories in the `/OVS` directory.

   b. Explore the `/OVS/running_pool` directory:

```
# cd /OVS/running_pool
# ls -l
```

```
drwxr-xr-x ... host01
drwxr-xr-x ... host02
drwxr-xr-x ... host03
drwxr-xr-x ... host04
drwxr-xr-x ... host05
drwxr-xr-x ... host06
```

- The files needed to create the VMs are in separate directories in the `/OVS/running_pool` directory.
- This example shows that six VM directories exist, for VMs **host01**, **host02**, **host03**, **host04**, **host05**, and **host06**.
- The **host04** VM is pre-configured with access to Oracle's Public Yum Server. This VM is used in "Practices for Lesson 7: Advanced Software Package Management."
- The **host05** VM has the virtualization package groups installed. This VM is used in "Practices for Lesson 14: Virtualization with Linux."
- The **host06** VM does not have the operating system installed and is used only in "Practice 12-4: Creating a Btrfs `root` File System by Installing Oracle Linux from the UEK Boot ISO."

c. Explore the **host01** VM directory.

```
# cd /OVS/running_pool/host01
# ls –l
-rw-r--r-- ... system.img
-rw-r--r-- ... u01.img
-rw-r--r-- ... u02.img
-rwxr-xr-x ... vm.cfg
```

- The `system.img` file is the operating system virtual disk.
- The `u01.img` and `u02.img` files are utility virtual disks that are used in various practices in this course.
- The `vm.cfg` file is the configuration file for the virtual machine. This file is read when the virtual machine is created.

d. View the `vm.cfg` file.

```
# cat vm.cfg
name = 'host01'
builder = 'hvm'
memory = 1536
boot = 'cd'
disk = [ 'file:/OVS/running_pool/host01/system.img,hda,w',
         'file:/OVS/running_pool/host01/u01.img,hdb,w',
         'file:/OVS/running_pool/host01/u02.img,hdd,w',
         'file:/OVS/seed_pool/OracleLinux-R6-U5-Server-x86_64-
dvd.iso,hdc:cdrom,r']
vif = [ 'mac=00:16:3e:00:01:01, bridge=virbr0',
        'mac=00:16:3e:00:02:01, bridge=virbr1']
device_model = '/usr/lib/xen/bin/qemu-dm'
```

```
kernel = '/usr/lib/xen/boot/hvmloader'
vnc = 1
vncunused=1
vcpus = 1
timer_mode = 0
apic = 1
acpi = 1
pae = 1
serial = 'pty'
on_reboot = 'restart'
on_crash = 'restart'
usb = 1
usbdevice = 'tablet'
```

- Note that there are three virtual disks represented by the three `.img` files.
- Note that the Oracle Linux `dvd.iso` is mounted on a virtual CD ROM device.
- Note that there are two virtual network interfaces. The interface on the `virbr0` bridge is `eth0` and the interface on the `virbr1` bridge is `eth1`.

e. Explore the `/OVS/sharedDisk` directory:

```
# cd /OVS/sharedDisk
# ls -l
-rw-r--r-- ... physDisk1.img
-rw-r--r-- ... physDisk2.img
-rw-r--r-- ... physDisk3.img
-rw-r--r-- ... physDisk4.img
-rw-r--r-- ... physDisk5.img
```

- These virtual disk image files are used in “Practices for Lesson 10: iSCSI and Multipathing.”
  - In these practices, you configure **dom0** as an iSCSI target.
  - You configure a VM as an iSCSI initiator with access to these remote disks.
- The `physDisk1.img` file is also used as a shared disk (shared by all VM guests) in “Practices for Lesson 9: OCFS2 and Oracle Clusterware.”

f. Explore the `/OVS/seed_pool` directory:

```
# cd /OVS/seed_pool
# ls -l
drwxr-xr-x  ...  btrfs
drwxr-xr-x  ...  cgroup
drwxr-xr-x  ...  debug
drwxr-xr-x  ...  dtrace_rpms
drwxr-xr-x  ...  dtrace_scripts
drwxr-xr-x  ...  host07
-rw-r--r--  ...  OracleLinux-R6-U5-Server-x86_64-dvd.iso
-rwxr-xr-x  ...  physDisk1.tgz
```

```
-rwxr-xr-x  ...  physDisk2.tgz
-rwxr-xr-x  ...  physDisk3.tgz
-rwxr-xr-x  ...  physDisk4.tgz
-rwxr-xr-x  ...  physDisk5.tgz
drwxr-xr-x  ...  sfws
-rwxr-xr-x  ...  system01.tgz
-rwxr-xr-x  ...  system02.tgz
-rwxr-xr-x  ...  system03.tgz
-rwxr-xr-x  ...  system04.tgz
-rwxr-xr-x  ...  system05.tgz
-rwxr-xr-x  ...  system06.tgz
-rwxr-xr-x  ...  u01_01.tgz
-rwxr-xr-x  ...  u01_03.tgz
-rwxr-xr-x  ...  u02_01.tgz
-rwxr-xr-x  ...  u02_02.tgz
-rwxr-xr-x  ...  u02_03.tgz
drwxr-xr-x  ...  UEKR3_1
-rwxr-xr-x  ...  V41364-01.iso
-rwxr-xr-x  ...  V42906-01.zip
-rwxr-xr-x  ...  vm01.cfg
-rwxr-xr-x  ...  vm02.cfg
-rwxr-xr-x  ...  vm03.cfg
-rwxr-xr-x  ...  vm04.cfg
-rwxr-xr-x  ...  vm05.cfg
-rwxr-xr-x  ...  vm06.cfg
```

- This directory contains many files used to create the initial environment.
- Oracle Linux 6.5 is installed on the **host01**, **host02**, and **host03** VMs from the `OracleLinux-R6-U5-Server-x86_64-dvd.iso` file in this directory.
- Other files in this directory are used in various practices, for example, the `V41364-01.iso` file is the UEK Boot ISO that is used in "Practice 12-4: Creating a Btrfs `root` File System by Installing Oracle Linux from the UEK Boot ISO."

g. Explore the `/OVS/OL6U5REPO` directory:
   - Your system is configured as a local Yum repository.

```
# cd /OVS/OL6U5REPO
# ls
EFI    Packages      ResilientStorage
EULA   README-en     RPM-GPG-KEY
...
```

   - This directory contains the contents of the Oracle Linux 6 Update 5 base ISO.

h. View the contents of the following directory:

```
# ls -l /var/www/html/repo/OracleLinux/OL6/5/base/
lrwxrwxrwx ... x86_64 -> /OVS/OL6U5REPO
```

- Notice that the web server URL it is pointing to `OL6U5REPO`.
- A `.repo` file exists on each VM pointing to this Yum repository.

Practice 1-2: Starting, Stopping, and Listing Guests

Overview

In this practice, you use `xm` commands to list, create, and shut down virtual machines.

Assumptions

- You are logged on to **dom0**.
- You have a terminal window open.
- You are the `root` user.

Tasks

1. List all currently active guests, as well as **dom0** itself.

   Use the `xm list` command. The output shown here is a sample, the `ID` and `Time(s)` values will be different on your system.

   ```
   # xm list
   Name                 ID    Mem  VCPUs      State      Time(s)
   Domain-0              0   2048      2      r-----       281.1
   host01                1   1536      1      -b----       157.6
   host02                2   1536      1      -b----       159.0
   host03                3   1536      1      -b----        13.2
   ```

   - You have three guests (**host01**, **host02**, and **host03**) running.

2. Shut down a VM.

   Use the `xm shutdown –w <VM name>` command to shut down the **host01** VM. The `–w` option tells the system to wait until all services in the domain shut down cleanly. Run `xm list` to display the running VMs.

   - The `xm shutdown` command takes a few seconds to complete.

   ```
   # xm shutdown –w host01
   Domain host01 terminated
   All domains terminated
   # xm list
   Name                 ID    Mem  VCPUs      State      Time(s)
   Domain-0              0   2048      2      r-----       289.6
   host02                2   1536      1      -b----       159.0
   host03                3   1536      1      -b----        13.2
   ```

   - Notice that **host01** is no longer active.

3. Start a VM.

   Use the `xm create <config_file>` command to start the **host01** VM. The `<config_file>` is named `vm.cfg` and is located in the `/OVS/running_pool/<VM_name>` directory. Run `xm list` to display the running VMs.

   ```
   # cd /OVS/running_pool/host01
   # xm create vm.cfg
   Using config file "./vm.cfg".
   Started domain host01 (id=#)
   ```

```
# xm list
Name                    ID   Mem  VCPUs      State     Time(s)
Domain-0                 0  2048      2     r-----       304.5
host01                   4  1536      1     r-----        18.7
host02                   2  1536      1     -b----       159.0
host03                   3  1536      1     -b----        13.2
```

- Notice that **host01** is now active.
- The State column for **dom0** and **host01** shows ‘`r`’ (run state). The State column for **host02** and **host03** shows ‘`b`’ (blocked). The following describes these values:
  - `r`: The domain is currently running and healthy
  - `b`: The domain is blocked, and not running or runnable. This can be caused because the domain is waiting on IO (a traditional wait state) or has gone to sleep because there was nothing else for it to do.

Practice 1-3: Exploring the host01 VM

Overview

In this practice, you perform the following:

- Log in to **host01**.
- View the storage devices available on **host01**.
- View the network configuration on **host01**.
- View the Unbreakable Enterprise Kernel version on **host01**.
- View the Yum repository configuration on **host01**.

Assumptions

- You are logged on to **dom0** as the `root` user.
- The **host01** VM guest is running.

Tasks

1. Explore the **host01** VM guest.
   a. Use the `ssh` command to log in to **host01**.
      - Because this is the first time you have logged in using `ssh`, the command checks to make sure that you are connecting to the host that you think you are connecting to. Enter **`yes`**.
      - The `root` password is `oracle` (all lowercase).
      - If you get a message, "**ssh: connect to host host01 port 22: No route to host**", wait a few seconds to allow **host01** to boot and then run the `ssh host01` command again.

```
# ssh host01
The authenticity of host 'host01 (192.0.2.101)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host01,192.0,2,101' (RSA) to the
list of known hosts.
root@host01's password: oracle
[root@host01 ~]# hostname
host01.example.com
```

      - The `hostname` command confirms you have successfully logged in to **host01**.

   b. Use the `fdisk` command to view the storage devices.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *      1      64     512000   83   Linux
/dev/xvda2         64    1567   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes
Disk /dev/mapper/vg_host01-lv-root: 11.1 GB, 11068768256 bytes
Disk /dev/mapper/vg_host01-lv-swap: 1287 MB, 1287651328 bytes
```

- Three devices are available: /dev/xvda, /dev/xvdb, and /dev/xvdd.
- Do not run the following commands, this is information only:
  - The /dev/xvda disk device represents a 12 GB system image file created with the following command (in the /OVS/running_pool/host01 directory on **dom0**):

    ```
    # dd if=/dev/zero of=system.img bs=1M count=12288
    ```
  - The /dev/xvdb disk device represents a 10 GB utility image file created with the following command (in the /OVS/running_pool/host01 directory on **dom0**):

    ```
    # dd if=/dev/zero of=u01.img bs=1M count=10240
    ```
  - The /dev/xvdd disk device represents a 10 GB utility image file created with the following command (in the /OVS/running_pool/host01 directory on **dom0**):

    ```
    # dd if=/dev/zero of=u02.img bs=1M count=10240
    ```
- The /dev/xvda device has Oracle Linux 6.5 Basic Server packages installed plus:
  - System administration tools from the Servers package group
  - Unbreakable Enterprise Kernel (UEK) Release 3 Update 1 (3.8.13-26.1.1)
- This system disk uses LVM volumes for the root and swap partitions.

c. Use the ifconfig command to display the network interfaces.

```
# ifconfig -a
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:01
          inet6 addr:192.0.2.101 ...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:01
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 ...
...
```

- The system has two Ethernet network interfaces, eth0 and eth1.
- The eth0 interface is on the 192.0.2 subnet, and provides access to **dom0** and the other VM guest systems.
- The eth1 interface does not have an IP address yet but is configured in “Practices for Lesson 2: Network Addressing and Name Services.”
- The eth1 interface is configured on a private subnet, 192.168.1, and is used in “Practices for Lesson 9: OCFS2 and Oracle Clusterware.”

d. View the /etc/hosts file on **host01**.

- No changes are needed in this file.

```
# cat /etc/hosts
127.0.0.1       localhost.localdomain localhost
192.0.2.1       example.com              dom0
192.0.2.101     host01.example.com       host01
192.0.2.102     host02.example.com       host02
192.0.2.103     host03.example.com       host03
```

e. Use the `uname -r` command to determine your running kernel version.

```
# uname -r
3.8.13-26.1.1.el6uek.x86_64
```

- You are running the Unbreakable Enterprise Kernel (UEK) Release 3.1. The UEK became the default kernel beginning with Oracle Linux 5.5.

f. View the `/etc/yum/repos.d` directory.

```
# cd /etc/yum.repos.d
# ls
public-yum-ol6.repo  vm.repo
```

- Two `.repo` files exists, the Public Yum repository file for Oracle Linux 6 and a custom repository file, `vm.repo`, for the local Yum repository on **dom0**.

g. Use the `grep` command to view enabled repositories in both files.

```
# grep enabled *
public-yum-ol6.repo:enabled=0
...
vm.repo:enabled=1
```

- Only the `vm.repo` file contains an enabled repository (`enabled=1`).

h. Use the `cat` command to view the contents of `vm.repo`.

```
# cat vm.repo
[OL6U5Dom0]
Name="Oracle Linux 6 U5 Dom0 Repo"
baseurl=http://192.0.2.1/repo/OracleLinux/OL6/5/base/x86_64
enabled=1
gpgkey=http://192.0.2.1/repo/OracleLinux/OL6/5/base/x86_64/RPM-GPG-KEY-oracle
gpgcheck=1
```

- Notice that the `baseurl` references the local Yum repository on **dom0** (`192.0.2.1`).

i. Use the `exit` command to log off **host01**.

```
# exit
logout
Connection to host01 closed.
```

Practice 1-4: Exploring the host02 VM

Overview

In this practice, you perform the following:

- Log in to **host02**.
- View the storage devices available on **host02**.
- View the network configuration on **host02**.

Assumptions

- You are logged on to **dom0** as the `root` user.
- The **host02** VM guest is running.

Tasks

1. Explore the **host02** VM guest.
   a. Use the `ssh` command to log in to **host02**.
      - Because this is the first time you have logged in using `ssh`, the command checks to make sure that you are connecting to the host that you think you are connecting to. Enter **`yes`**.
      - The `root` password is `oracle` (all lowercase).

```
# ssh host02
The authenticity of host 'host02 (192.0.2.102)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host02,192.0.2.102' (RSA) to the
list of known hosts.
root@host02's password: oracle
[root@host02 ~]# hostname
host02.example.com
```

      - The `hostname` command confirms whether you have successfully logged in to **host02**.

   b. Use the `fdisk` command to view the storage devices.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes
/dev/xvda1   *      1      64     512000   83   Linux
/dev/xvda2         64    2611   20458496   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes
Disk /dev/mapper/vg_host02-lv-root: 18.8 GB, 18798870528 bytes
Disk /dev/mapper/vg_host01-lv-swap: 2147 MB, 2147483648 bytes
```

      - Three devices are available: `/dev/xvda`, `/dev/xvdb`, and `/dev/xvdd`
      - Do not run the following commands, this is information only:

- The /dev/xvda disk device represents a 20 GB system image file created with the following command (in the /OVS/running_pool/host02 directory on **dom0**):

  `# dd if=/dev/zero of=system.img bs=1M count=20480`
- The /dev/xvdb disk device represents a 10 GB shared disk image file created with the following command (in the /OVS/sharedDisk directory on **dom0**):

  `# dd if=/dev/zero of=physDisk1.img bs=1M count=10240`
- The /dev/xvdd disk device represents a 20 GB utility image file created with the following command (in the /OVS/running_pool/host02 directory on **dom0**):

  `# dd if=/dev/zero of=u02.img bs=1M count=20480`

- The /dev/xvda device has Oracle Linux 6.5 Desktop packages installed plus:
  - System administration tools from the Servers package group
  - Unbreakable Enterprise Kernel (UEK) Release 3 Update 1 (3.8.13-26.1.1)
- This system disk uses LVM volumes for the root and swap partitions.

c. Use the ifconfig command to display the network interfaces.

```
# ifconfig
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:02
          inet6 addr:192.0.2.102 ...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:02
          inet6 addr:192.168.1.102 ...
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 ...
...
```

- The system has two Ethernet network interfaces: eth0 and eth1
- The eth0 interface is on the 192.0.2 subnet, and provides access to **dom0** and the other VM guest systems.
- The eth1 interface is on a private subnet, 192.168.1, and is used in “Practices for Lesson 9: OCFS2 and Oracle Clusterware.”

The /etc/hosts, kernel version, and Yum configuration is the same on all three VM guests.

d. Use the exit command to log off **host02**.

```
# exit
logout
Connection to host02 closed.
```

Practice 1-5: Exploring the host03 VM

Overview

In this practice, you perform the following:

- Log in to **host03**.
- View the storage devices available on **host03**.
- View the network configuration on **host03**.

Assumptions

- You are logged on to **dom0** as the `root` user.
- The **host03** VM guest is running.

Tasks

1. Explore the **host03** VM guest.
   a. Use the `ssh` command to log in to **host03**.
      - Because this is the first time you have logged in by using `ssh`, the command checks to make sure that you are connecting to the host that you think you are connecting to. Enter **`yes`**.
      - The `root` password is `oracle` (all lowercase).

```
# ssh host03
The authenticity of host 'host03 (192.0.2.103)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host03,192.0.2.103' (RSA) to the
list of known hosts.
root@host03's password: oracle
[root@host03 ~]# hostname
host03.example.com
```

      - The `hostname` command confirms whether you have successfully logged in to **host03**.

   b. Use the `fdisk` command to view the storage devices. The **host03** VM has the same disk configuration as the **host01** VM.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *      1      64     512000   83   Linux
/dev/xvda2         64    1567   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes
Disk /dev/mapper/vg_host01-lv-root: 11.1 MB, 11068768256 bytes
Disk /dev/mapper/vg_host01-lv-swap: 1287 MB, 1287651328 bytes
```

      - Three devices are available: `/dev/xvda`, `/dev/xvdb`, and `/dev/xvdd`.
      - This is the same disk configuration as **host01**.
      - The `/dev/xvda` device has Oracle Linux 6.5 Desktop packages installed plus:

    - System administration tools from the Servers package group
    - Unbreakable Enterprise Kernel (UEK) Release 3 Update 1 (`3.8.13-26.1.1`)
  - This system disk uses LVM volumes for the `root` and `swap` partitions.

c. Use the `ifconfig` command to display the network interfaces.

```
# ifconfig
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:03
          inet6 addr:192.0.2.103 ...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:03
          inet6 addr:192.0.2.104 ...
...
eth2      Link encap:Ethernet HWaddr 00:16:3E:00:03:03
          inet6 addr:192.168.1.103 ...
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 ...
...
```

  - The system has three Ethernet network interfaces: `eth0`, `eth1`, and `eth2`.
  - The `eth0` and `eth1` interfaces are on the `192.0.2` subnet. These interfaces are used in “Practices for Lesson 10: iSCSI and Multipathing.”
  - The `eth2` interface is on a private subnet, `192.168.1`, and is used in “Practices for Lesson 9: OCFS2 and Oracle Clusterware.”

The `/etc/hosts`, kernel version, and Yum configuration is the same on all three VM guests.

d. Use the `exit` command to log off **host03**.

```
# exit
logout
Connection to host03 closed.
```

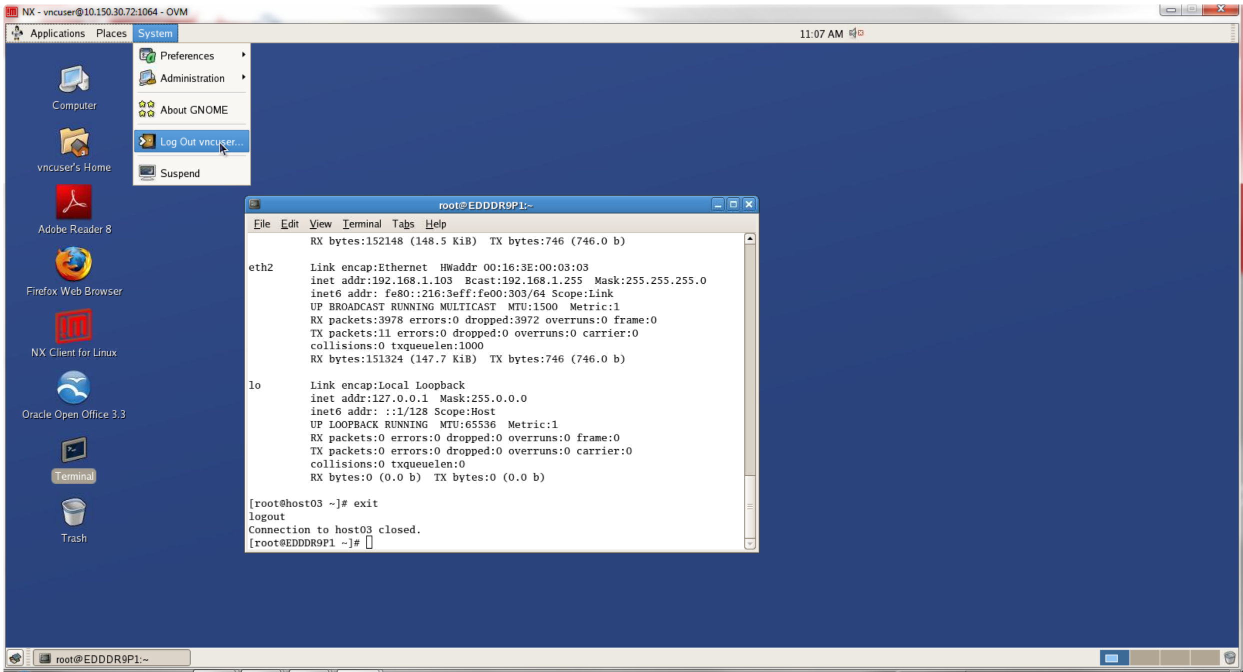
Practice 1-6: Logging Off from Your Student PC

Overview

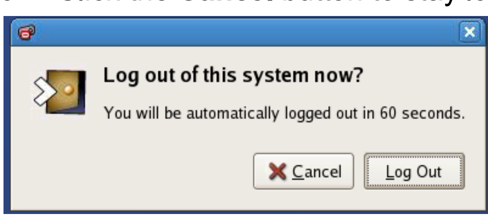
In this practice, you learn how to log off from your system.

Tasks

1. Learn how to log off your student PC.
   a. Open the **System** menu on the GNOME desktop.



   b. Select **Log Out vncuser** from the System menu.
      - You would click the **Log Out** button to log out.
      - Do not log out, however, until the end of each day of training.
   c. Click the **Cancel** button to stay logged in.



Practices for Lesson 2: Network Addressing and Name Services

Chapter 2

Practices for Lesson 2: Network Addressing and Name Services

Practices Overview

In these practices, you:

- Configure **host03** VM as a DHCP server and **host01** VM as a DHCP client
- Dynamically obtain an IP address for `eth1` on **host01**
- View the DNS server configuration on **dom0**
- Configure **host03** to use DNS for name-to-IP address resolution
- Test the DNS configuration

Practice 2-1: Configuring a DHCP Server

Overview

In this practice, you configure **host03** VM as a DHCP server.

Assumptions

You are the `root` user on **dom0**.

Tasks

1. Log in to the **host03** VM guest.

   Use the `ssh` command to log in to **host03**.

   - The `root` password is `oracle` (all lowercase).

```
[dom0]# ssh host03
root@host03's password: oracle
Last login: ...
[host03]#
```

2. Install the `dhcp` package on **host03** if necessary.

   a. Use the `rpm` command to check whether the `dhcp` package is installed.

```
# rpm -qa | grep dhcp
dhcp-common-...
```

   - In this example, only the `dhcp-common` package is installed, this package provides common files used by DHCP and the `dhclient` packages.
   - The `dhcp` package does need to be installed in this example.

   b. Use the `yum` command to install the `dhcp` package.

   - Answer "`y`" when prompted "**Is this ok [y/N]:**".
   - You are asked about the GPG key only the first time you use the `yum install` command.

```
# yum install dhcp
...
Transaction Summary
============================================================
Install       1 Package(s)
Total download size: 817 k
Installed size: 1.9 M
Is this ok [y/N]: y
...
Retrieving key from http://192.0.2.1/repo/OracleLinux/OL6/5/...
...
Importing GPG key ...
...
Is this ok [y/N]: y
...
Complete!
```

3. Use the `vi` editor to edit `/etc/dhcp/dhcpd.conf` as follows:

   Note: A pre-configured `dhcpd.conf` file exists on **dom0** in the `/OVS/seed_pool/sfws` directory.

   - You can edit the `dhcpd.conf` file as follows using the `vi` command, or you can use the `sftp` command and copy `/OVS/seed_pool/sfws/dhcpd.conf` from **dom0** to `/etc/dhcp/dhcpd.conf` on **host03**. See your instructor if you need help in using the `sftp` command.

```
# vi /etc/dhcp/dhcpd.conf
option subnet-mask              255.255.255.0;
option domain-name              "example.com";
option domain-name-servers      192.0.2.1;
option broadcast-address        192.168.1.255;
default-lease-time              21600;
max-lease-time                  43200;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.200 192.168.1.254;
}
```

4. Use the `vi` editor to edit `/etc/sysconfig/dhcpd` as follows:

```
# vi /etc/sysconfig/dhcpd
DHCPDARGS=eth2
```

   - This causes the server to start the service only on the `eth2` interface.

5. Start the `dhcpd` service.

```
# service dhcpd start
Starting dhcpd:                           [  OK  ]
```

6. Enable the `dhcpd` service to start at boot time.

```
# chkconfig dhcpd on
# chkconfig dhcpd --list
dhcpd    0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

Practice 2-2: Configuring a DHCP Client

Overview

In this practice, you:

- Configure **host01** VM as a DHCP client
- Obtain an IP address from the DHCP server (**host03**) for the `eth1` network interface

You begin this practice by opening a second terminal window on **dom0** and logging in to **host01** as the `root` user. You are already logged in as the `root` user to **host03** from Practice 2-1.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- You are currently logged in to **host03** (from Practice 2-1).
- The prompts in the solution section include either **host01** or **host03** to indicate which system to enter the command from.

Tasks

1. Log in to the **host01** VM guest from **dom0**.
   a. Open a second terminal window on **dom0**.
   b. From the second terminal window on **dom0**, use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
$ su -
Password: oracle
#
```

   c. As the root user on **dom0**, use the `ssh` command to log in to **host01**.
      - The `root` password is `oracle` (all lowercase).

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[host01]#
```

2. Use the `rpm` command to verify that the `dhclient` package is installed on **host01**.

```
[host01]# rpm -q dhclient
dhclient-4.1.1-38.P1.0.1.el6.x86_64
```

   - In this example, the package is already installed.
3. Ensure that IPv4 networking is enabled on **host01** by verifying that `NETWORKING=yes` in the `/etc/sysconfig/network` file.

```
[host01]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=host01.example.com
GATEWAY=192.0.2.1
```

   - IPv4 networking is enabled in this example.

4. Configure `eth1` on **host01** for DHCP.

   Use the `vi` editor and change `/etc/sysconfig/network-scripts/ifcfg-eth1`.

   - The only change needed is `ONBOOT=yes`.
   - The interface is configured to use DHCP by default.

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
HWADDR=00:16:3e:00:02:01
TYPE=Ethernet
UUID=...
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

5. Use the `ifconfig` command to display the network interfaces on **host01**.

```
[host01]# ifconfig -a
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:01
          inet addr:192.0.2.101...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:01
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1
...
```

- Notice that `eth1` is not configured.

6. From **host01**, use the `dhclient` command to request a lease for `eth1` from the DHCP server.

```
[host01]# dhclient eth1
```

7. Use the `ifconfig` command on **host01** to verify that `eth1` obtained an IP address.

```
[host01]# ifconfig eth1
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:01
          inet addr:192.168.1.200...
...
```

   - Notice `eth1` now has an IP address, `192.168.1.200`.

8. View the `/etc/resolv.conf` file on **host01**.

```
[host01]# cat /etc/resolv.conf
; generated by /sbin/dhclient-script
search example.com
nameserver 192.0.2.1
```

   - Notice that the nameserver and search domain were also obtained from the DHCP server.

9. View information about the lease on the client (**host01**).

```
[host01]# cat /var/lib/dhclient/dhclient.leases
lease {
  interface “eth1”;
  fixed-address 192.168.1.200;
  ...
}
```

10. View information about the lease on the server (**host03**).

```
[host03]# cat /var/lib/dhcpd/dhcpd.leases
...
lease 192.168.1.200 {
  starts ...
}
...
```

11. Use the `exit` command to log off **host01**.

```
[host01]# exit
logout
Connection to host01 closed.
[dom0]#
```

- In this window, you are logged in as the `root` user on **dom0**.
- Leave this window open for the next practice (Practice 2-3).

Practice 2-3: DNS Configuration

Overview

In this practice, you:

- View the DNS configuration on **dom0**
- Configure **host03** to use DNS
- Test the lookup functionality of DNS from **host03**

Assumptions

- **Dom0** is already configured as a DNS server.
- This practice is performed on **dom0** and on **host03** VM.
- You are logged in as the `root` user on **dom0** from one terminal window.
- You are logged in as the `root` user on **host03** from a second terminal window.
- The prompts in the solution section include either **dom0** or **host03** to indicate which system to enter the command from.

Tasks

1. Use the `rpm` command to verify that the `bind` package is installed on **dom0**.

```
[dom0]# rpm –qa | grep bind
ypbind-...
bind-utils-...
bind-...
bind-libs-...
```

   - In this example, the package is installed.

2. Use the `service` command to verify that the `named` service is started on **dom0**.

```
[dom0]# service named status
number of zones: 3
debug level: 0
...
server is up and running
named (pid ...) is running...
```

   - In this example, the service is running.

3. Use the `chkconfig` command to verify that the `named` service is configured to start at boot time on **dom0**.

```
[dom0]# chkconfig named --list
named      0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

   - In this example, the service is configured to start at boot time.

4. On **dom0**, view the main BIND configuration file, `/etc/named.conf`.

```
[dom0]# cat /etc/named.conf
...
options {
          directory  "/var/named";
```

```
...
zone "example.com" {
          type master;
          file "data/master-example.com";
          allow-update { key "rndckey"; };
          notify yes;
};
...
zone "2.0.192.in-addr.arpa" IN {
          type master;
          file "data/reverse-192.0.2";
          allow-update { key "rndckey"; };
          notify yes;
};
...
```

- This file lists location and characteristics of your domain's zone files.
- Notice that the zone file, `/var/named/data/master-example.com`, is defined.
- Notice that a reverse lookup zone file, `/var/named/data/reverse-192.0.2`, is also defined.

5. On **dom0**, view the `/var/named/data/master-example.com` zone file.

```
[dom0]# cat /var/named/data/master-example.com
...
dns                A       192.0.2.1
example.com        A       192.0.2.1
host01             A       192.0.2.101
host02             A       192.0.2.102
host03             A       192.0.2.103
host04             A       192.0.2.104
...
```

- This file defines IPv4 addresses ("`A`" records) for the DNS server, the DNS domain, and the three VM guest systems.

6. On **dom0**, view the `/var/named/data/reverse-192.0.2` file.

```
[dom0]# cat /var/named/data/reverse-192.0.2
...
1                  PTR       dns.us.oracle.com.
101                PTR       host01.example.com.
102                PTR       host02.example.com.
103                PTR       host03.example.com.
104                PTR       host04.example.com.
```

- This file defines "`PTR`" records for reverse name resolution.

7. On **host03**, use the `ping` command to contact **host01** and **host02**.

```
[host03]# ping host01
PING host01.example.com (192.0.2.101) 56(84) bytes of data.
64 bytes from host01.example.com (192.0.2.101): icmp_seq=1...
...
CTRL-C
[host03]# ping host02
PING host02.example.com (192.0.2.102) 56(84) bytes of data.
64 bytes from host02.example.com (192.0.2.102): icmp_seq=1...
...
CTRL-C
```

- You can successfully contact these systems by name, because `/etc/hosts` resolves names to IP addresses.

8. On **host03**, use the `vi` editor to edit the `/etc/hosts` file and comment out the lines for the VMs with a `#` sign as follows.

```
[host03]# vi /etc/hosts
#192.0.2.101   host01.example.com         host01
#192.0.2.102   host02.example.com         host02
#192.0.2.103   host03.example.com         host03
```

9. On **host03**, use the `ping` command to contact **host01** and **host02**.

```
[host03]# ping host01
ping: unknown host host01
[host03]# ping host02
ping: unknown host host02
```

- You cannot contact these systems by name now.

10. On **host03**, use the `vi` editor to add the following two entries to the `/etc/sysconfig/network-scripts/ifcfg-eth0` file.

```
[host03]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
DOMAIN=example.com
DNS1=192.0.2.1
```

11. On **host03**, use the `cat` command to display the contents of the `/etc/resolv.conf` file on **host03**.

```
[host03]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

- Notice that `NetworkManager` generated the `/etc/resolv.conf` file.
- The **DOMAIN** entry in the `ifcfg-eth0` file became the **search** entry in `resolv.conf`.
- The **DNS1** entry in the `ifcfg-eth0` file became the **nameserver** entry in `resolv.conf`.

12. On **host03**, use the `ping` command to contact **host01** and **host02**.

```
[host03]# ping host01
PING host01.example.com (192.0.2.101) 56(84) bytes of data.
64 bytes from host01.example.com (192.0.2.101): icmp_seq=1...
...
CTRL-C
[host03]# ping host02
PING host02.example.com (192.0.2.102) 56(84) bytes of data.
64 bytes from host01.example.com (192.0.2.102): icmp_seq=1...
...
CTRL-C
```

- You can successfully contact these systems by name, because DNS resolves names to IP addresses.
- The nameserver, `192.0.2.1`, for the `example.com` domain, stores the zone files that provide resolution.

13. On **host03**, update the name service switch configuration file, `/etc/nsswitch.conf`, to query the DNS server before querying `/etc/hosts` when resolving names to IP addresses.

```
[host03]# vi /etc/nsswitch.conf
...
hosts:    files dns (old entry)
hosts:    dns files (new entry)
...
```

14. On **host03**, use the `host` command to perform DNS lookups.
    a. Query DNS for the nameserver for the `example.com` domain.

```
[host03]# host -t NS example.com
example.com name server dns.example.com.
```

    b. Query DNS for the IP address that corresponds to `host01` system.

```
[host03]# host host01
host01.example.com has address 192.0.2.101
```

    c. Perform a reverse lookup by querying DNS for the domain name that corresponds to IP address `192.0.2.102`.

```
[host03]# host 192.0.2.102
102.2.0.192.in-addr-arpa domain name pointer host02.example.com
```

15. On **host03**, use the `vi` editor to edit the `/etc/hosts` file and remove the comment (`#` sign) from the entries previously commented out (in step 8) as follows.

```
[host03]# vi /etc/hosts
192.0.2.101    host01.example.com        host01
192.0.2.102    host02.example.com        host02
192.0.2.103    host03.example.com        host03
```

16. Use the `exit` command to log off **host03**.

```
[host03]# exit
logout
Connection to host03 closed.
[dom0]#
```

Practices for Lesson 3: Authentication and Directory Services

Chapter 3

Practices for Lesson 3: Authentication and Directory Services

Practices Overview

In these practices, you configure:

- NIS server
- NIS client and implement NIS authentication
- Auto-mounting of a user home directory
- OpenLDAP server and enable LDAP authentication
- OpenLDAP client and log in as an LDAP user

Practice 3-1: Configuring an NIS Server

Overview

In this practice, you:

- Configure an NIS server in preparation to implement NIS authentication
- Install the NIS service package (`ypserv`), start the service, and configure the service to automatically start at boot time
- Explore the various NIS files and directories, and create NIS maps

You must complete the NIS practices if you want to complete the SELinux practices in Lesson 16. These NIS practices are pre-requisites for the SELinux practice.

Assumptions

You are the `root` user on **dom0**.

Tasks

1. Log in to **host03** using `vncviewer`.
   a. From **dom0**, determine the VNC port number for **host03** by running the `xm list -l host03 | grep location` command.

```
[dom0]# xm list -l host03 | grep location
                (location 0.0.0.0:5903)
                (location 3)
```

   - The sample shown indicates that the port number is `5903`. This might not be true in your case.

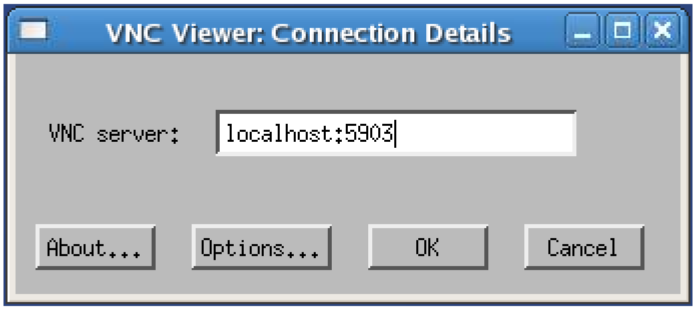
   b. From **dom0**, run the `vncviewer&` command.

```
[dom0]# vncviewer&
```

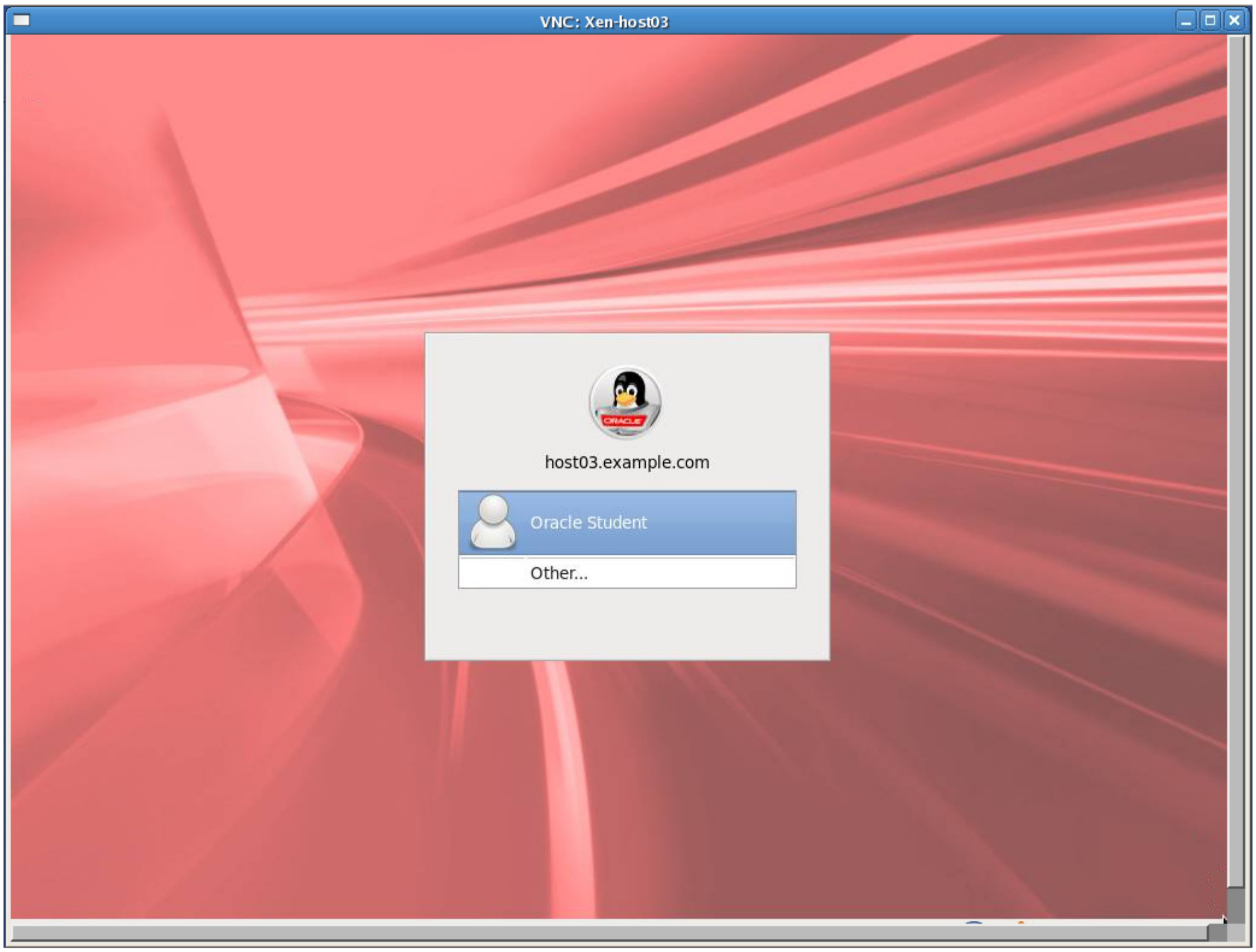
   - The **VNC Viewer: Connection Details** dialog box appears as shown:

   c. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list -l host03 | grep location` command.

   - For example, if the port number is `5903`, enter `localhost:5903` and click **OK**.



- The GNOME login screen appears:

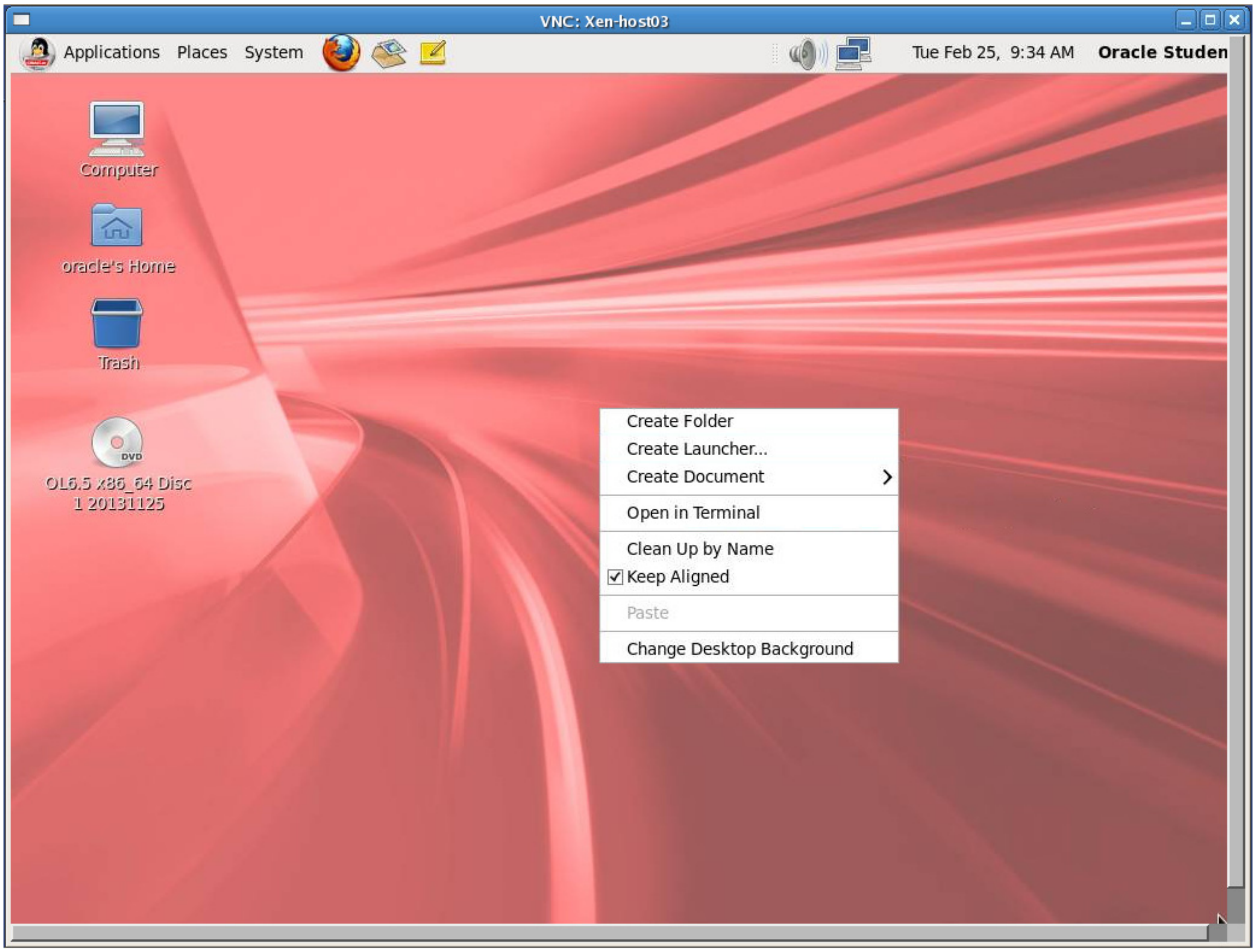


d. Click **Oracle Student** in the list of users. You are prompted for the password.

e. Enter `oracle` for the Password and click **Log In**.

- The GNOME desktop appears.

f. Right-click on the desktop to display the pop-up menu:



g. From the pop-up menu, click **Open in Terminal**.
   - A terminal window appears.

2. Become the `root` user on **host03**.
   a. In the terminal window, use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
$ su -
Password: oracle
#
```

3. Use the `yum` command to install the `ypserv` package on **host03**.

```
# yum install ypserv
...
Transaction Summary
==========================================================
Install        1 Package(s)
Total download size: 130 k
Installed size: 318 k
Is this ok [y/N]: y
...
Complete!
```

4. Use the `vi` editor to edit the `/etc/sysconfig/network` file and add the following line.

```
# vi /etc/sysconfig/network
NISDOMAIN=nis.example.com
```

5. Use the `vi` editor to create the following access rule in the `/etc/ypserv.conf` file.

```
# vi /etc/ypserv.conf
192.0.2.1/24 : * : * : none
```

- This rule allows access to the NIS server from any system on the `192.0.2` subnet.

6. Use the `vi` editor to create the `/var/yp/securenets` file with the following content:

```
# vi /var/yp/securenets
255.255.255.255      127.0.0.1
255.255.255.0        192.0.2.0
```

- These entries enhance system security by allowing access only from **localhost** (`127.0.0.1`) and from systems with IP addresses starting with `192.0.2`.

7. View the `/var/yp/Makefile` file (using `cat`, `less`, or `vi`) and locate the following block of lines:

```
# cat /var/yp/Makefile
...
all: passwd group hosts rpc services netid protocols mail \
# netgrp shadow publickey networks ethers bootparams printcap \
# amd.home auto.master auto.home auto.local. passwd.adjunct \
# timezone locale netmasks
```

- The "`all:`" target specifies which NIS maps to create.
- Do not make any changes to this file at this time.

8. Use the `chkconfig` command to configure the following NIS services to start at boot time:

- `ypserv`
- `ypxfrd`
- `yppasswdd`

```
# chkconfig --list | grep yp
ypbind      0:off  1:off  2:off  3:off  4:off  5:off  6:off
yppasswdd   0:off  1:off  2:off  3:off  4:off  5:off  6:off
ypserv      0:off  1:off  2:off  3:off  4:off  5:off  6:off
ypxfrd      0:off  1:off  2:off  3:off  4:off  5:off  6:off
# chkconfig ypserv on
# chkconfig ypxfrd on
# chkconfig yppasswdd on
# chkconfig --list | grep yp
ypbind      0:off  1:off  2:off  3:off  4:off  5:off  6:off
yppasswdd   0:off  1:off  2:on   3:on   4:on   5:on   6:off
ypserv      0:off  1:off  2:on   3:on   4:on   5:on   6:off
ypxfrd      0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

- The `ypbind` daemon binds NIS clients to an NIS domain.

- The `ypbind` service needs to run only on NIS clients.

9. Use the `service` command to start the three services.

```
# service ypserv start
Setting NIS domain name nis.example.com:        [  OK  ]
Starting YP server services:                    [  OK  ]
# service ypxfrd start
Starting YP map server:                         [  OK  ]
# service yppasswdd start
Starting YP passwd service:                     [  OK  ]
```

10. Use the `cd` command to change to the `/var/yp` directory. Use the `ls` command to view the contents.

```
# cd /var/yp
# ls
binding  Makefile  nicknames  securenets
```

- Notice that there are only four entries in this directory at this time.

11. The `ypinit` command is not in your search path. Use the `find` command to find the location of the `ypinit` command.

```
# which ypinit
/usr/bin/which: no ypinit in (...)
# find / -name ypinit
/usr/lib64/yp/ypinit
```

- In this example, the `ypinit` command is located in the `/usr/lib64/yp` directory.

12. Include the absolute path name and run `ypinit` with the `–m` option. Do not add any additional hosts; press **Ctrl + D** to complete the list.

```
# /usr/lib64/yp/ypinit -m
At this point, we have to construct a list of the hosts which
will run NIS servers. host03.example.com is in the list of NIS
server hosts. Please continue to add the names for the other
hosts, one per line. When you are done with the list, type a
<control D>.
    Next host to add:  host03.example.com
    Next host to add: <CTRL-D>
The current list of NIS servers looks like this:
host03.example.com
Is this correct? [y/n: y] ENTER
...
```

13. Use the `ls` command to view the contents of the `/var/yp` directory.
    - The current directory is `/var/yp` (you changed to this directory in step 10).

    ```
    # ls
    binding  Makefile  nicknames  nis.example.com  securenets
    ypservers
    ```

    - Notice the addition of the `nis.example.com` directory and the `ypservers` file.
14. Use the `cat` command to display the contents of the `ypservers` file.

    ```
    # cat ypservers
    host03.example.com
    ```

    - Notice that this file contains the host name of the NIS server.
15. Use the `ls` command to view the contents of the `nis.example.com` directory.

    ```
    # ls nis.example.com
    group.byid  mail.aliases  protocols.bynames  services.byname
    ...
    ```

    - Notice that this directory contains the NIS maps.

Practice 3-2: Configuring an NIS Client

Overview

In this practice, you configure an NIS client in preparation for implementing NIS authentication. You verify that the NIS client packages are installed, you configure the NIS client, and you start the NIS client service.

Assumptions

- This practice is performed on **host01**.
- There is one command in step 5 that might be necessary to run on **host03**.

Tasks

1. Log in to the **host01** VM guest from **dom0**.
   a. If necessary, open a second terminal window on **dom0**.
   b. From the second terminal window on **dom0**, use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
$ su -
Password: oracle
#
```

   c. As the `root` user on **dom0**, use the `ssh` command to log in to **host01**.
      - The `root` password is `oracle` (all lowercase).

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[host01]#
```

2. On **host01**, use the `rpm` command to verify that the required NIS client packages are installed.

```
# rpm –q yp-tools
yp-tools-2.9-12.el6.x86_64
# rpm –q ypbind
ypbind-1.20.4-30.el6.x86_64
```

   - In this example, the two required packages are already installed.
3. Use the `vi` editor to edit the `/etc/sysconfig/network` file. Add the following entry to set the NIS domain name (`NISDOMAIN`) to `nis.example.com`.

```
# vi /etc/sysconfig/network
NISDOMAIN=nis.example.com
```

   - Alternatively, you could use the `nisdomainname` command to set the NIS domain name to `nis.example.com`.

```
# nisdomainname nis.example.com
```

   - Setting the `NISDOMAIN` from the command line is not persistent across a reboot.

4. Use the `vi` editor to edit `/etc/yp.conf` and specify the NIS domain and NIS server.

```
# vi /etc/yp.conf
domain nis.example.com server host03.example.com
```

5. Use the `service` command to start the `ypbind` service.

```
# service ypbind start
Setting NIS domain: domain is 'nis.example.com'   [  OK  ]
Starting NIS service:                              [  OK  ]
Binding NIS service: ...................           [FAILED]
Shutting down NIS service:                         [  OK  ]
```

   a. If the `ypbind` service fails to start, disable the firewall on **host03** by running the following command.
      - Enter the following command on **host03**.

```
[host03]# service iptables stop
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Flushing firewall rules:                 [  OK  ]
iptables: Unloading modules:                       [  OK  ]
```

   b. After stopping the `iptables` service, use the `service` command to start the `ypbind` service on **host01**.

```
# service ypbind start
Setting NIS domain:                                [  OK  ]
Binding NIS service: .                             [  OK  ]
```

6. Run the `ypwhich` command to display the NIS server name.

```
# ypwhich
host03.example.com
```

Practice 3-3: Implementing NIS Authentication

Overview

In this practice, you use the Authentication Configuration Tool to implement NIS authentication.

Assumptions

- This practice is performed exclusively on **host03**.
- Ensure that you are using `vncviewer` to connect to **host03** and not using `ssh`.
- You are the `root` user on **host03**.

Tasks

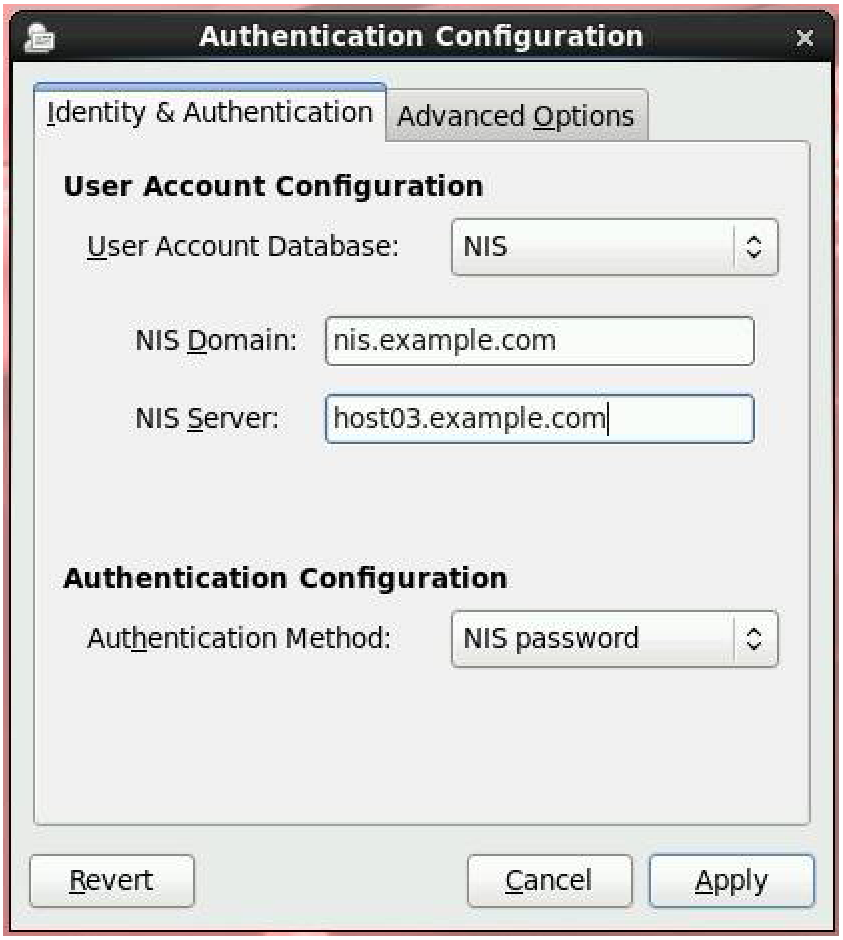
1. From **host03**, open the Authentication Configuration Tool by running the `system-config-authentication` command.

```
# system-config-authentication
```

- The GUI appears:

2. Make the following changes:
   a. Select `NIS` from the User Account Database drop-down list.
   b. Ensure that the NIS Domain is `nis.example.com`.
   c. Enter `host03.example.com` as the NIS Server.
   d. Leave the Authentication Method as `NIS password`.

- Ensure that your screen looks like the following:



- Alternatively, you could implement NIS authentication from the command line by using the following command:

```
# authconfig --enablenis --nisdomain nis.example.com --nisserver host03.example.com --update
```

3. Click **Apply** to save your changes.
   - After clicking Apply, the Authentication Configuration Tool closes and the following messages appear in your terminal window:

```
Starting NIS service:                                    [  OK  ]
Binding NIS service:                                     [  OK  ]
```

4. If either service fails to start, do the following:
   a. Use the `service` command to restart the `ypserv` service.

```
# service ypserv restart
Stopping YP server services:                        [  OK  ]
Starting YP server services:                        [  OK  ]
```

   b. Run the Authentication Configuration Tool again (repeat step 1).
   c. Ensure that the settings are correct (repeat step 2).
   d. Click **Apply** (repeat step 3).

5. Run the `authconfig --test` command to view the authentication settings.
   - You can pipe the output to `less` if desired.

```
# authconfig --test
caching is disabled
...
nss_nis is enabled
 NIS server = “host03.example.com”
 NIS domain = “nis.example.com”
...
```

Practice 3-4: Testing NIS Authentication

Overview

In this practice, you test NIS authentication by creating a new user on the NIS server and logging in as the new user from a remote system.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- You are logged in as the `root` user on each system.
- The prompts include either **host01** or **host03** to indicate which system to enter the command from.

Tasks

1. On **host03**, use the `useradd` and `passwd` commands to create a new user, `nis_user`. Assign a password of `password`. Ignore the “BAD PASSWORD” warning.

```
[host03]# useradd nis_user
[host03]# passwd nis_user
Changing password for user nis_user.
New password: password
BAD PASSWORD: it is based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
```

   - These commands create entries in the `/etc/passwd` and `/etc/shadow` files on the NIS server, **host03**.

2. On **host03**, update the NIS maps by running `ypinit –m` (include the absolute path name). Do not add any additional hosts; press **Ctrl + D** to complete the list.

```
[host03]# /usr/lib64/yp/ypinit -m
At this point, we have to construct a list of the hosts which
will run NIS servers. host03.example.com is in the list of NIS
server hosts. Please continue to add the names for the other
hosts, one per line. When you are done with the list, type a
<control D>.
    Next host to add:  host03.example.com
    Next host to add: <CTRL-D>
The current list of NIS servers looks like this:
host03.example.com
Is this correct? [y/n: y] ENTER
...
```

3. From **host01**, use the `vi` editor to edit the `/etc/nsswitch.conf` file. Change the following entries to query NIS maps first before querying local files.

```
[host01]# vi /etc/nsswitch.conf
passwd:   files                         (old entry)
shadow:   files                         (old entry)
group:    files                         (old entry)
passwd:   nis files                     (new entry)
shadow:   nis files                     (new entry)
group:    nis files                     (new entry)
```

4. From **host01**, log in as `nis_user` to test NIS authentication.
   a. Use the `logout` command to log out as `root` user.

```
[host01]# logout
Connection to host01 closed.
```

   b. From **dom0**, log in as `nis_user`. Password is `password`.

```
[dom0]# ssh nis_user@host01
nis_user@host01's password: password
Could not chdir to home directory /home/nis_user: No such file
or directory
[host01]$ hostname
host01.example.com
[host01]$ whoami
nis_user
[host01]$ pwd
/
```

- Notice that you were able to authenticate from the NIS server and log in from **host01**.
- However, the `nis_user` home directory is on the NIS server, **host03**.
- In the next practice, you export the `nis_user` home directory as an NFS share and auto-mount the home directory upon `nis_user` remote login.

Practice 3-5: Auto-mounting a User Home Directory

Overview

In this practice, you export a user home directory as an NFS share, and you configure auto-mounter on a client system to auto-mount the remote home directory on login.

Assumptions

- This practice is performed on **host01** and on **host03** VMs.
- The prompts include either **host01** or **host03** to indicate which system to enter the command from.

Tasks

1. From **host01**, use the `logout` command to log out as `nis_user`.

```
[host01]# logout
Connection to host01 closed.
```

2. From **dom0**, log in as `root` on **host01**. Password is `oracle`.

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[root@host01 ~]#
```

3. Use the `rpm` command to verify that the `autofs` package is installed.

```
[host01]# rpm –qa autofs
autofs-5.0.5-88.0.1.el6.x86_64
```

- In this example, the `autofs` package is already installed.

4. Use the `vi` editor to edit `/etc/auto.master` and add the following entry to the beginning of the file:

```
[host01]# vi /etc/auto.master
/home   /etc/auto.home
```

5. Use the `vi` editor to create the `/etc/auto.home` file with the following entry:

```
[host01]# vi /etc/auto.home
nis_user    -fstype=nfs    host03:/home/nis_user
```

6. Use the `service` command to restart the `autofs` service.

```
[host01]# service autofs restart
Stopping automount:                            [  OK  ]
Starting automount:                            [  OK  ]
```

7. From **host03**, use the `vi` editor to edit `/etc/exports` and add the following entry:

```
[host03]# vi /etc/exports
/home/nis_user   *(rw)
```

8. From **host03**, use the `service` command to start the `nfs` service.

```
[host03]# service nfs start
Starting NFS services:                         [  OK  ]
```

```
Starting NFS quotas:                         [  OK  ]
Starting NFS mountd:                         [  OK  ]
Starting NFS daemon:                         [  OK  ]
Starting RPC idmapd:                         [  OK  ]
```

9. From **host03**, use the `showmount` command to display the exported file systems.

```
[host03]# showmount -e
/home/nis_user *
```

- Notice that the `nis_user` home directory is now exported.

10. From **host01**, log in as `nis_user` to test auto-mounter.
    a. Use the `logout` command to log out as `root` user.

```
[host01]# logout
Connection to host01 closed.
```

    b. Log in as `nis_user`. Password is `password`.

```
[dom0]# ssh nis_user@host01
nis_user@host01's password: password
[nis_user@host01]$ hostname
host01.example.com
[nis_user@host01]$ whoami
nis_user
[nis_user@host01]$ pwd
/home/nis_user
```

- Notice that you were able to authenticate from the NIS server and log in, and auto-mount the `nis_user` home directory.

11. Use the `df` command to display the mounted file systems.

```
[nis_user@host01 ~]$ df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
...
host03:/home/nis_user
                11G  3.6G   6.0G   38%  /home/nis_user
```

- Notice that the `host03:/home/nis_user` file system is mounted on local file system `/home/nis_user`.

12. Use the `vi` editor to create a file and confirm read-write permissions.
    - Create any file name in the `nis_user` home directory with any content.
    - Save the file to confirm write permission to the file system.

```
[nis_user@host01 ~] $ vi test_file
Insert some text
Save the file by pressing <Esc> then :wq
```

13. Disable NIS on **host01**.
    a. Use the `logout` command to log out as `nis_user`.

```
[nis_user@host01]# logout
Connection to host01 closed.
```

    b. From **dom0**, log in as `root` on **host01**. Password is `oracle`.

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[root@host01 ~]#
```

    c. Use the `vi` editor to edit `/etc/sysconfig/network` and remove the following entry that sets the NIS domain name (`NISDOMAIN`) to `nis.example.com`.

```
# vi /etc/sysconfig/network
NISDOMAIN=nis.example.com
```

    d. Use the `vi` editor to edit `/etc/yp.conf` and remove the following entry that specifies the NIS domain and NIS server.

```
# vi /etc/yp.conf
domain nis.example.com server host03.example.com
```

    e. Use the `service` command to stop the `ypbind` service.

```
# service ypbind stop
Shutting down NIS service:                              [  OK  ]
```

14. Disable automounter on **host01**.
    a. Use the `vi` editor to edit `/etc/auto.master` and remove the following entry that references home directories.

```
# vi /etc/auto.master
/home    /etc/auto.home
```

    b. Use the `rm` command to remove the `/etc/auto.home` file.

```
# rm /etc/auto.home
rm: remove regular file '/etc/auto.home'? y
```

    c. Use the `service` command to stop the `autofs` service.

```
# service autofs stop
Stopping automount:                                [  OK  ]
```

    d. Use the `logout` command to log out of **host01**.

```
# logout
Connection to host01 closed.
```

Practice 3-6: Configuring an OpenLDAP Server

Overview

In this practice, you:

- Configure an OpenLDAP server in preparation to implement LDAP authentication
- Install the OpenLDAP server package (`openldap-servers`) and the `migrationtools` package
- Configure the server by using the `slapd.conf` file and then use the `slaptest` command to convert the installation to the new format, which uses a configuration database instead of a configuration file
- Configure the base domain and test the LDAP server
- Migrate users and groups into the LDAP directory
- Modify `iptables` to allow access on port 389

Assumptions

- You are the `root` user on **host03**.
- You are connected to **host03** by using `vncviewer` and did not connect by using `ssh`.

Tasks

1. Add users and groups to **host03**.
   - This step populates the `/etc/passwd` and `/etc/group` files that are used later in this practice.

   a. Use the `useradd` command to add users as follows.

```
# useradd –c “Oracle Student1” student1
# useradd –u 555 –c “Oracle Student2” –s /bin/sh student2
# useradd –c “Oracle Student3” –s /bin/sh student3
# useradd new_user
```

   b. Use the `passwd` command to create a password (of `password`) for the `student1` user.

```
# passwd student1
Changing password for user student1.
New password: password
BAD PASSWORD: it is based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
```

   - Ignore the “BAD PASSWORD” warning, continuing to use `password` as the password.

   c. Use the `groupadd` command to add the `students` group.

```
# groupadd students
```

d. Use the `tail /etc/group` command to obtain the GID for the `students` group.

```
# tail /etc/group
...
students:x:558:
```

- The output shows that the GID for the `students` group is `558`.

e. Use the `usermod` command to add `oracle`, `student1`, and `student2` users to the `students` group.

```
# usermod -aG 558 oracle
# usermod -aG 558 student1
# usermod -aG 558 student2
# tail /etc/group
...
students:x:558:oracle,student1,student2
```

2. Use the `yum` command to install two packages on **host03**: `openldap-servers` and `migrationtools`.

```
# yum install openldap-servers migrationtools
...
Transaction Summary
============================================================
Install       3 Package(s)
Total download size: 2.2 M
Installed size: 5.1 M
Is this ok [y/N]: y
...
Complete!
```

3. Copy the `slapd.conf` template file to the `/etc/openldap` directory.

a. Use the `cd` command to change to the `/etc/openldap` directory. Use the `ls -l` command to display the contents of the directory.

```
# cd /etc/openldap
# ls -l
drwxr-xr-x. 2 root root ... certs
-rw-r--r--. 1 root root ... ldap.conf
drwxr-xr-x. 2 root root ... schema
drwx------. 3 ldap ldap ... slapd.d
```

- Notice that, in this version of OpenLDAP, there is no `slapd.conf` file.
- Instead, there is a configuration database which is located in the `slapd.d` directory.
- However, a `slapd.conf` file is provided, which can be edited and then converted to the database format.

b. Use the `cp` command to copy the `/usr/share/openldap-servers/slapd.conf.obsolete` file into the current directory and rename the copied file as `slapd.conf`.

```
# cp /usr/share/openldap-servers/slapd.conf.obsolete slapd.conf
```

c. Use the `ls -l` command of the current directory.

```
# ls -l
drwxr-xr-x. 2 root root ... certs
-rw-r--r--. 1 root root ... ldap.conf
drwxr-xr-x. 2 root root ... schema
-rw-r--r--. 1 root root ... slapd.conf
drwx------. 3 ldap ldap ... slapd.d
```

- Notice that the `slapd.conf` file now exists in the `/etc/openldap` directory.

4. Create an encrypted user password and update the `slapd.conf` file.

a. Use the `slappasswd` command to create an encrypted user password. Enter a password of `oracle`.

```
# slappasswd
New password: oracle
Re-enter new password: oracle
{SSHA}sqMERRqsv0yoZuvGnTwlYjP7JWjN6/4n
```

- Notice that the encrypted password is displayed. This is a sample only; yours is different.

b. Select the encrypted password and copy it into the buffer.

- Highlight the encrypted password as shown:

- With encrypted password highlighted, select **Edit > Copy** from the terminal window menu.

c. Use the `vi` editor to edit the `slapd.conf` file. Use the `:set nu` command to turn on line numbers.

```
# vi slapd.conf
...
:set nu
```

d. At around line 122, set the value of `rootpw` to the encrypted password created in step 3a.

- To paste contents of the buffer, get into insert mode in `vi`, and then select **Edit > Paste** from the terminal window menu.
- Press **Esc** to exit insert mode.
- Un-comment the line by removing the `#` character and the first space.
- The `rootpw` directive must start in the first column.

```
rootpw            {SSHA}sqMERRqsv0yoZuvGnTwlYjP7JWjN6/4n
```

- Do not exit the `vi` editor.

5. Update the domain component (`dc=`) values in the `slapd.conf` file. Replace occurrences of `dc=my-domain` with `dc=example`.
   a. At around line number 107, change `dc=my-domain` to `dc=example`.

```
database monitor
access to *
        by dn.exact="gidNumber=0+uidNumber=0, ...
        by dn.exact="cn=Manager,dc=example,dc=com" read
```

   b. At around line number 115, change `dc=my-domain` to `dc=example`.

```
suffix     "dc=example,dc=com"
```

   c. At around line number 117, change `dc=my-domain` to `dc=example`.

```
rootdn     "cn=Manager,dc=example,dc=com"
```

   d. Save the `slapd.conf` file and exit `vi`.
6. Copy the `DB_CONFIG` template file to the `/var/lib/ldap` directory and change the ownership.
   a. Use the `cp` command to copy the `/usr/share/openldap-servers/DB_CONFIG.example` file into the `/var/lib/ldap` directory and rename the copied file `DB_CONFIG`.

```
# cp /usr/share/openldap-servers/DB_CONFIG.example
/var/lib/ldap/DB_CONFIG
```

   b. Use the `ls -l` command to list the contents of the `/var/lib/ldap` directory.

```
# ls -l /var/lib/ldap
-rw-r--r--. 1 root root ... DB_CONFIG
```

   - Notice that the current owner and group is `root`. Both the owner and group need to be changed to `ldap`.

   c. Use the `chown -R` command to change both the owner and group of the `/var/lib/ldap` directory to `ldap`.

```
# chown -R ldap.ldap /var/lib/ldap
```

   d. Use the `ls -l` command to show the new owner and group.

```
# ls -l /var/lib/ldap
-rw-r--r--. 1 ldap ldap ... DB_CONFIG
```

   - Notice that the owner and group are now set to `ldap`.

7. Convert the configuration to the new format.
   a. Use the `rm -rf` command to remove contents of the `slapd.d` directory.

```
# rm -rf slapd.d/*
```

   b. Use the `echo "" | slapadd -f slapd.conf` command to initialize DB files for content in `/var/lib/ldap`. Use the `ls` command to list the contents of the `/var/lib/ldap` directory.
   - Without initializing the DB files, you get a `bdb_db_open` error when running the command in step 7d.
   - The `""` are two double-quotes with no spaces between them.

```
# echo "" | slapadd -f slapd.conf
The first database does not allow slapadd; using the first
available one (2)
# ls -l /var/lib/ldap
-rw-r--r--. 1 root root ... alock
-rw-------. 1 root root ... __db.001
...
-rw-r--r--. 1 ldap ldap ... DB_CONFIG
...
```

- Notice that there are now several files in `/var/lib/ldap`.
- Also notice that, with the exception of `DB_CONFIG`, the files are owned by `root`.

c. Use the `chown -R` command to change both the owner and group of the `/var/lib/ldap` directory to `ldap`.

```
# chown -R ldap.ldap /var/lib/ldap
```

d. Use the `slaptest` command to convert the configuration to the new format.

```
# slaptest -f slapd.conf -F slapd.d
config file testing succeeded
```

e. Use the `ls -l` command to list the contents of the `slap.d` directory.

```
# ls -l slapd.d
drwxr-x---. 1 root root ... cn=config
-rw-------. 1 root root ... cn=config.ldif
```

- Notice that the current owner and group is `root`. Both the owner and group need to be changed to `ldap`.

f. Use the `chown -R` command to change both the owner and group of the `slapd.d` directory to `ldap`.

```
# chown -R ldap.ldap slapd.d
```

8. Use the `service` command to start the `slapd` service.

```
# service slapd start
Starting slapd:                                  [  OK  ]
```

9. Configure the base domain and test the LDAP server.

a. Use the `vi` editor to create the `base.ldif` file as follows.

- The `.ldif` extension begins with the lowercase letter `l`, not the number `1`.

Note: A sample `base.ldif` file exists on **dom0** in the `/OVS/seed_pool/sfws` directory.

- You can create the `base.ldif` file as follows using the `vi` command, or you can use the `sftp` command and copy `/OVS/seed_pool/sfws/base.ldif` from **dom0** to `/etc/openldap/base.ldif` on **host03**. See your instructor if you need help in using the `sftp` command.

```
# vi base.ldif
dn: dc=example,dc=com
dc: example
objectClass: top
```

```
objectClass: domain

dn: ou=People,dc=example,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

dn: ou=Group,dc=example,dc=com
ou: Group
objectClass: top
objectClass: organizationalUnit
```

b. Use the `ldapadd` command to import the base information to the LDAP directory. The LDAP password is `oracle`.

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f base.ldif
Enter LDAP Password: oracle
adding new entry "dc=example,dc=com"

adding new entry "ou=People,dc=example,dc=com"

adding new entry "ou=Group,dc=example,dc=com"
```

c. Use the `ldapsearch` command to test the LDAP server.

```
# ldapsearch -x -b "dc=example,dc=com"
...
# example.com
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain

# People, example.com
dn: ou=People,dc=example,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

# Group, example.com
dn: ou=Group,dc=example,dc=com
ou: Group
objectClass: top
objectClass: organizationalUnit

# search result
```

```
search: 2
result: 0 Success

# numResponses: 4
# numEntries: 3
```

10. Update the `migrate_common.ph` file for correct domain.
    a. Use the `vi` editor to edit the `/usr/share/migrationtools/migrate_common.ph` file. Use the `:set nu` command to turn on line numbers.

```
# vi /usr/share/migrationtools/migrate_common.ph
...
:set nu
```

    b. At around line number 71, set the value of `$DEFAULT_MAIL_DOMAIN` to `example.com`.

```
$DEFAULT_MAIL_DOMAIN = "example.com";
```

    c. At around line number 74, change `dc=padl` to `dc=example`.

```
$DEFAULT_BASE = "dc=example,dc=com";
```

    d. Save the `migrate_common.ph` file and exit `vi`.
11. Migrate the users.
    a. Use the `grep` command to list users in the `/etc/passwd` file with UID in the `500` range.
       - The purpose of step 3 was to populate this file as shown.
       - Do not be concerned if your entries do not match exactly.

```
# grep ":5[0-9][0-9]" /etc/passwd
oracle:x:500:500:Oracle Student:/home/oracle:/bin/bash
nis_user:x:501:501::/home/nis_user:/bin/bash
student1:x:502:502:Oracle Student1:/home/student1:/bin/bash
student2:x:555:555:Oracle Student2:/home/student2:/bin/sh
student3:x:556:556:Oracle Student3:/home/student3:/bin/sh
new_user:x:557:557::/home/new_user:/bin/bash
```

    b. Run the same command but redirect the output to `passwd`.

```
# grep ":5[0-9][0-9]" /etc/passwd > passwd
```

    c. Run the `migrate_passwd.pl` command to migrate user information in the `passwd` file into an LDAP format. Redirect the output to `users.ldif`.
       - Use the absolute path name, because the `/usr/share/migrationtools` directory is not in your path.

```
# /usr/share/migrationtools/migrate_passwd.pl passwd >
users.ldif
```

    d. Use the `ldapadd` command to import the user information to the LDAP directory. The LDAP password is `oracle`.

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f users.ldif
Enter LDAP Password: oracle
```

```
adding new entry "uid=oracle,ou=People,dc=example,dc=com"

adding new entry "uid=nis_user,ou=People,dc=example,dc=com"

adding new entry "uid=student1,ou=People,dc=example,dc=com"

adding new entry "uid=student2,ou=People,dc=example,dc=com"

adding new entry "uid=student3,ou=People,dc=example,dc=com"

adding new entry "uid=new_user,ou=People,dc=example,dc=com"
```

e. Use the `ldapsearch` command to display the new `oracle` user entry in the LDAP server.
   - The common name (`cn`) is "`Oracle Student`".

```
# ldapsearch -x "cn=Oracle Student" -b "dc=example,dc=com"
...
# oracle, People, example.com
dn: uid=oracle,ou=People,dc=example,dc=com
uid: oracle
cn: Oracle Student
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0...
shadowLastChange: ...
shadowMin: 0
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 500
gidNumber: 500
homeDirectory: /home/oracle
gecos: Oracle Student

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

12. Migrate the user groups.
    a. Use the `grep` command to list groups in the `/etc/group` file with GID in the `500` range.
       - This was the purpose of step 3, to populate this file as shown.
       - Do not be concerned if your entries do not match exactly.

```
# grep ":5[0-9][0-9]" /etc/group
oracle:x:500:
nis_user:x:501:
student1:x:502:
student2:x:555:
student3:x:556:
new_user:x:557:
students:x:558:oracle,student1,student2
```

    b. Run the same command but redirect the output to `group`.

```
# grep ":5[0-9][0-9]" /etc/group > group
```

    c. Run the `migrate_group.pl` command to migrate group information in the `group` file into an LDAP format. Redirect the output to `group.ldif`.
       - Use the absolute path name, because the `/usr/share/migrationtools` directory is not in your path.

```
# /usr/share/migrationtools/migrate_group.pl group > group.ldif
```

    d. Use the `ldapadd` command to import the group information to the LDAP directory. The LDAP password is `oracle`.

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f group.ldif
Enter LDAP Password: oracle
adding new entry "cn=oracle,ou=Group,dc=example,dc=com"

adding new entry "cn=nis_user,ou=Group,dc=example,dc=com"

adding new entry "cn=student1,ou=Group,dc=example,dc=com"

adding new entry "cn=student2,ou=Group,dc=example,dc=com"

adding new entry "cn=student3,ou=Group,dc=example,dc=com"

adding new entry "cn=new_user,ou=Group,dc=example,dc=com"

adding new entry "cn=students,ou=Group,dc=example,dc=com"
```

e. Use the `ldapsearch` command to display the new `students` group entry in the LDAP server.

```
# ldapsearch –x “cn=students” -b “dc=example,dc=com”
...
# students, Group, example.com
dn: cn=students,ou=Group,dc=example,dc=com
objectClass: posixGroup
objectClass: top
cn: students
userPassword:: e2NyeXB0...
gidNumber: 558
memberUid: oracle
memberUid: student1
memberUid: student2

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

13. Modify `iptables` to allow access on port 389.

a. Use the `vi` editor to edit the `/etc/sysconfig/iptables` file. Use the `:set nu` command to turn on line numbers.

```
# vi /etc/sysconfig/iptables
...
:set nu
```

b. Add the following two new entries after line 10:

```
-A INPUT -m state --state NEW -m udp -p udp --dport 389 -j
ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 389 -j
ACCEPT
```

c. Use the `service` command to start the `iptables` service.

```
# service iptables start
iptables: Applying firewall rules:              [  OK  ]
```

Practice 3-7: Implementing OpenLDAP Authentication

Overview

In this practice, you use the Authentication Configuration Tool to implement OpenLDAP authentication.

Assumptions

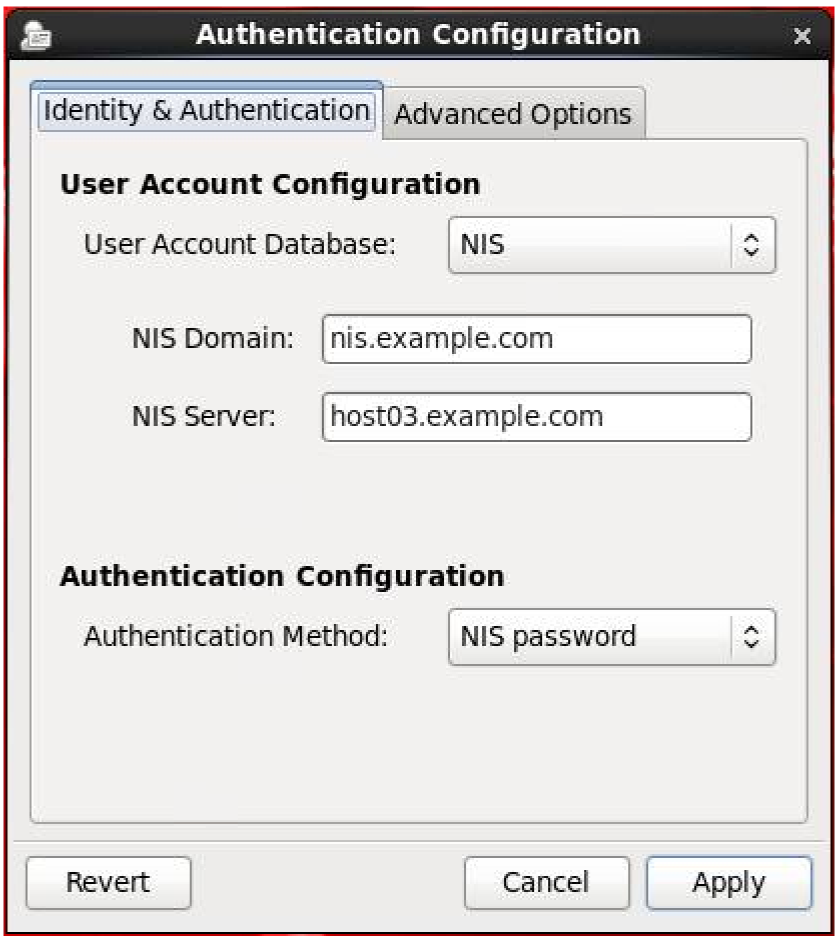
- Ensure that you are using `vncviewer` to connect to **host03** and not using `ssh`.
- You are the `root` user on **host03** VM.

Tasks

1. From **host03**, open the Authentication Configuration Tool by running the `system-config-authentication` command.

```
# system-config-authentication
```

- The GUI appears as follows with NIS configured from Practice 3-3.



2. Make the following changes:
   a. Select `LDAP` from the User Account Database drop-down list.
   b. Enter `dc=example,dc=com` as the LDAP Search Base DN.
   c. Enter `ldap://host03.example.com` as the LDAP Server.
   d. Click “Use TLS to encrypt connections.”

e. Select `LDAP password` as the Authentication Method.

- Ensure that your screen is configured as shown:

3. Click **Apply** to save your changes.
   - After clicking Apply, the Authentication Configuration Tool closes and the following message appears in your terminal window:

```
Starting sssd:                                  [  OK  ]
```

4. Run the `authconfig --test` command to view the authentication settings.

```
# authconfig --test
caching is disabled
...
nss_ldap is enabled
 LDAP+TLS is enabled
 LDAP server = “ldap://host03.example.com”
 LDAP base DN = “dc=example,dc=com”
nss_nis is disabled
...
pam_ldap is enabled
 LDAP+TLS is enabled
 LDAP server = “ldap://host03.example.com”
 LDAP base DN = “dc=example,dc=com”
...
```

Practice 3-8: Authenticating from an OpenLDAP Client

Overview

In this practice, you:

- Install the OpenLDAP client packages
- Configure the OpenLDAP client
- Log in as OpenLDAP user to test LDAP authentication
- Disable OpenLDAP authentication

You begin this practice by opening a second terminal window on **dom0** and logging in to **host01** as the `root` user. You are already logged in as the `root` user to **host03** from Practice 3-1.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- You are currently logged in to **host03**.

Tasks

1. Log in to the **host01** VM guest from **dom0**.
   a. If necessary, open a second terminal window on **dom0**.
   b. From the second terminal window on **dom0**, use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
$ su -
Password: oracle
#
```

   c. As the root user on **dom0**, use the `ssh` command to log in to **host01**.
      - The `root` password is `oracle` (all lowercase).

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[host01]#
```

2. Attempt to log in as user `student1`.
   a. From **host01**, use the `su - student1` command to attempt to log in as user `student1`.

```
# su - student1
su: user student1 does not exist
```

      - Notice that user `student1` is not a valid user on **host01**.

   b. Use the `grep` command to search for user `student1` in local `/etc/passwd` file.

```
# grep student1 /etc/passwd
```

      - The command produces no output indicating `student1` is not a local user on **host01**.

3. Install the authentication packages on **host01**.
   a. Use the `yum` command to install the `openldap-clients` package.
      - Answer "`y`" when prompted "**Is this ok [y/N]:**".
      - You are asked about the GPG key only the first time you use the `yum install` command.

```
# yum install openldap-clients
...
Transaction Summary
=============================================================
Install       1 Package(s)
Total download size: 165 k
Installed size: 609 k
Is this ok [y/N]: y
...
Retrieving key from http://192.0.2.1/repo/OracleLinux/OL6/5/...
...
Importing GPG key ...
...
Is this ok [y/N]: y
...
Complete!
```

   b. Use the `yum` command to install the `pam_ldap` package.

```
# yum install pam_ldap
...
Transaction Summary
=============================================================
Install       1 Package(s)
Total download size: 87 k
Installed size: 154 k
Is this ok [y/N]: y
...
Complete!
```

   c. Use the `yum` command to install the `nss-pam-ldapd` package.

```
# yum install nss-pam-ldapd
...
Transaction Summary
=============================================================
Install       2 Package(s)
Total download size: 370 k
Installed size: 630 k
Is this ok [y/N]: y
...
Complete!
```

4. Configure the `/etc/openldap/ldap.conf` file on **host01**.
   a. Use the `cd` command to change to the `/etc/openldap` directory. Use the `ls -l` command to display the contents of the directory.

```
# cd /etc/openldap
# ls -l
drwxr-xr-x. 2 root root ... certs
-rw-r--r--. 1 root root ... ldap.conf
```

   b. Use the `vi` editor to make the following changes to the `ldap.conf` file.
      - Uncomment the lines by removing the # character.

```
# vi ldap.conf
BASE         dc=example,dc=com
URI          ldap://192.0.2.103/
```

5. Configure the `/etc/nslcd.conf` file on **host01**.
   a. Use the `cd` command to change to the `/etc` directory.

```
# cd /etc
```

   b. Use the `vi` editor to make the following changes to the `nslcd.conf` file.
      - These entries are the last two lines in the file.
      - The "`base`" line does not need changed.

```
# vi nslcd.conf
uri          ldap://192.0.2.103/
base         dc=example,dc=com
```

6. Configure the `/etc/pam_ldap.conf` file on **host01**.
   a. Use the `vi` editor to make the following changes to the `pam_ldap.conf` file. Use the `:set nu` command to turn on line numbers.

```
# vi pam_ldap.conf
...
:set nu
```

   b. At around line 17, comment out `host 127.0.0.1`, by inserting a # character at the beginning of the line.

```
#host 127.0.0.1
```

   c. At around line 25, uncomment the "`uri`" line and set the value shown as follows:

```
uri          ldap://192.0.2.103/
```

      - Notice that, at around line 20, `base` is set to `dc=example,dc=com`; therefore, no further changes are needed.

7. Configure the `/etc/pam.d/system-auth` file on **host01**.
   a. Use the `cd` command to change to the `/etc/pam.d` directory.

```
# cd /etc/pam.d
```

   b. Use the `cp` command to make a backup copy of the `system-auth` file.

```
# cp system-auth system-auth.backup
```

      - This backup file is used later in this practice to restore the original configuration.

c. Use the `vi` editor to make the following changes to the `system-auth` file. In the first section (lines beginning with `auth`) of the file, add the following **bold** line in the location as shown:

Note: A sample `system-auth` file exists on **dom0** in the `/OVS/seed_pool/sfws` directory.

- You can edit the `system-auth` file as follows using the `vi` command, or you can use the `sftp` command and copy `/OVS/seed_pool/sfws/system-auth` from **dom0** to `/etc/pam.d/system-auth` on **host01**. See your instructor if you need help in using the `sftp` command.
- You must make several changes to this file. Do not exit the `vi` editor until step 7g.

```
# vi system-auth
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is
run.
auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_ldap.so use_first_pass
auth        required      pam_deny.so
```

d. In the second section of the file (lines beginning with `account`), add the following **bold** line in the location as shown:

- Ensure that the new entry is on a single line.

```
account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     [default=bad success=ok user_unknown=ignore]
pam_ldap.so
account     required      pam_permit.so
```

e. In the third section of the file (lines beginning with `password`), add the following **bold** line in the location as shown:

```
password    requisite     pam_cracklib.so try_first_pass retry=3
type=
password    sufficient    pam_unix.so sha512 shadow nullok
try_first_pass use_authtok
password    sufficient    pam_ldap.so use_authtok
password    required      pam_deny.so
```

f. In the fourth section of the file (lines beginning with `session`), add the following two **bold** lines in the location as shown:

- Ensure that the two new entries are each on a separate single line.

```
session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
```

```
session     [success=1 default=ignore] pam_succeed_if.so service
in crond quiet use_uid
session     required      pam_unix.so
session     optional      pam_ldap.so
session     optional      pam_mkhomedir.so skel=/etc/skel
umask=077
```

g. Save the file and exit `vi`.

8. Configure the `/etc/nsswitch.conf` file on **host01**.

   a. Use the `cd` command to change to the `/etc` directory.

```
# cd /etc
```

   b. Use the `vi` editor to remove `nis` and add `ldap` to the `passwd`, `shadow`, and `group` directives as shown:

```
# vi nsswitch.conf
passwd:   nis files                  (old entry)
shadow:   nis files                  (old entry)
group:    nis files                  (old entry)
passwd:   files ldap                 (new entry)
shadow:   files ldap                 (new entry)
group:    files ldap                 (new entry)
```

   c. Save the file and exit `vi`.

9. Configure the `/etc/sysconfig/authconfig` file on **host01**.

   a. Use the `cd` command to change to the `/etc/sysconfig` directory.

```
# cd /etc/sysconfig
```

   b. Use the `vi` editor to edit the `authconfig` file and change `USELDAP=no` to `USELDAP=yes` as shown:

```
# vi authconfig
USELDAP=no                           (old entry)
USELDAP=yes                          (new entry)
```

10. Use the `service` command to start the `nslcd` service on **host01**.

```
# service nslcd start
Starting nslcd:                                  [  OK  ]
```

11. Log in as the OpenLDAP user from **host01**.

   a. Use the `grep` command to search for user `student1` in the local `/etc/passwd` file.

```
# grep student1 /etc/passwd
```

   - The command produces no output, indicating that `student1` is not a local user.

   b. Use the `ls` command to list the contents of the `/home` directory.

```
# ls /home
```

   - Notice that there is no home directory for the `student1` user.

   c. Use the `ldapsearch` command to search for `student1` in the OpenLDAP directory.

   - The common name (`cn`) for `student1` is "`Oracle Student1`".

```
# ldapsearch –x "cn=Oracle Student1" -b "dc=example,dc=com"
...
# student1, People, example.com
dn: uid=student1,ou=People,dc=example,dc=com
uid: student1
cn: Oracle Student1
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0...
shadowLastChange: ...
shadowMin: 0
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 501
gidNumber: 501
homeDirectory: /home/student1
gecos: Oracle Student1

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

d. Use the `su – student1` command to log in as OpenLDAP user `student1`. Use the `whoami` command to verify you are logged in as `student1`.

```
# su – student1
Creating directory '/home/student1'.
[student1@host01 ~]$ whoami
student1
```

- Notice that you can successfully log in as `student1` even though the user account does not exist locally.
- Notice that a home directory was created for `student1`.

e. Use the `pwd` command to verify that the `/home/student1` directory was created on the localhost.

```
[student1@host01 ~]$ pwd
/home/student1
```

f. Use the `ls -la` command to view the contents of the directory.

```
[student1@host01 ~]$ ls -la
...
-rw-------. 1   student1  student1  ...  .bash_logout
-rw-------. 1   student1  student1  ...  .bash_profile
-rw-------. 1   student1  student1  ...  .bashrc
```

- Notice that the contents of `/etc/skel` were copied into the user's home directory.

g. Use the `exit` command to log off as `student1`.

```
[student1@host01 ~]$ exit
logout
```

12. Disable the OpenLDAP client authentication on **host01**.

a. From **host01**, use the `service` command to stop the `nslcd` service.

```
# service nslcd stop
Stopping nslcd:                                     [  OK  ]
```

b. Use the `vi` editor to edit the `authconfig` file and change `USELDAP=yes` to `USELDAP=no` as shown:

```
# cd /etc/sysconfig
# vi authconfig
USELDAP=yes                               (old entry)
USELDAP=no                                (new entry)
```

c. Use the `vi` editor to remove `ldap` from the `passwd`, `shadow`, and `group` directives as shown:

```
# cd /etc
# vi nsswitch.conf
passwd:   files ldap
shadow:   files ldap
group:    files ldap
```

d. Use the `cp` command to restore the `system-auth` file.

```
# cd /etc/pam.d
# cp system-auth.backup system-auth
cp: overwrite 'system-auth? y
```

e. Use the `su - student` command to attempt to log in as user `student1`.

```
# su – student1
su: user student1 does not exist
```

- This confirms OpenLDAP client authentication is disabled.

f. Use the `exit` command to log off of **host01**.

```
# exit
logout
Connection to host01 closed.
```

Perform the next step from **host03**.

13. Disable the OpenLDAP server authentication.

a. From **host03**, open the Authentication Configuration Tool by running the `system-config-authentication` command.

```
# system-config-authentication
```

- The GUI appears as follows:

b. Select `Local accounts only` from the User Account Database drop-down list.

- Ensure that your screen is configured as shown:

c. Click **Apply** to save your changes.

- After clicking Apply, the Authentication Configuration Tool closes and the following message appears in your terminal window:

```
Stopping sssd:                                          [  OK  ]
```

d. Use the `service` command to stop the `slapd` service.

```
# service slapd stop
Stopping slapd:                                         [  OK  ]
```

Do not log off **host03**. The next practice (Practice 4-1) assumes that you are still logged on with `vncviewer`.

Practices for Lesson 4: Web and Email Services

Chapter 4

Practices for Lesson 4: Web and Email Services

Practices Overview

In these practices, you configure the Apache Web Server.

Practice 4-1: Configuring Apache Web Server

Overview

In this practice, you:

- Verify that the `httpd` package is installed, start the service, and ensure that the service starts at boot time
- Create a test page to verify that Apache is working correctly
- Configure two virtual hosts, each serving different web content

Assumptions

- You perform this practice exclusively on **host03** VM.
- You are still connected to **host03** using `vncviewer` from the Practice 3.
- You are the `root` user on **host03** VM.

Tasks

1. If you are not connected to **host03** using `vncviewer`, refer to Practice 3-1 for instructions.
2. As the root user on **host03**, use the `rpm` command to verify that the `httpd` package is installed.

```
# rpm -qa | grep httpd
httpd-2.2.15-29.0.1.el6.x86_64
httpd-tools-2.2.15-29.0.1.el6.x86_64
```

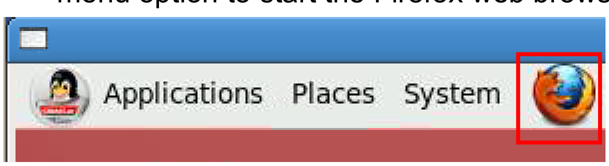
   - In this example, the `httpd` package is installed.
   - The `httpd-tools` package is also installed. This package contains tools that can be used with the Apache HTTP server.
3. Use the `service` command to start the `httpd` service.

```
# service httpd start
Starting httpd:                                          [  OK  ]
```

4. Use the `chkconfig` command to enable the `httpd` service to start at boot time.

```
# chkconfig httpd on
```

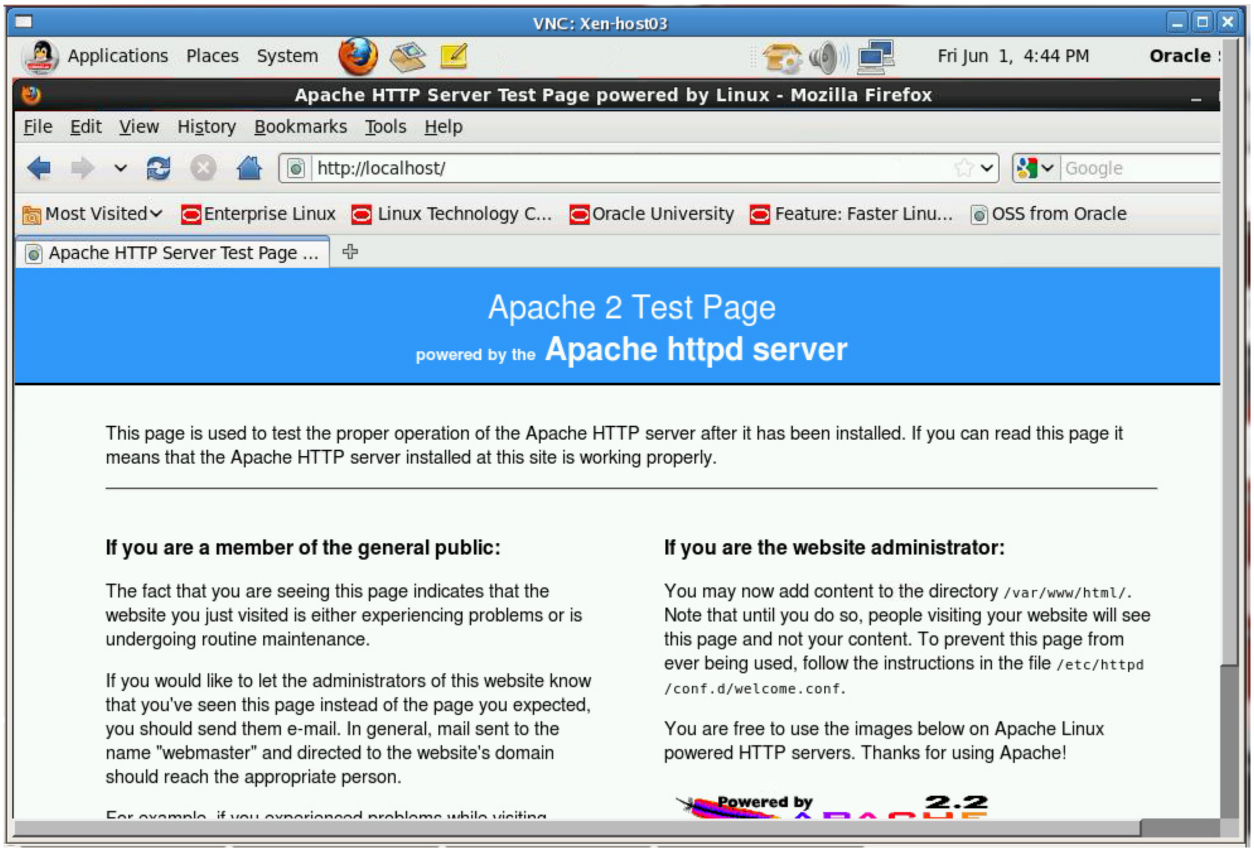
5. Confirm that Apache is working, by pointing a browser on **host03** to `http://localhost`.
   a. On the GNOME menu bar, click the Firefox We Browser icon to the right of the System menu option to start the Firefox web browser.



   - The browser appears.

b. Enter `http://localhost` in the browser and press Enter.

- The Apache Test Page appears.



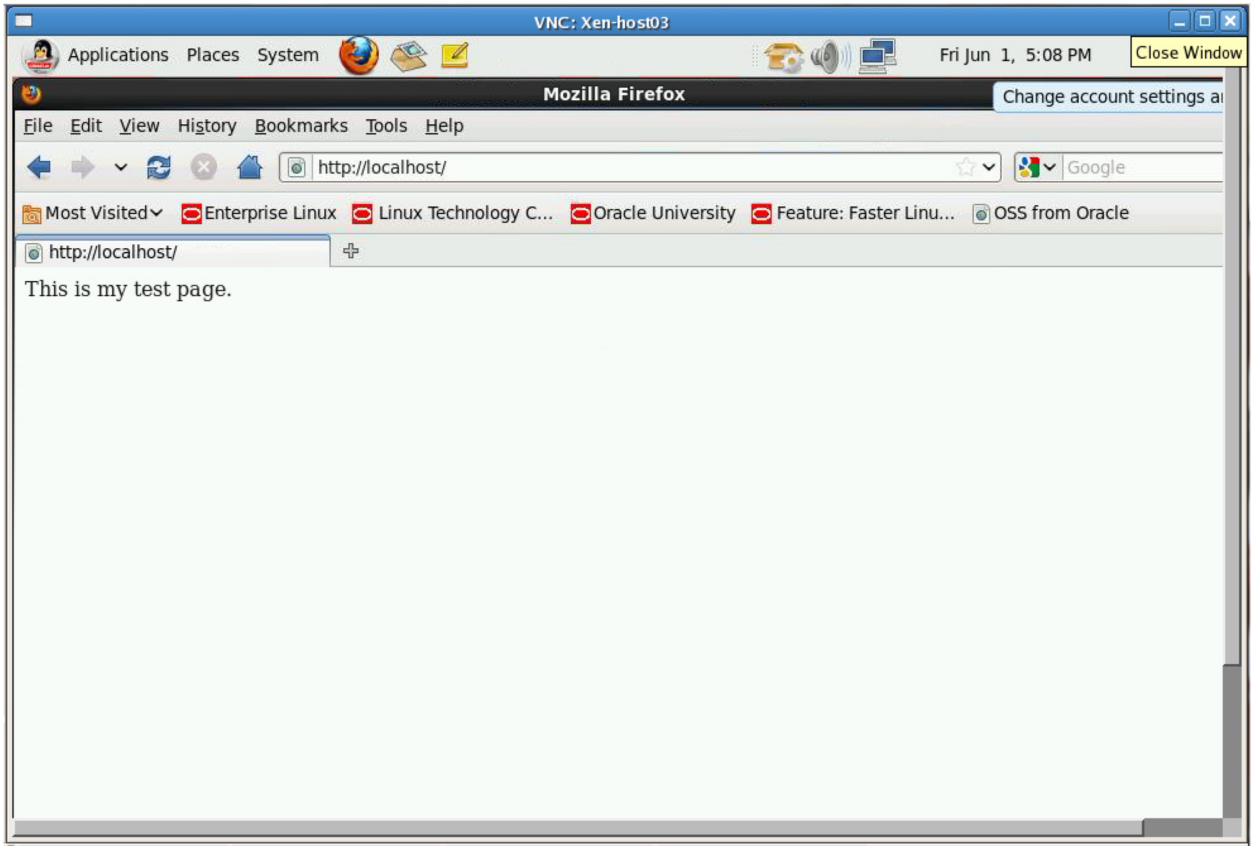
- The Test Page confirms that Apache is working correctly.

c. Close the Firefox web browser by selecting **File > Quit** from the menu bar.

6. Create a test webpage by using the `vi` editor to create the `/var/www/html/index.html` file and add the following entry.

```
# vi /var/www/html/index.html
<html><body><p>This is my test page.</p></body></html>
```

7. Restart the Firefox browser and point to `http://localhost`.
   - The test webpage appears.



8. Close the Firefox web browser by selecting **File > Quit** from the menu bar.
9. Create a virtual host on the Apache web server and name it `www.example1.com`.
   a. Use the `vi` editor to edit the `/etc/httpd/conf/httpd.conf` file to add the following entries to the end of the file:
      - There are lines similar to the following in the `httpd.conf` file.
      - You can un-comment these lines and change them as follows, or create new lines.

```
# vi /etc/httpd/conf/httpd.conf
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName www.example1.com
    DocumentRoot /var/www/example1
    ErrorLog /var/log/httpd/example1.error_log
    <Directory /var/www/example1>
        Order deny,allow
        Deny from all
        Allow from 192.0.2
    </Directory>
</VirtualHost>
```

b. Use the `vi` editor to edit the `/etc/hosts` file and append `www.example1.com` to the `192.0.2.103` entry as follows:

```
# vi /etc/hosts
192.0.2.103   host03.example.com   host03   www.example1.com
```

c. Use the `mkdir` command to make the `/var/www/example1` directory.

```
# mkdir /var/www/example1
```

d. Use the `cp` command to copy the `/var/www/html/index.html` file to the `/var/www/example1` directory.

```
# cp /var/www/html/index.html /var/www/example1/
```

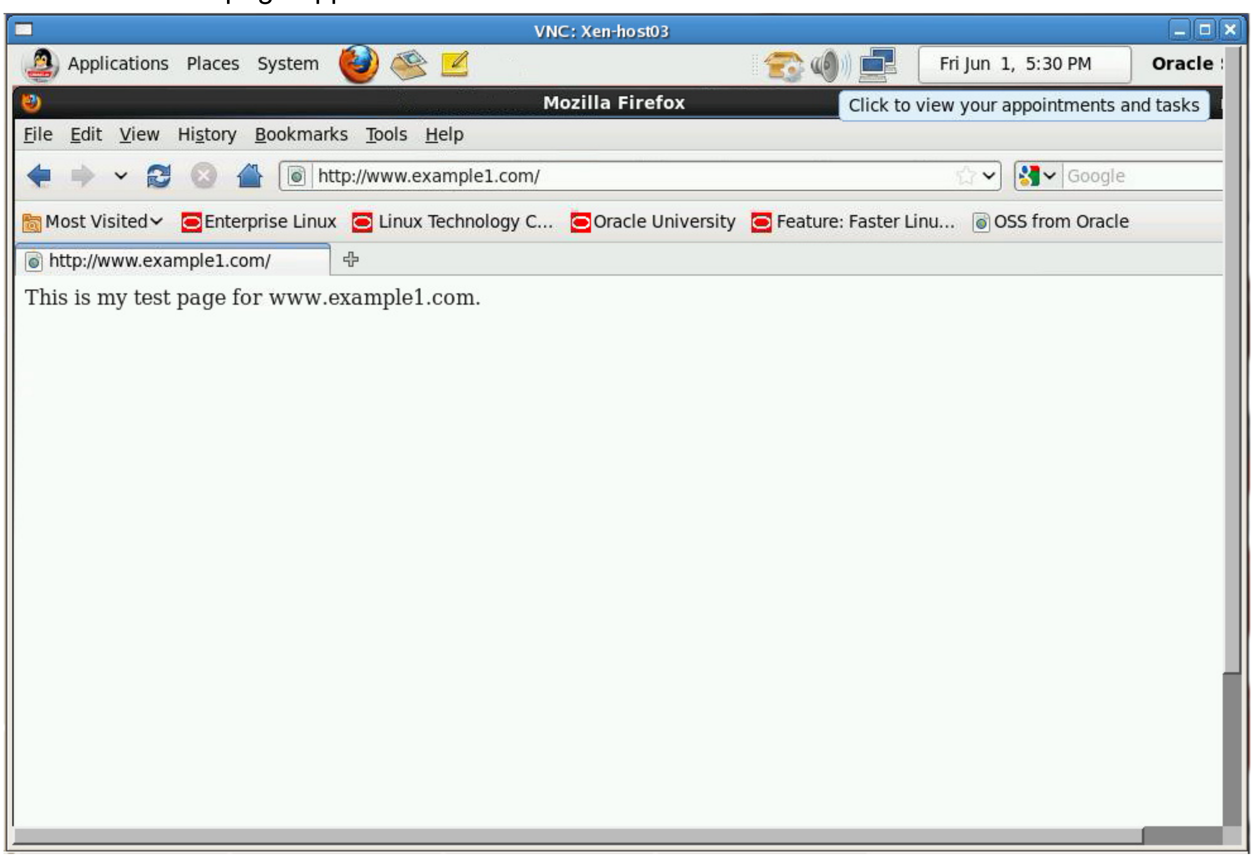
e. Use the `vi` editor to edit the `/var/www/example1/index.html` file as follows:

```
# vi /var/www/example1/index.html
<html><body><p>This is my test page for
www.example1.com.</p></body></html>
```

f. Use the `service` command to restart the `httpd` service.

```
# service httpd restart
Stopping httpd                                  [  OK  ]
Starting httpd:                                 [  OK  ]
```

10. Restart the Firefox browser and point to `http://www.example1.com`.
   - The test webpage appears.



11. Close the Firefox web browser by selecting **File > Quit** from the menu bar.

12. Create a second virtual host on the Apache web server named `www.example2.com`.

   a. Use the `vi` editor to edit the `/etc/httpd/conf/httpd.conf` file to add the following entries to the end of the file:

```
# vi /etc/httpd/conf/httpd.conf
<VirtualHost *:80>
    ServerName www.example2.com
    DocumentRoot /var/www/example2
    ErrorLog /var/log/httpd/example2.error_log
    <Directory /var/www/example2>
        Order deny,allow
        Deny from all
        Allow from 192.0.2
    </Directory>
</VirtualHost>
```

   b. Use the `vi` editor to edit the `/etc/hosts` file to append `www.example2.com` to the `192.0.2.103` entry as follows:

```
# vi /etc/hosts
192.0.2.103  host03... www.example1.com  www.example2.com
```

   c. Use the `mkdir` command and make the `/var/www/example2` directory.

```
# mkdir /var/www/example2
```

   d. Use the `cp` command to copy the `/var/www/example1/index.html` file to the `/var/www/example2` directory.

```
# cp /var/www/example1/index.html /var/www/example2
```

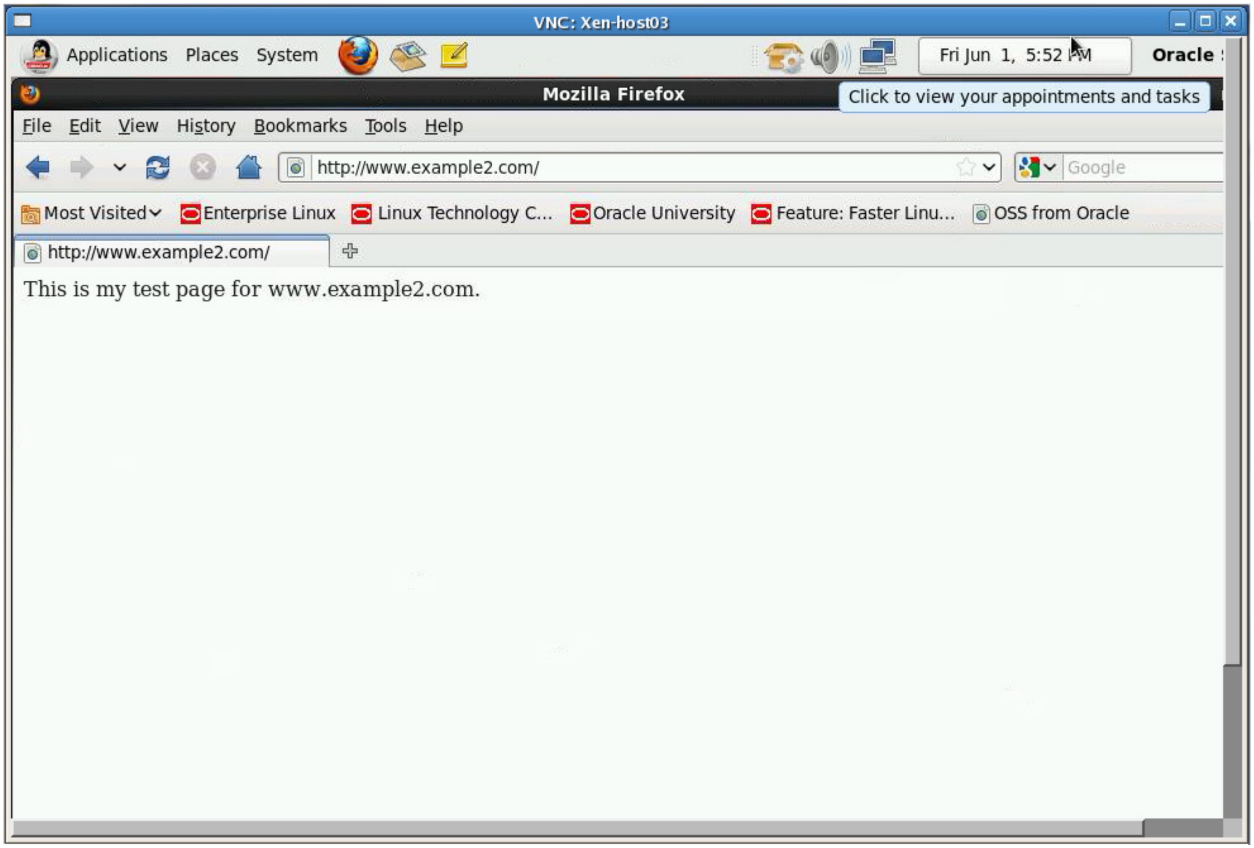
   e. Use the `vi` editor to edit the `/var/www/example2/index.html` file as follows:

```
# vi /var/www/example2/index.html
<html><body><p>This is my test page for
www.example2.com.</p></body></html>
```

   f. Use the `service` command to restart the `httpd` service.

```
# service httpd restart
Stopping httpd                              [  OK  ]
Starting httpd:                             [  OK  ]
```

13. Restart the Firefox browser and point to `http://www.example2.com`.
    - The example2 test webpage appears as shown.



14. Close the Firefox web browser by selecting **File > Quit** from the menu bar.

Do not log off **host03**. The next practice (Practice 5-1) assumes that you are still logged on with `vncviewer`.

Practices for Lesson 5: Installing Oracle Linux by Using Kickstart

Chapter 5

Practices for Lesson 5: Installing Oracle Linux Using Kickstart

Practices Overview

In these practices, you:

- Explore the Kickstart Configurator GUI
- Create a new **host07** virtual machine and perform a Kickstart installation on **host07**
- Use rescue mode to repair a boot problem on **host07**

Practice 5-1: Using the Kickstart Configurator

Overview

In this practice, you become familiar with the Kickstart Configurator. You perform the following:

- Install the Kickstart Configuration packages on **host03** VM.
- Start the Kickstart Configurator GUI.
- Load the `/root/anaconda-ks.cfg` file into the Configurator.
- View the different configuration page options in the Configurator.
- Preview the `/root/anaconda-ks.cfg` file using the Configurator.
- Close the Kickstart Configurator.

Assumptions

- You are still connected to **host03** using `vncviewer` from Practice 4-1.
- You are the `root` user on **host03** VM.

Tasks

1. If you are not connected to **host03** using `vncviewer`, refer to Practice 3-1 for instructors.
2. Install the Kickstart Configurator.

   As the `root` user on **host03**, run the `yum install system-config-kickstart` command to install the Configurator.

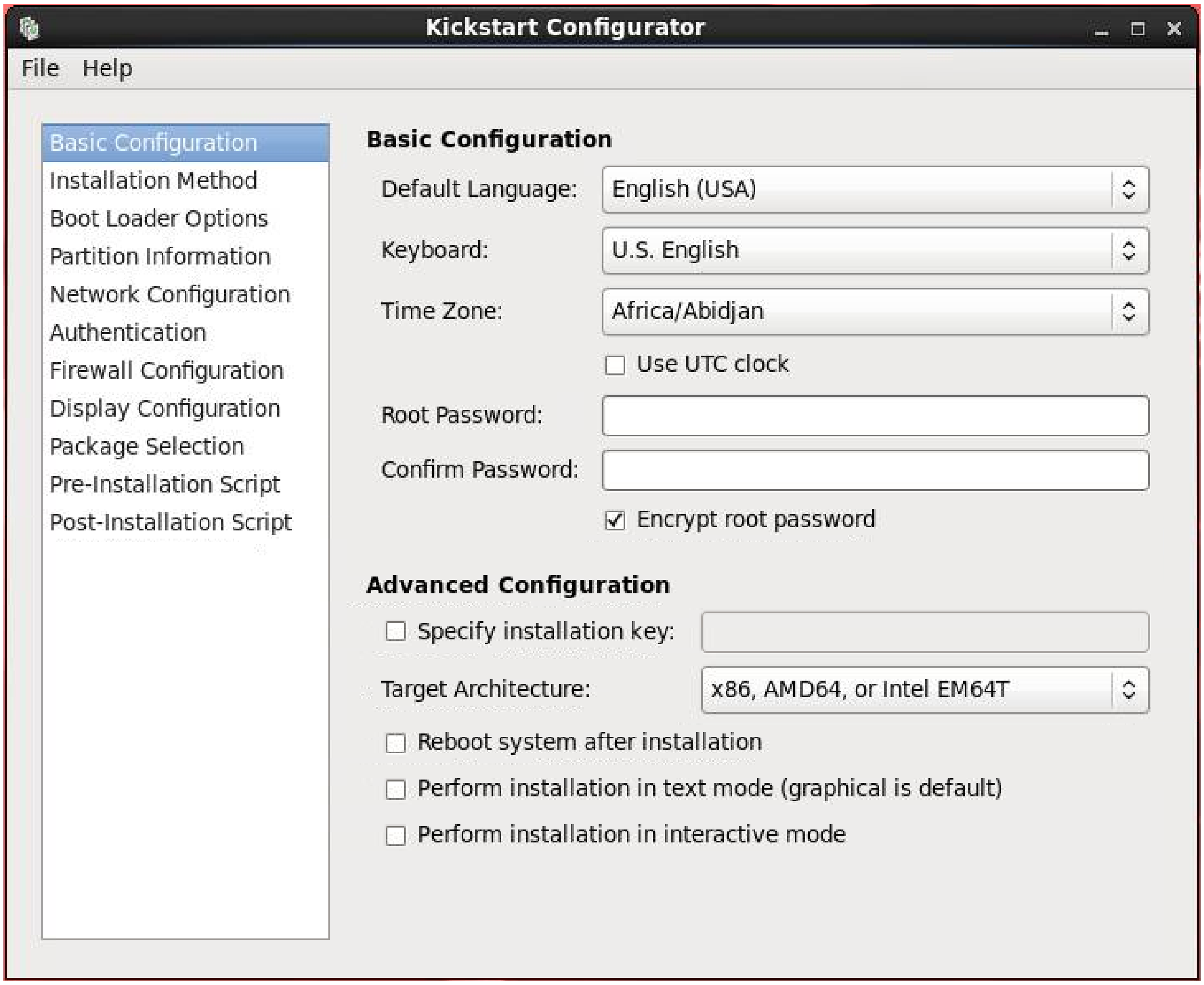
```
# yum install system-config-kickstart
...
Transaction Summary
==============================================================
Install      25 Package(s)

Total download size: 7.9 M
Installed size: 28 M
Is this ok [y/N]: y
...
Complete!
```

3. Start the Kickstart Configurator.

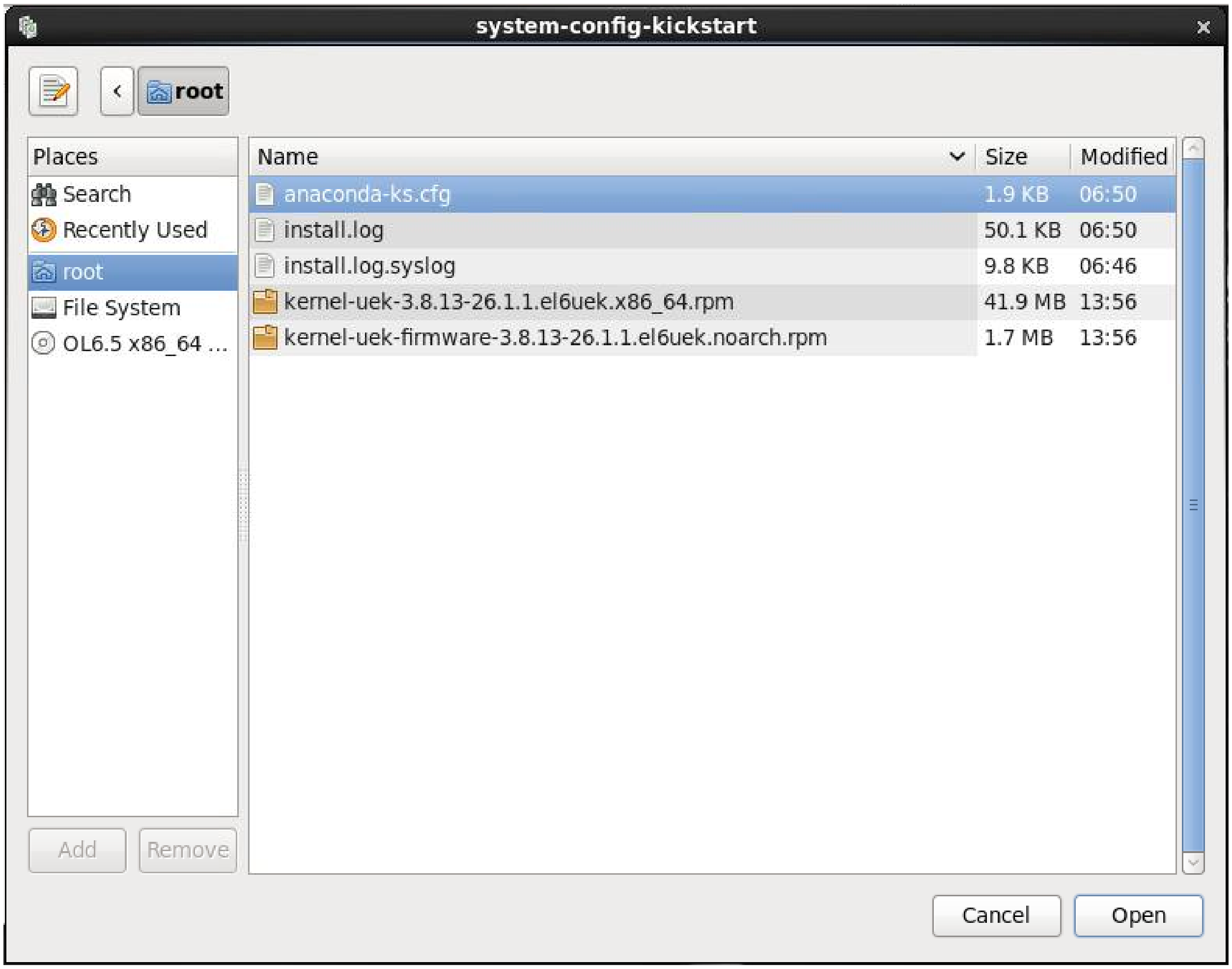
   Run the `system-config-kickstart` command to start the Kickstart Configurator.

```
# system-config-kickstart
```

   - The Kickstart Configuration GUI appears.

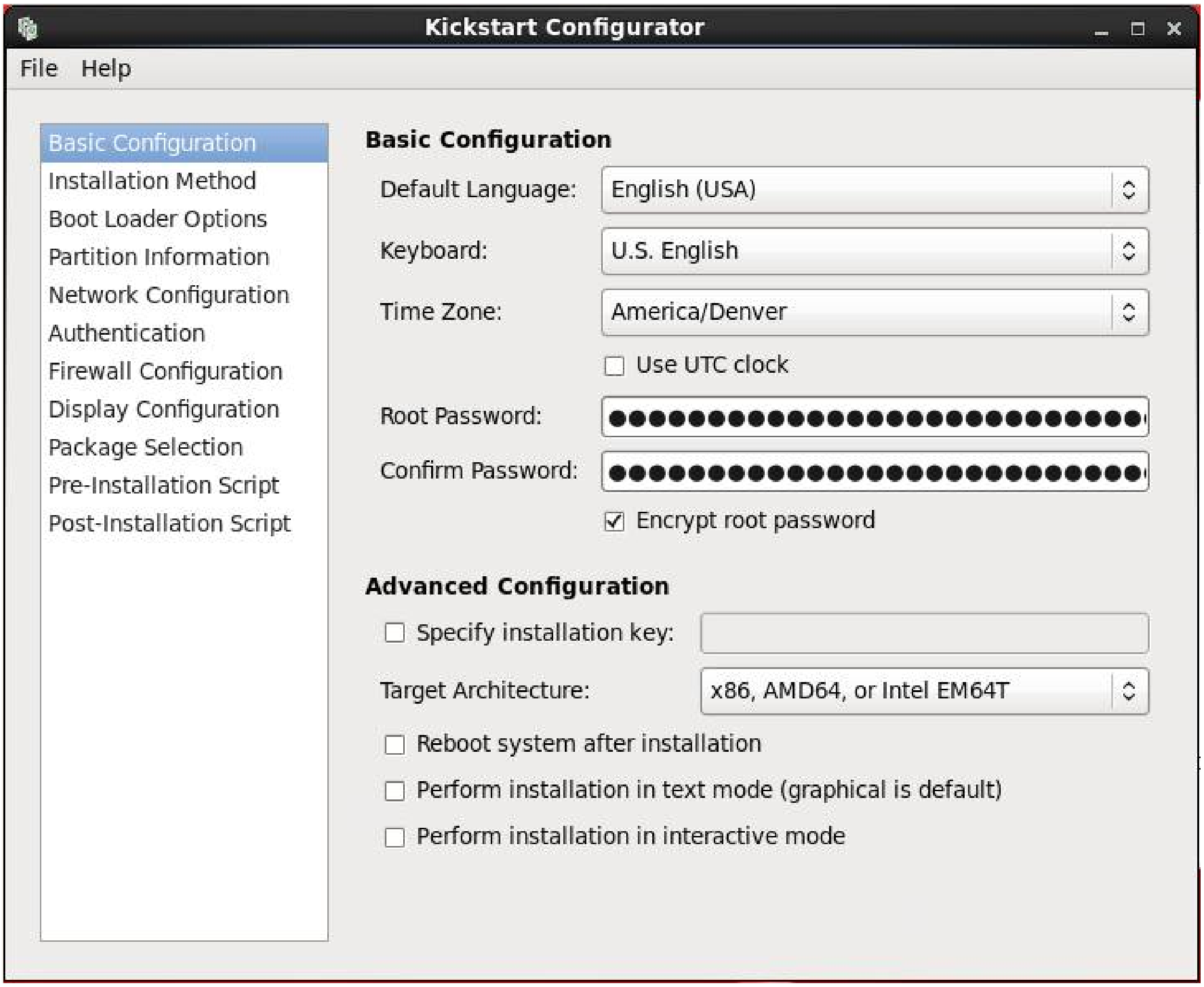


4. Load the `/root/anaconda-ks.cfg` Kickstart file into the Configurator.
   a. Select **File > Open File** to display the following screen.



b. Select `/root/anaconda-ks.cfg` and click **Open**.
   - The `/root/anaconda-ks.cfg` file is loaded into the Configurator.

5. View the different configuration pages in the Kickstart Configurator.
   - The Basic Configuration page is shown below.



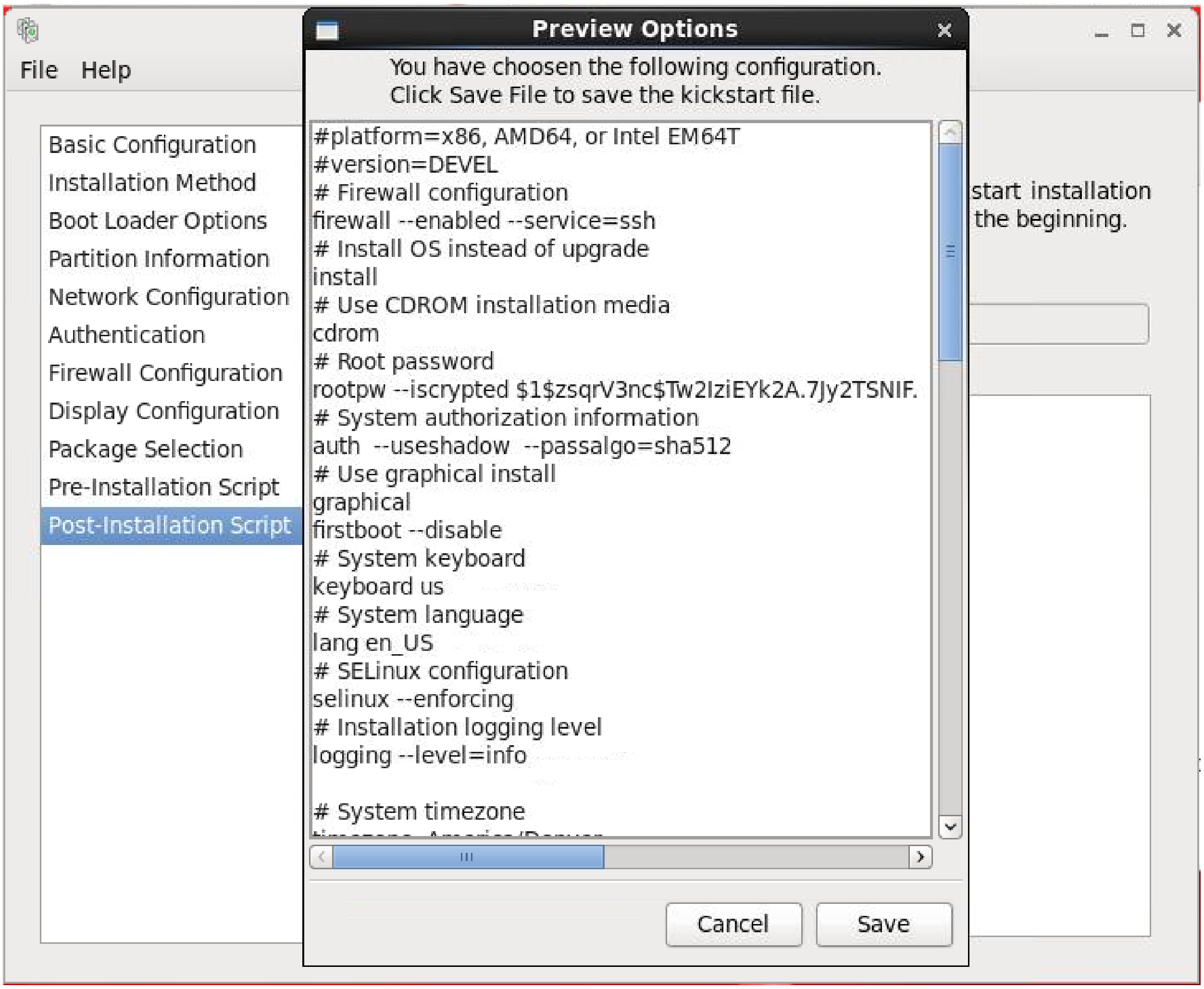
a. After viewing the Basic Configuration screen, select the next option from the list on the left side of the GUI, Installation Method.

b. After viewing the Installation Method screen, continue selecting the other options from the list—Boot Loader Options, Partition Information, and so on—and view each page.

- This is an exploratory task for you to familiarize yourself with the use of the Configurator.
- The Package Selection option displays the message, “Package selection is disabled due to problems downloading package information.” Currently, `system-config-kickstart` does not support package selection and deselection. You need to update the package section in your Kickstart file manually.

6. Preview the Kickstart file in the Configurator.

   a. Select **File > Preview** to display the following screen.



b. Use the scroll bars to view the entire Kickstart file.

c. Click **Cancel** to close the file.

7. Close the Kickstart Configurator.

   a. Select **File > Quit** to close the Kickstart Configurator.

8. Log off of **host03**.

   a. Select **System > Log Out oracle…** from the GNOME menu bar.

   b. Click **Log Out** in the confirmation dialog box.

   c. Click the `X` in the top-right of the GNOME login window to close the window.

Practice 5-2: Performing a Kickstart Installation

Overview

In this practice, you do the following:

- Configure **dom0** as an HTTP server.
- Make the installation tree available from the HTTP server.
- Create the Kickstart file and make it available from the HTTP server.
- Shut down the **host01** VM and create a new **host07** VM.
- Initiate the Kickstart installation on **host07**.
- Log in to **host07** and verify the installation.
- Shut down **host07** and restart **host01**.

Assumptions

You are logged on as the `root` user on **dom0**.

Tasks

1. Ensure that **dom0** is configured as an HTTP server.
   a. If necessary, open a terminal window on **dom0** and become the `root` user.
      - From a terminal window on **dom0**, use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
$ su –
Password: oracle
#
```

   b. As the `root` user on **dom0**, use the `rpm` command to ensure that the http package is installed.

```
# rpm –qa | grep http
httpd-2.2.3-22.0.1.el5
```

      - The package is installed.

   c. Use the `service` command to query the status of the `httpd` service.

```
# service httpd status
httpd (pid ...) is running...
```

      - In this example, the `httpd` service is running. If the service is not running, use the `service` command to start the `httpd` service:

```
# service httpd start
...
```

2. Make the installation tree available.
   - In this task, you make the installation tree available from the HTTP server running on **dom0**.

   a. From **dom0**, use the `cd` command to change to the `/OVS/seed_pool` directory. Use the `ls` command to list the contents of the directory.

- The Oracle Linux 6 update 5 DVD image is the `OracleLinux-R6-U5-Server-x86_64-dvd.iso` file in the `/OVS/seed_pool` directory.

```
# cd /OVS/seed_pool
# ls
...
OracleLinux-R6-U5-Server-x86_64-dvd.iso
...
```

b. Use the `mkdir` command to make a temporary mount, `/mnt/iso`.

- Using a temporary mount point other than `/mnt` is a requirement imposed by Oracle University (OU). On OU systems, there is a FAT file system mounted in `/mnt/cdrive`. This file system holds binaries that monitor the machine status and take care of initiating the build for the next class after the current class is finished. If you are mounting an ISO on `/mnt`, it mounts on top of `/mnt/cdrive`. This causes the binaries to fail to report to the OU Dashboard. Outside of the OU environment, you can use `/mnt` for this procedure.

```
# mkdir /mnt/iso
```

c. Use the `mount` command to mount the OL6.5 DVD image on `/mnt/iso`.

```
# mount -t iso9660 -o loop OracleLinux-R6-U5-Server-x86_64-
dvd.iso /mnt/iso
```

d. Use the `mkdir` command to create the `/var/www/html/OL6.5` directory.

```
# mkdir /var/www/html/OL6.5
```

e. Use the `cp` command to copy all files and directories from `/mnt/iso` to `/var/www/html/OL6.5`.

- This command takes a few minutes to complete.

```
# cp –r /mnt/iso/* /var/www/html/OL6.5/
```

- The installation tree is now available from the HTTP server running on **dom0**.

f. Use the `umount` command to unmount `/mnt/iso`. Use the `rmdir` command to remove the `/mnt/iso` directory.

```
# umount /mnt/iso
# rmdir /mnt/iso
```

3. Create the Kickstart file.

- The installation of Oracle Linux creates a Kickstart file, `/root/anaconda-ks.cfg`, based on the options that you selected during installation.
- Use this file as a template for creating the `ks.cfg` file.

a. From **dom0**, use the `scp` command to copy `/root/anaconda-ks.cfg` from **host03** to `/var/www/html/ks.cfg` on **dom0**.

- The password is `oracle`.

```
# cd /var/www/html
# scp host03:~/anaconda-ks.cfg ks.cfg
root@host03's password: oracle
anaconda-ks.cfg                    100%  ...
```

- The Kickstart file is now available from the HTTP server running on **dom0**.

- You use the `vi` editor to change this Kickstart file as instructed in step 3c.

Note: A pre-configured `ks.cfg` file exists on **dom0** in the `/OVS/seed_pool/host07` directory.

- If you do not wish to edit the `ks.cfg` file as instructed in step 3c, you can use the `cp` command to copy `/OVS/seed_pool/host07/ks.cfg` to `/var/www/html/ks.cfg`. If you use this Kickstart file, you do not need to edit the file in step 3c.

b. Use the `chown -R` command to change the owner and group to `apache` on `/var/www/html`.

- This is a requirement of HTTP; otherwise, you will get "permission denied" errors.

```
# chown -R apache.apache /var/www/html
```

c. Use the `vi` editor to edit the `ks.cfg` file. Change the file to make it like the following.

- Changes and additions are in bold.
- Delete any lines in the file that are not shown below.
- Ensure that the lines are in the same order as shown below.

```
# vi ks.cfg
# Kickstart file automatically generated by anaconda.

#version=DEVEL
install
url --url http://192.0.2.1/OL6.5/
lang en_US.UTF-8
keyboard us
network --onboot yes --device eth0 --bootproto static --ip
192.0.2.107 --netmask 255.255.255.0 --gateway 192.0.2.1 --noipv6
--hostname host07.example.com
rootpw --iscrypted ...
firewall --service=ssh
authconfig --enableshadow --passalgo=sha512
selinux --enforcing
timezone --utc America/Denver
bootloader --location=mbr --driveorder=xvda --append="
crashkernel=auto rhgb quiet"
zerombr yes
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all --drives=xvda

part /boot --fstype=ext4 --size=500
part pv.202002 --grow --size=1
```

```
volgroup vg_host07 --pesize=4096 pv.202002
logvol / --fstype=ext4 --name=lv_root --vgname=vg_host07 --grow --size=1024 --maxsize=51200
logvol swap --name=lv_swap --vgname=vg_host07 --grow --size=1228 --maxsize=1228

repo --name="Oracle Linux Server"  --baseurl=http://192.0.2.1/OL6.5/ --cost=100

%packages
@base
@client-mgmt-tools
@console-internet
@core
@debugging
@directory-client
@hardware-monitoring
@java-platform
@large-systems
@network-file-system-client
@performance
@perl-runtime
@server-platform
@server-policy
@system-admin-tools
pax
python-dmidecode
oddjob
sgpio
certmonger
pam_krb5
krb5-workstation
perl-DBD-SQLite
%end
```

4. Create a new **host07** VM.
   a. From **dom0**, use the `mkdir` command to make the `/OVS/running_pool/host07` directory.

```
# mkdir /OVS/running_pool/host07
```

   b. Use the `cd` command to change to the `/OVS/running_pool/host07` directory.

```
# cd /OVS/running_pool/host07
```

   c. Use the `dd` command to create a 12 GB `system.img` file.
      - This command takes a few minutes to complete.

```
# dd if=/dev/zero of=system.img bs=1M count=12288
12288+0 records in
12288+0 records out
12884901888 bytes (13 GB) copied...
```

d. Use the `cp` command to copy the `vm.cfg` file from the `/OVS/running_pool/host01` directory to the current directory.

```
# cp /OVS/running_pool/host01/vm.cfg .
```

- You use the `vi` editor to change this `vm.cfg` file as instructed in step 4e.

Note: A pre-configured `vm.cfg` file exists on **dom0** in the `/OVS/seed_pool/host07` directory.

- If you do not wish to edit the `vm.cfg` file as instructed in step 4e, you can use the `cp` command to copy `/OVS/seed_pool/host07/vm.cfg` to `/OVS/running_pool/host07/vm.cfg`. If you use this `vm.cfg` file, you do not need to edit the file in step 4e.

e. Use the `vi` editor to edit the `vm.cfg` file. Change the file to make it like the following.

- Changes and additions are in bold.
- Delete any lines in the file that are not shown below.

```
# vi vm.cfg
# Automatically generated xen config file
name = "host07"
builder = "hvm"
memory = "1536"
boot = 'cd'
disk = [ 'file:/OVS/running_pool/host07/system.img,hda,w',
         'file:/OVS/seed_pool/OracleLinux-R6-U5-Server-x86_64-dvd.iso,hdc:cdrom,r']
vif = [ 'mac=00:16:3e:00:01:07,bridge=virbr0']
device_model = '/usr/lib/xen/bin/qemu-dm'
kernel = '/usr/lib/xen/boot/hvmloader'
vnc=1
vncunused=1
vcpus = 1
timer_mode = 0
apic = 1
acpi = 1
pae = 1
serial = 'pty'
on_reboot = 'restart'
on_crash = 'restart'
usb = 1
usbdevice = 'tablet'
```

5. Connect to the **host07** guest by using **vncviewer**.
   a. Use the `xm shutdown` command to shut down the **host01** VM.
      - The available memory on **dom0** allows a maximum of only three VMs to be running.
      - Therefore, it is necessary to shut down one VM to start a new VM.

```
# xm shutdown -w host01
Domain host01 terminated
All domains terminated
```

      - If the `xm shutdown` command is taking more than a few seconds to complete, use `CTRL-C` to kill the command and run the following `xm destroy` command.

```
# xm destroy host01
```

   b. Run the `xm create` command to create the **host07** VM.

```
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host07 (id=...)
```

   c. Determine the VNC port number for **host07** by running the `xm list -l host07 | grep location` command.

```
# xm list -l host07 | grep location
               (location 0.0.0.0:5900)
               (location 3)
```

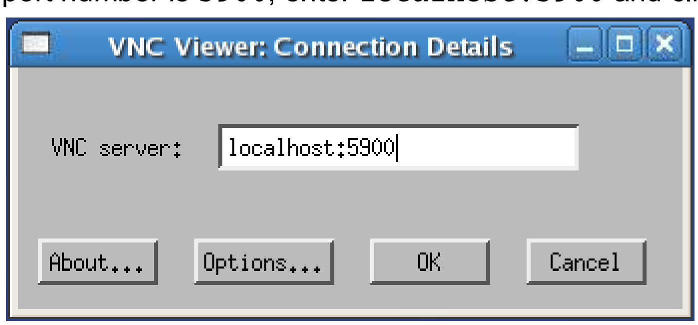
      - The sample shown indicates that the port number is **5900**. Your port number might be different.

   d. Run the `vncviewer&` command.

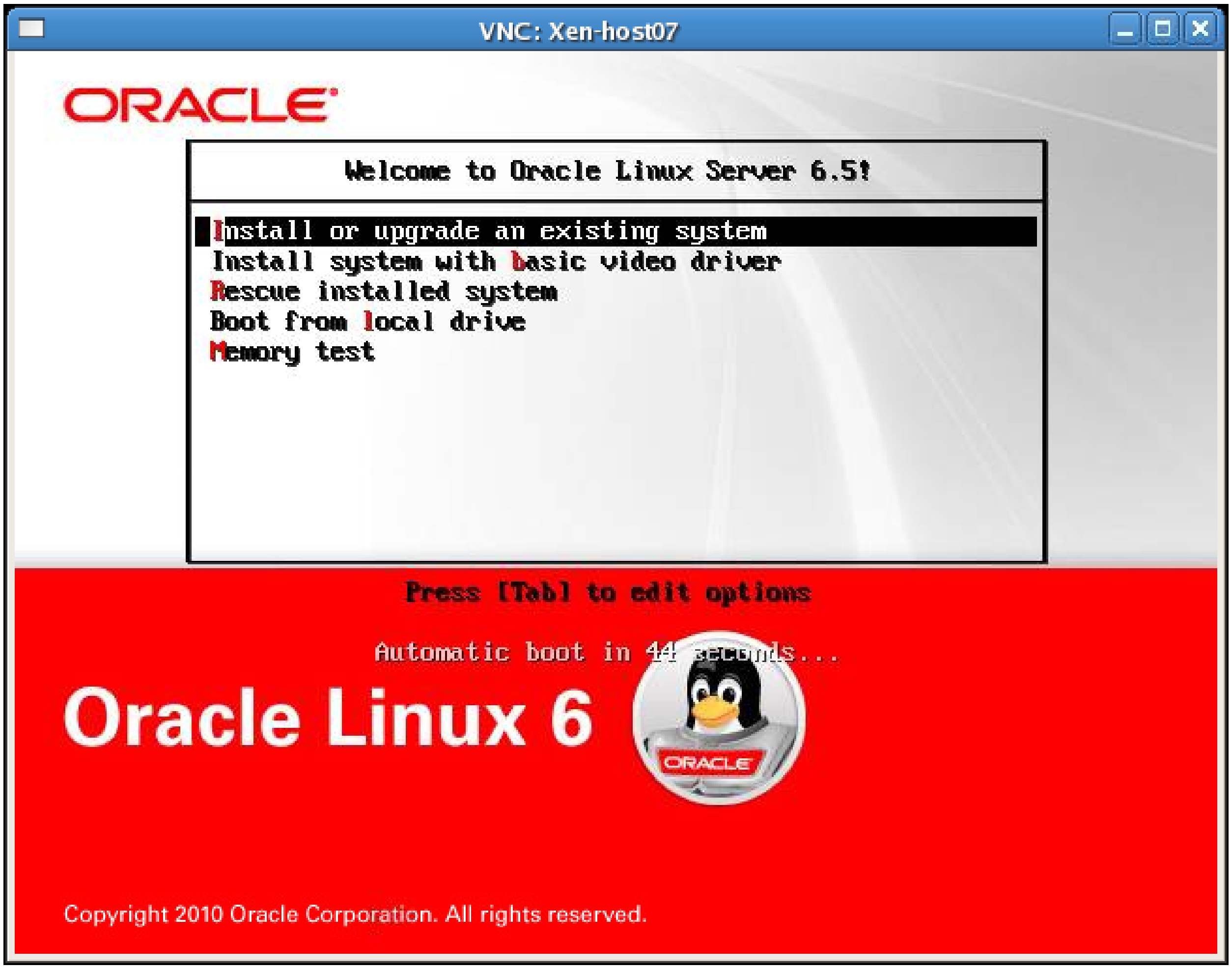
```
# vncviewer&
```

      - The **VNC Viewer: Connection Details** dialog box is displayed.

   e. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list -l host07 | grep location` command. For example, if the port number is `5900`, enter `localhost:5900` and click **OK**.



- The Oracle Linux boot menu appears:

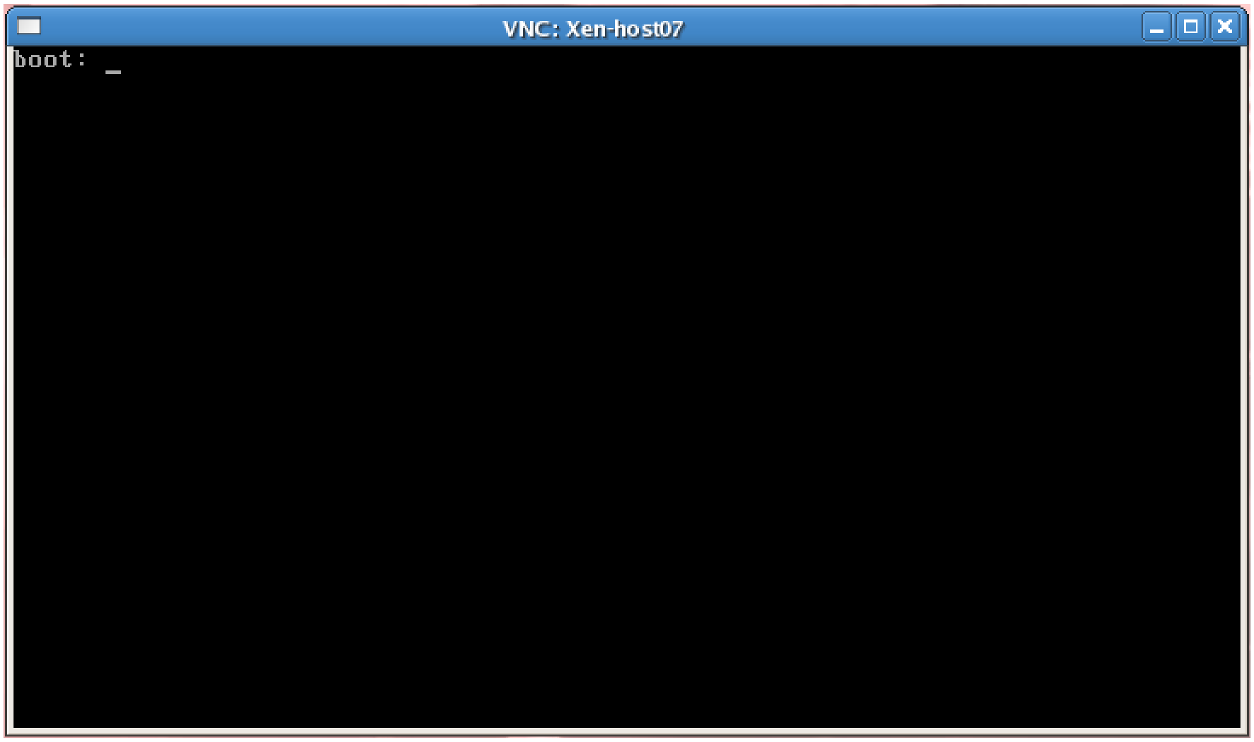


- The Oracle Linux boot menu screen appears for only 60 seconds, after which the “Install or upgrade an existing system” menu option is selected by default.
- If you do not see this screen, meaning the 60-second timeout has expired, click the `X` in the top-right corner of the current screen to close it, enter the following command from **dom0**, and begin step 5 again starting with 5b.

```
# xm destroy host07
```

f. From the Oracle Linux boot menu, press **Esc** to exit to the boot prompt.

- The boot prompt is shown:

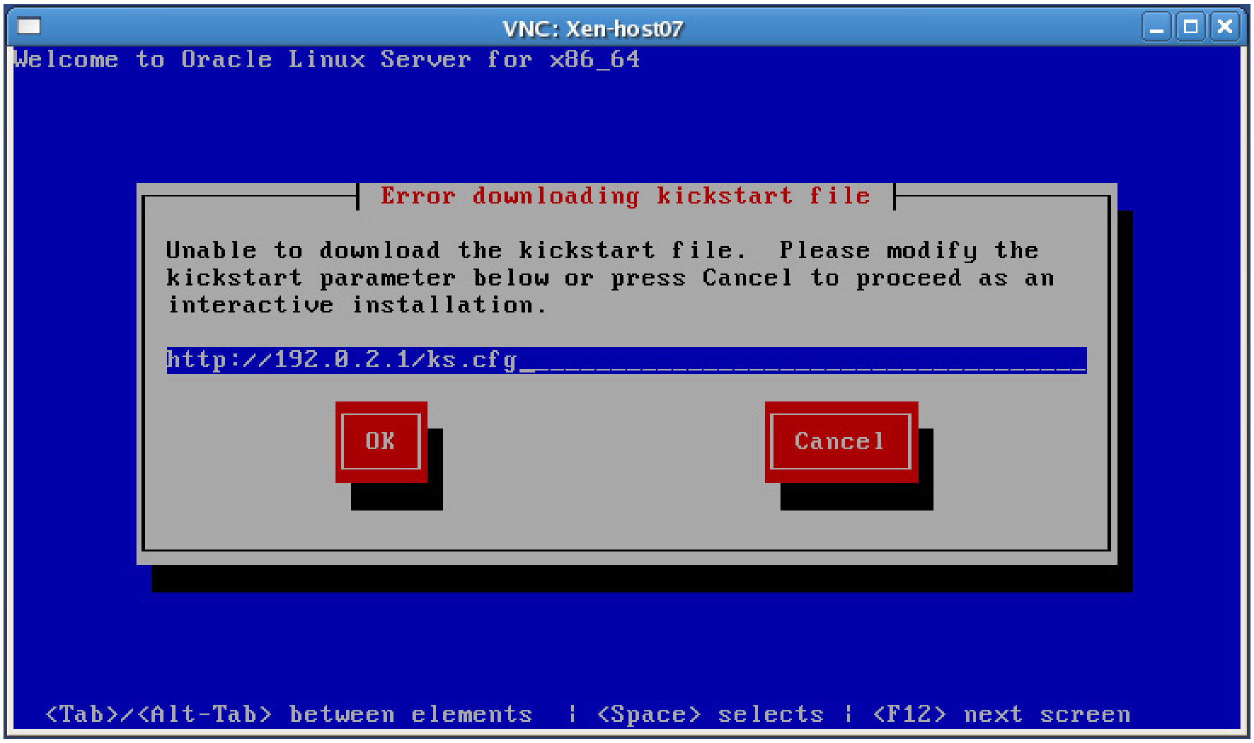


6. Initiate the kickstart installation.

   a. Enter the following command from the boot prompt, and press **Enter** to continue.

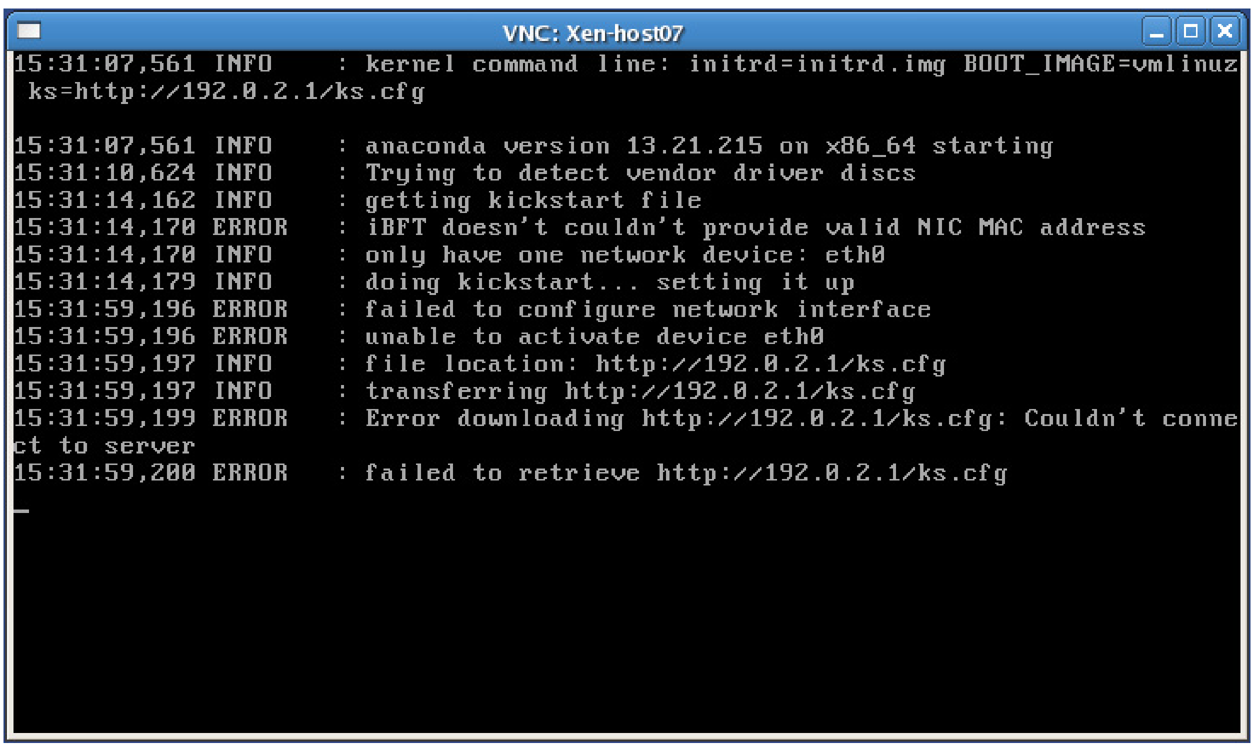
```
boot: linux ks=http://192.0.2.1/ks.cfg
```

- The unattended installation begins but fails to complete. The following screen is displayed.



b. Press **Alt + F3** to display the logging information in the alternate console.

- The following screen is displayed.



- As the errors indicate, the network interface failed to configure, which resulted in the inability to retrieve the Kickstart file, `ks.cfg`, from the `192.0.2.1` HTTP server.

c. Close the window by clicking the `X` in the top-right corner of the window.

7. Determine the problem with the network interface failing to configure.
   - The network interface normally obtains an IP address from DHCP running on the server.

   From **dom0**, use the `service` command to view the status of the `dhcpd` service.

```
[dom0]# service dhcpd status
dhcpd is stopped
```

   - The `dhcpd` service is stopped in this example.
   - Your options are to either (1) configure DHCP on the server, or (2) include the network information in the `boot` command.
   - The quickest method is option 2, to provide the network information in the `boot` command.

8. Re-initiate the Kickstart installation.

   a. From **dom0**, use the `xm destroy` command to stop the **host07** VM.

```
# xm destroy host07
```

   b. Use the `xm create` command to create the **host07** VM.

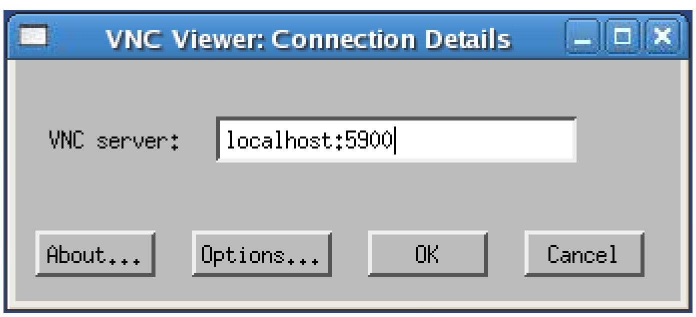
```
# xm create vm.cfg
...
```

   c. Use the `vncviewer` command to connect to the **host07** VM.

```
# vncviewer
```

   - The VNC Viewer: Connection Details dialog box is displayed.

   d. Enter `localhost:<port_number>`, substituting the port number displayed from the `xm list -l host07 | grep location` command in step 5c. For example, if the port number is `5900`, enter `localhost:5900` and click **OK**.



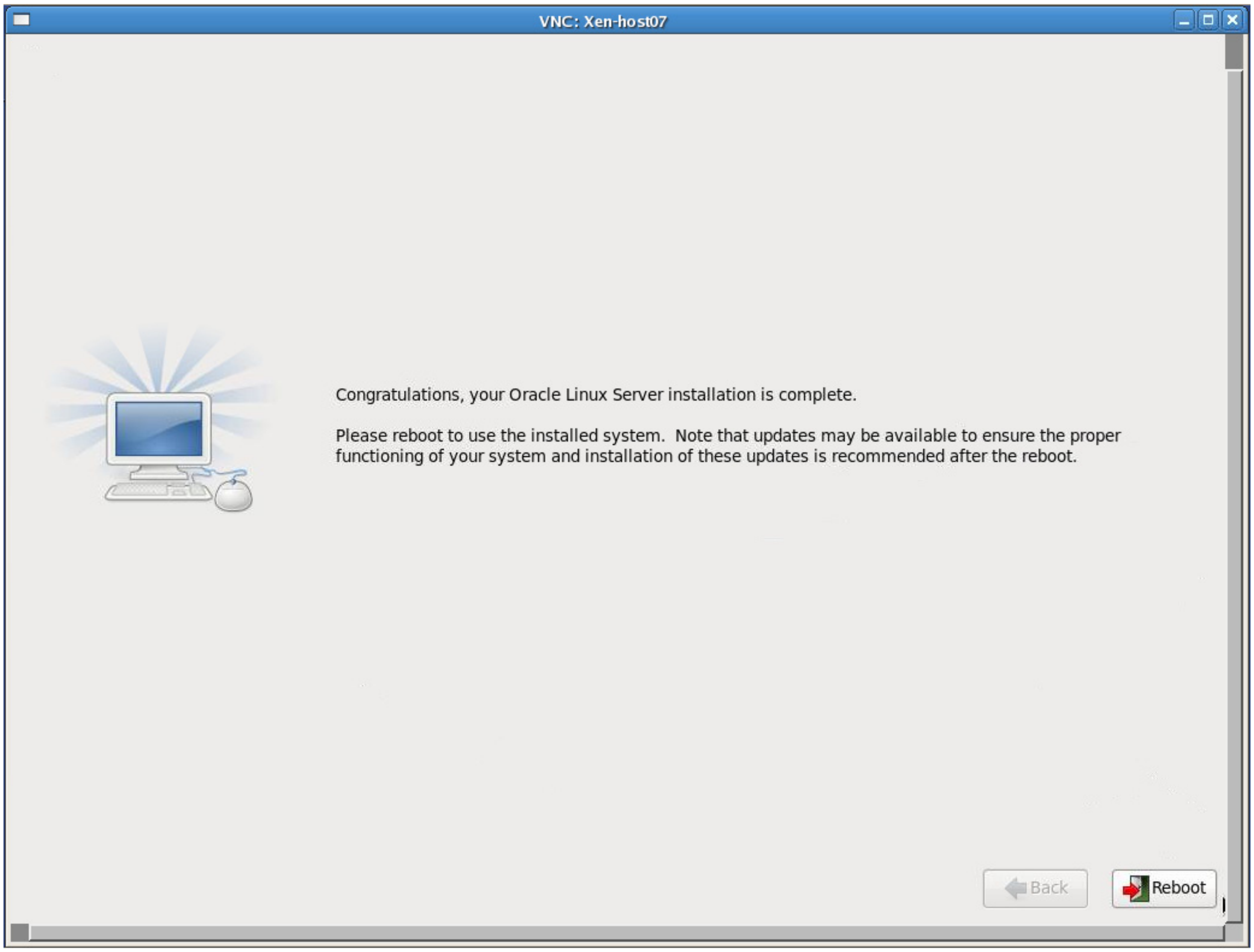
   - The Oracle Linux boot menu appears.

   e. From the Oracle Linux boot menu, press **Esc** to exit to the boot prompt.
   - The boot prompt is shown.

   f. Include the following network interface configuration information in addition to the location of the `ks.cfg` file in the boot command.
   - IP address (`ip=192.0.2.200`)
   - Netmask (`netmask=255.255.255.0`)
   - Gateway (`gw=192.0.2.1`)

```
boot: linux ip=192.0.2.200 netmask=255.255.255.0 gw=192.0.2.1
ks=http://192.0.2.1/ks.cfg
```

- The unattended installation begins and is successfully able to configure the network interface.
- When the installation is complete, you are prompted to reboot.



g. Click **Reboot** when prompted.

9. Log in to **host07** and verify the installation.

   a. From **dom0**, use the `ssh` command to connect to **host07** as the `root` user. The password is `oracle`.

   - Use the IP address for **host07** because the `/etc/hosts` file on **dom0** does not contain an entry to resolve the host name.
   - You need to wait a few seconds to allow **host07** to reboot.

```
[dom0]# ssh 192.0.2.107
The authenticity of host '192.0.2.107 (192.0.2.107)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.0.2.107' (RSA) to the list of
known hosts.
root@192.0.2.107's password: oracle
```

b. Use the `hostname` command to confirm that you are logged on to the **host07** VM.

```
# hostname
host07.example.com
```

c. Use the `ifconfig` command to display the network configuration.

```
# ifconfig
...
eth0      Link encap:Ethernet
          inet addr:192.0.2.107
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1
...
```

d. Use the `df` command to list the mounted partitions.

```
# df -h
Filesystem                      Size Used Avail Use% Mounted on
/dev/mapper/vg_host07-lv_root    11G 2.0G  7.6G  21%  /
tmpfs                          750M    0  750M   0%  /dev/shm
/dev/xvda1                     477M  55M  398M  12%  /boot
```

Practice 5-3: Using Rescue Mode

Overview

In this practice, you do the following:

- Corrupt a file on **host07** to cause boot failure.
- Boot into rescue mode to correct the file.

Assumptions

You are the `root` user on **host07**.

Tasks

1. Create an error in the `/etc/fstab` file to cause boot failure.
   a. Make a backup of `/etc/fstab`. Use the `vi` command to change `host07-lv_root` to `host7-lv_root` in the `/etc/fstab` file as follows:

```
# cp /etc/fstab /etc/fstab.backup
# vi /etc/fstab
/dev/mapper/vg_host07-lv_root                     (old entry)
/dev/mapper/vg_host7-lv_root                      (new entry)
```

   b. Use the `reboot` command to reboot **host07**.

```
# reboot
...
Connection to host07 closed.
[dom0]#
```

2. Attempt to log in to **host07**.
   a. From **dom0**, run the `xm list -l host07 | grep location` command to determine the VNC port number for **host07**.

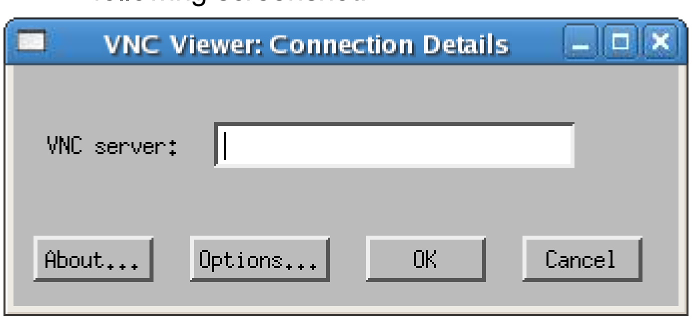
```
[dom0]# xm list -l host07 | grep location
               (location 0.0.0.0:5900)
               (location 3)
```

   - In this example, the VNC port number is `5900`. This might not be true in your case.

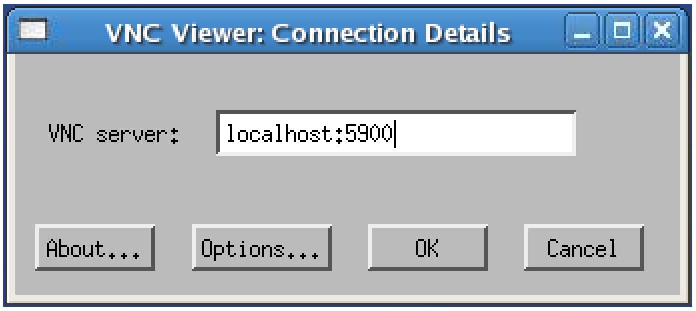
   b. Run the `vncviewer&` command:

```
[dom0]# vncviewer&
```

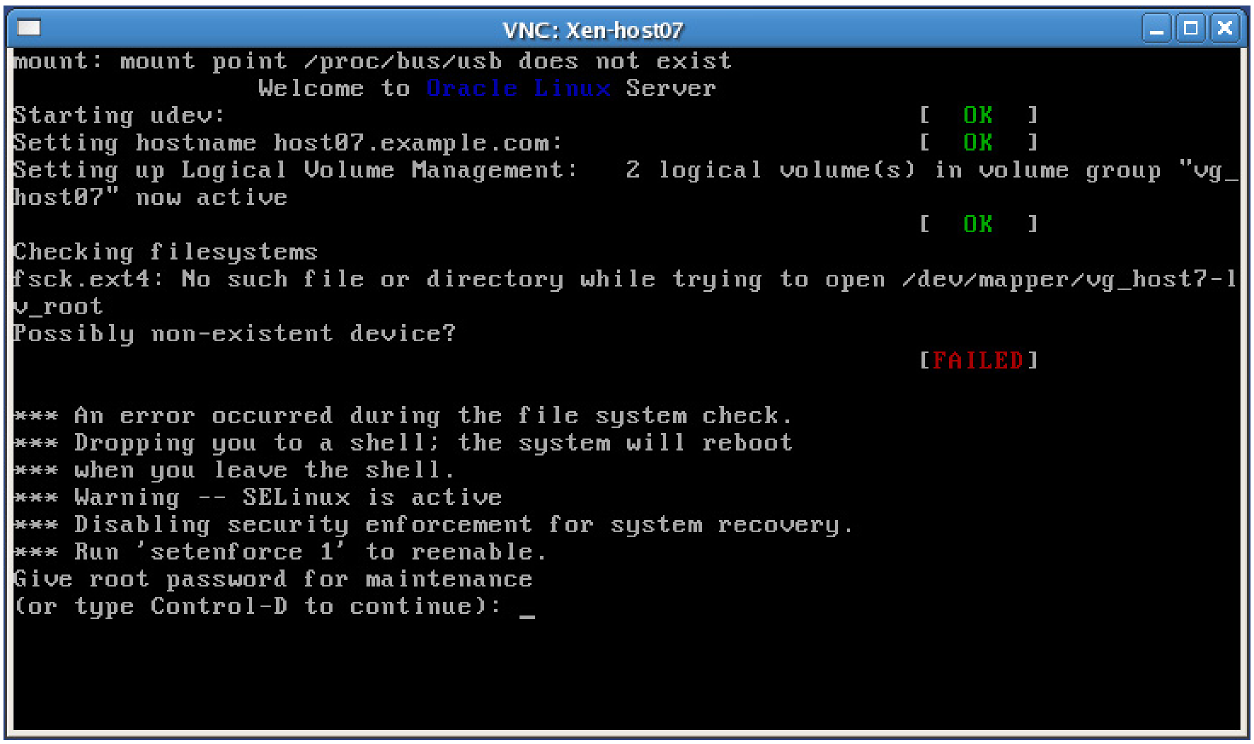
   - The **VNC Viewer: Connection Details** dialog box is displayed as shown in the following screenshot:



c. Enter the `localhost:<port_number>` command, substituting the VNC port number displayed from the previous `xm list` command. For example, if the port number is `5900`, enter `localhost:5900` as shown and click **OK**.



- The following screen shows that an error occurred during the file system check during the boot process.



d. Close the window by clicking the `X` in the upper right corner of the window.

3. Shut down **host07**.

a. Run the `xm destroy host07` command to shut down the **host07** VM. Run `xm list` to display the running VMs.

- The output shown is a sample, the `ID` and `Time(s)` values will be different on your system.

```
# xm destroy host07
# xm list
Name                 ID    Mem  VCPUs      State    Time(s)
Domain-0              0   2048      2     r-----      281.1
host02                2   1536      1     -b----      159.0
host03                3   1536      1     -b----       13.2
```

- Notice that **host07** is no longer active. You have two guests (**host02** and **host03**) running.

4. Configure **host07** to boot from Oracle Linux 6.5 installation media.
   - The procedure applies to Oracle VM Server for x86 version 2.2.1 Hardware Virtualized (HVM) Guests.
   - For Para-virtualized (PVM) Guests, refer to MOS note 549410.1.

   Use the `vi` editor to change the "`boot`" entry in the **host07** `vm.cfg` file from `boot = 'cd'` to `boot = 'd'`.

```
# cd /OVS/running_pool/host07
# vi vm.cfg
...
boot = 'cd'                                  (old entry)
boot = 'd'                                   (new entry)
...
```

5. Start the **host07** VM.

   Run the `xm create vm.cfg` command to start the **host07** VM. Run `xm list` to display the running VMs.

```
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host07 (id=#)
# xm list
Name                 ID    Mem  VCPUs      State    Time(s)
Domain-0              0   2048      2     r-----      281.1
host02                2   1536      1     -b----      159.0
host03                3   1536      1     -b----       13.2
host07               14   1536      1     -b----       13.2
```

   - Notice that **host07** is now active.

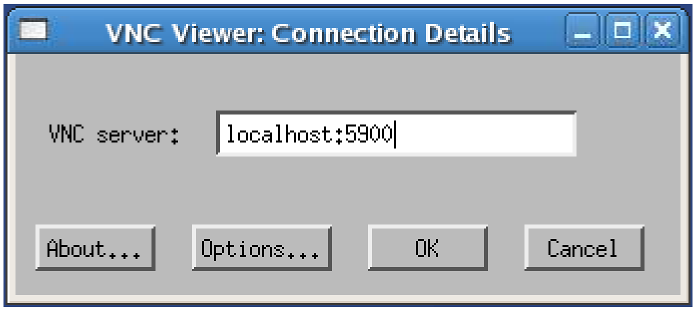
6. Log in to **host07**.

   a. Run the `vncviewer&` command:

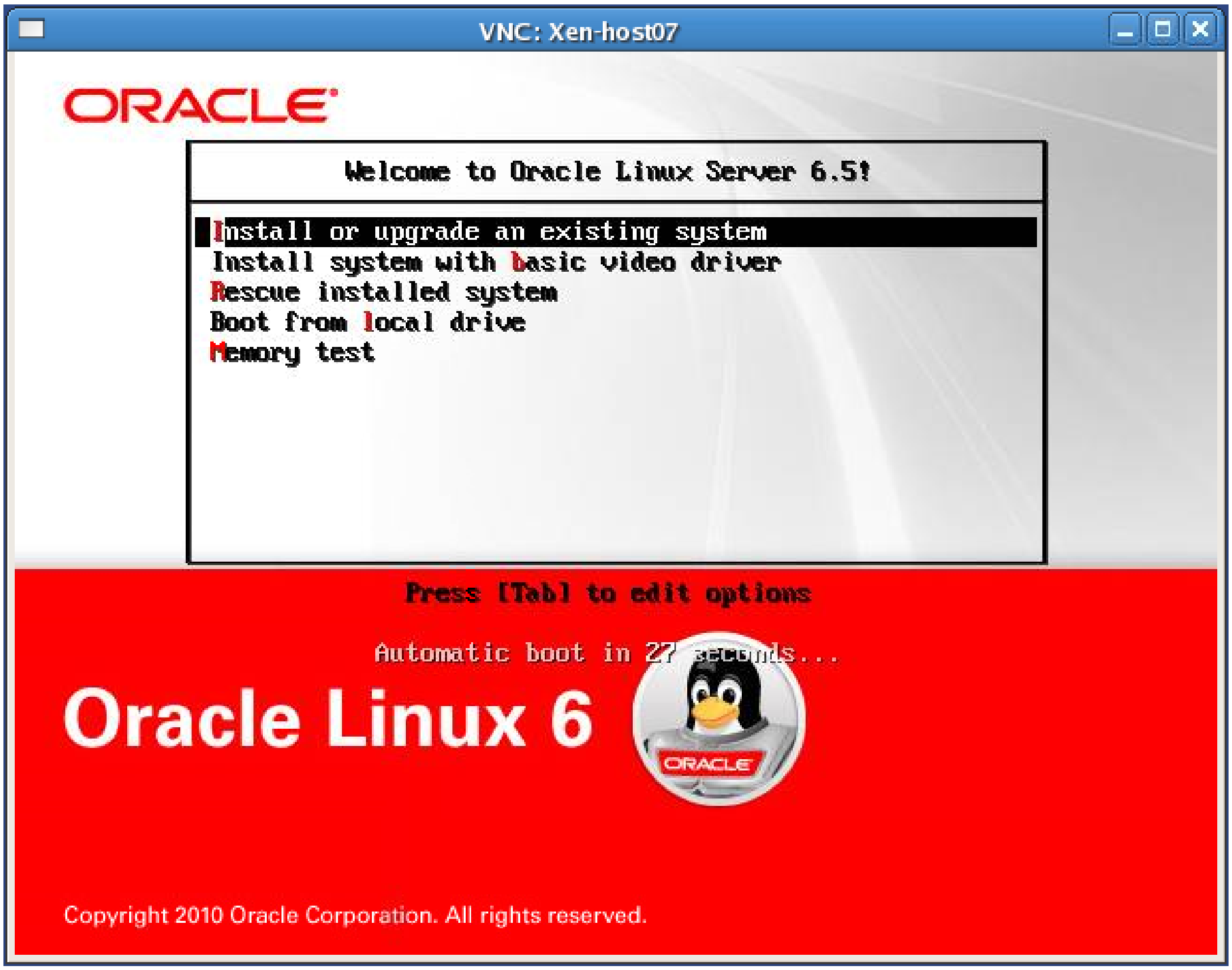
```
# vncviewer&
```

   - The **VNC Viewer: Connection Details** dialog box is displayed as shown in the following screenshot:

b. Enter the `localhost:<port_number>` command, substituting the VNC port number obtained in step 2a. For example, if the port number is `5900`, enter `localhost:5900` as shown and click **OK**.



- The Oracle Linux boot menu appears as shown:

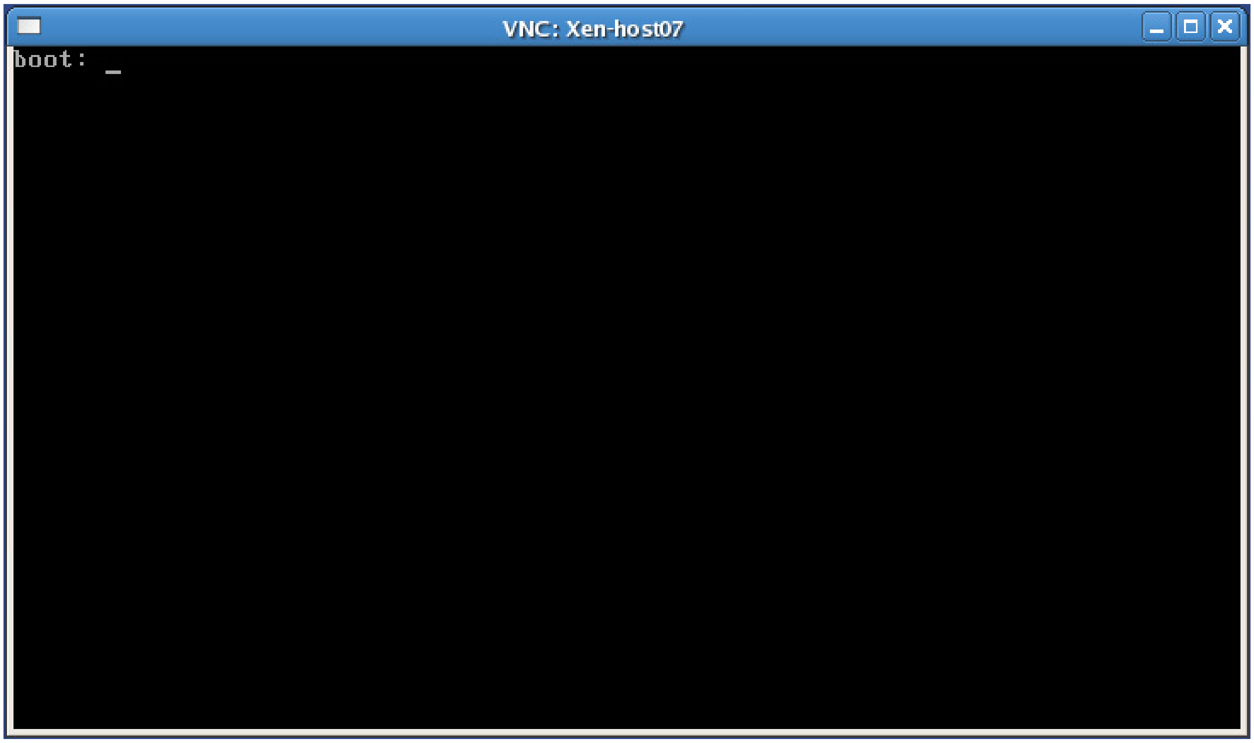


- The Oracle Linux boot menu screen appears for only 60 seconds after which the “Install or upgrade an existing system” menu option is selected by default.
- If you do not see this screen, meaning the 60-second timeout has expired, click the `X` in the top-right corner of the screen to close it, enter the following command from **dom0**, and begin again starting with step 5.

```
# xm destroy host07
```

c. From the Oracle Linux boot menu, press the **Esc** key to display the `boot:` prompt. The following screen appears:

- Alternatively, you could use the arrow keys to select "**Rescue installed system**" from the boot menu.

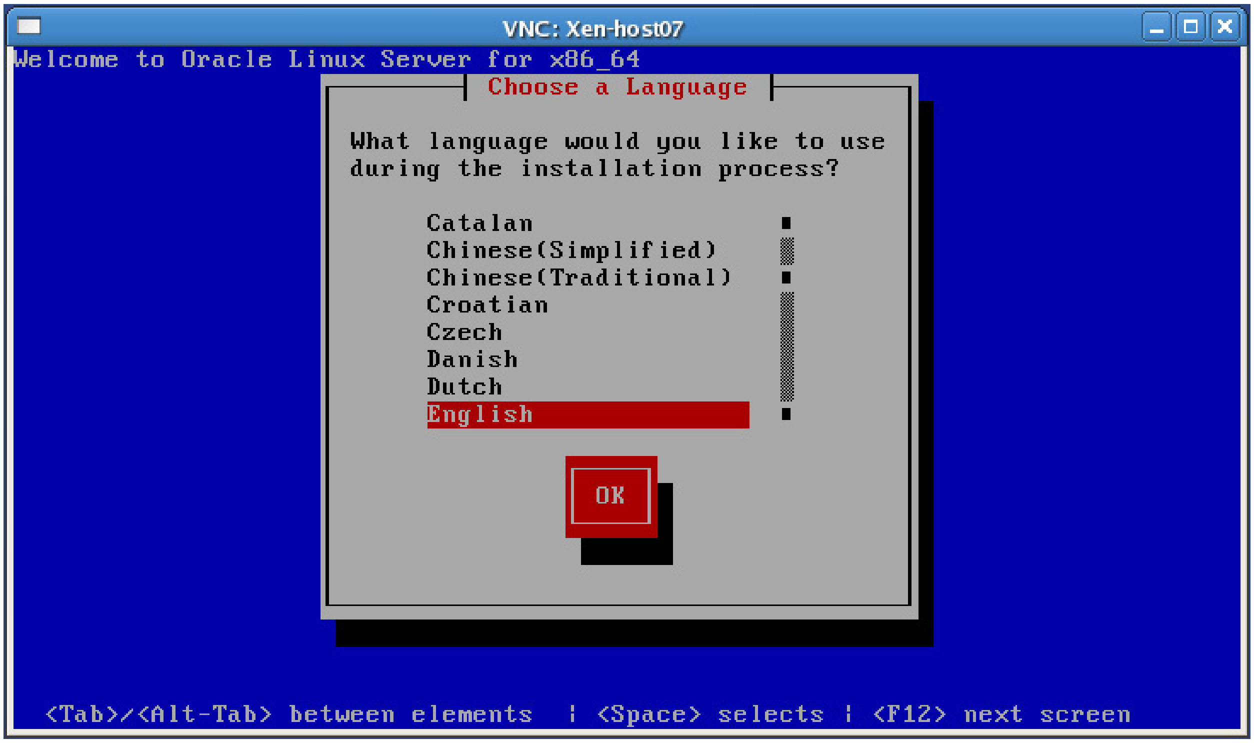


7. Boot into Rescue Mode.

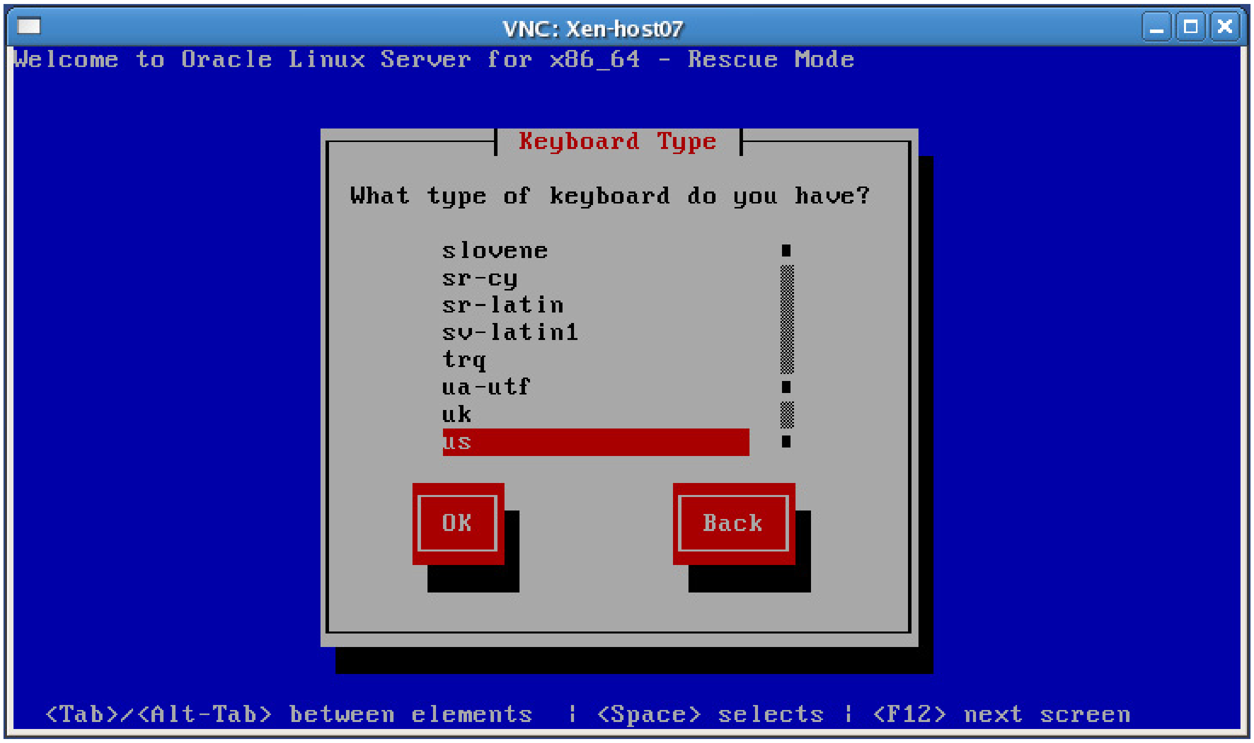
a. Enter `linux rescue` at the `boot:` prompt and press **Enter**.

```
boot: linux rescue
```

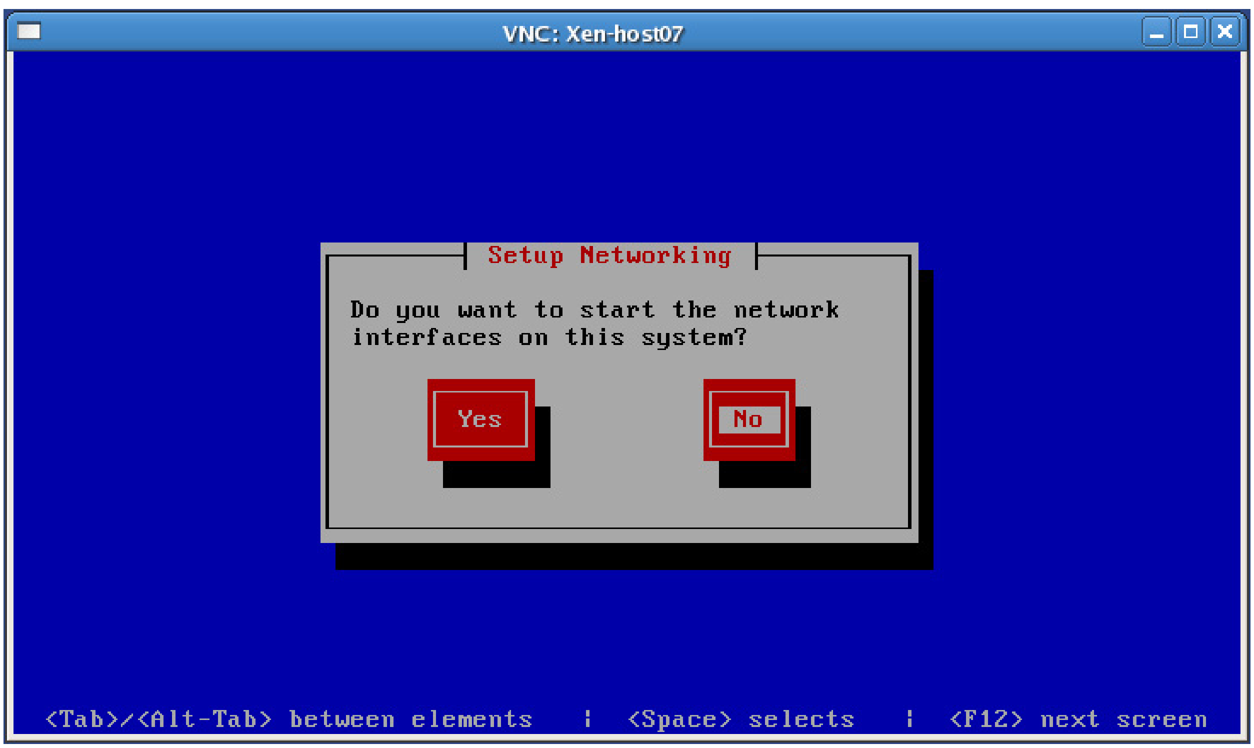
b. Choose the desired language using the arrow keys. Use the **Tab** key to select **OK** and press **Enter**.



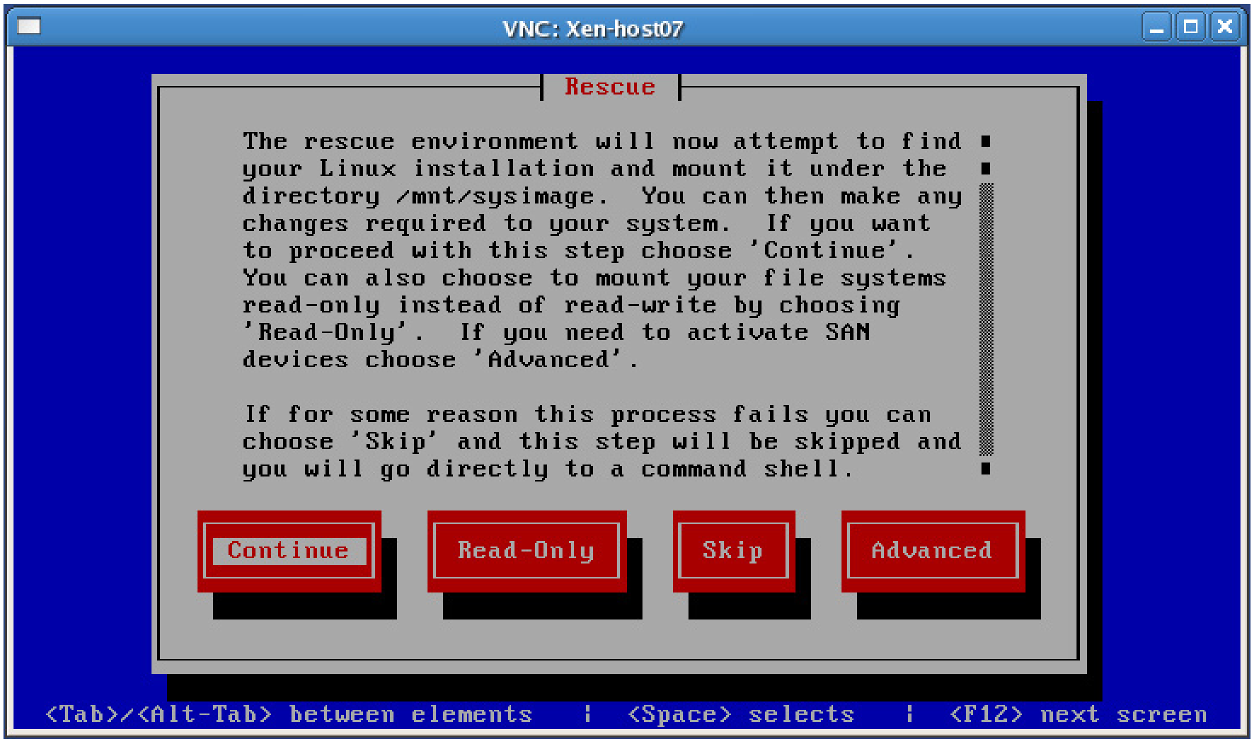
c. Choose the desired Keyboard Type by using the arrow keys. Use the **Tab** key to select **OK** and press **Enter**.



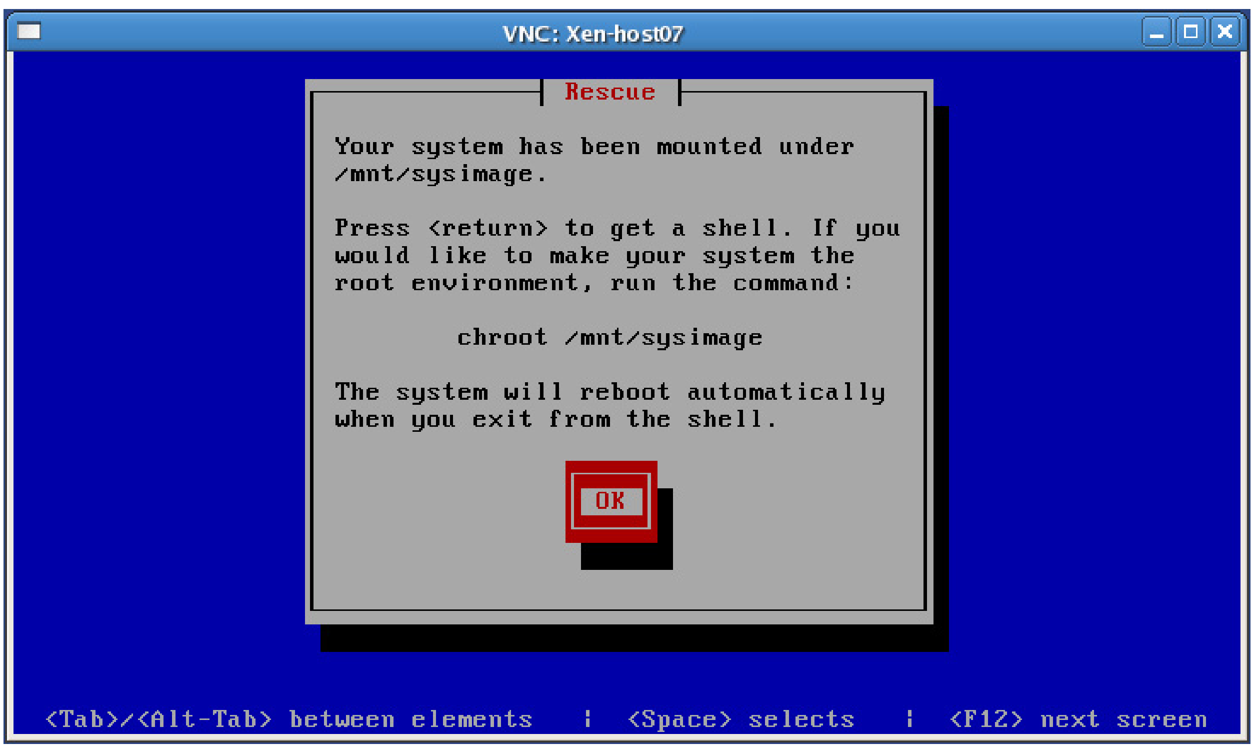
d. Starting the network interfaces is unnecessary. Use the **Tab** key to select **No** and press **Enter**.



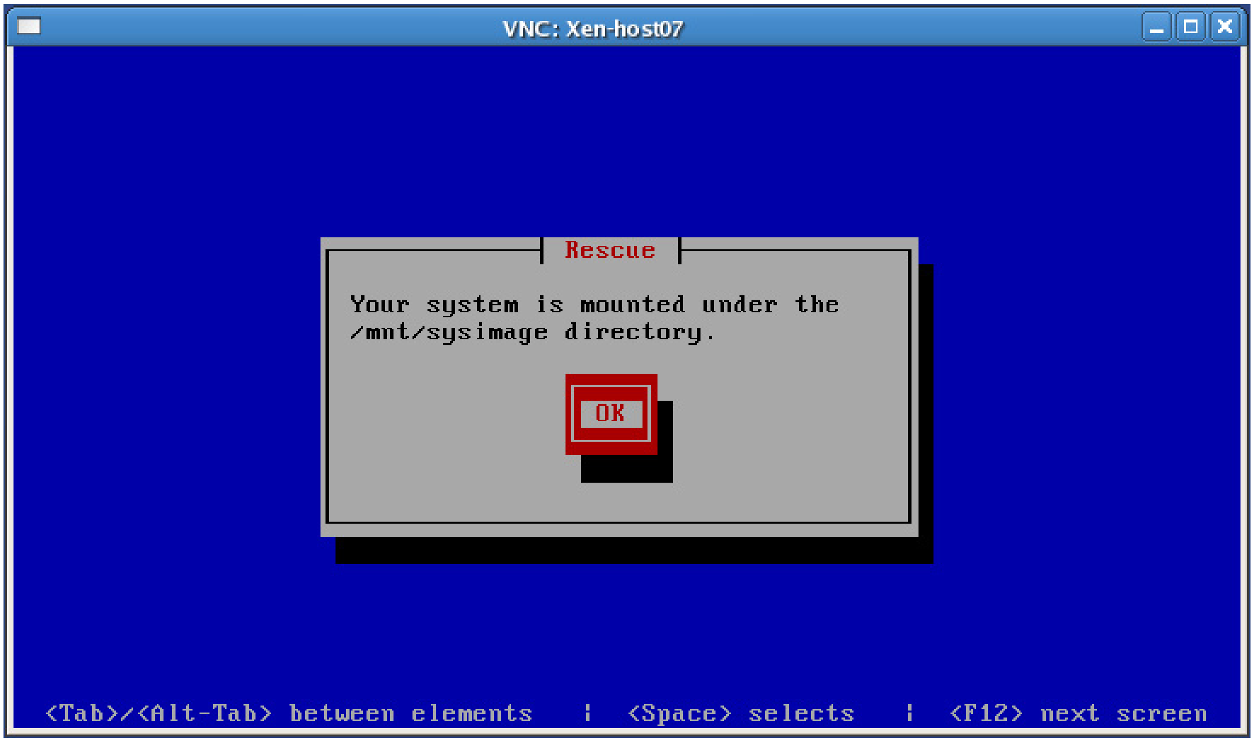
e. Review the information displayed on the following screen. Use the **Tab** key to select **Continue** and press **Enter**.



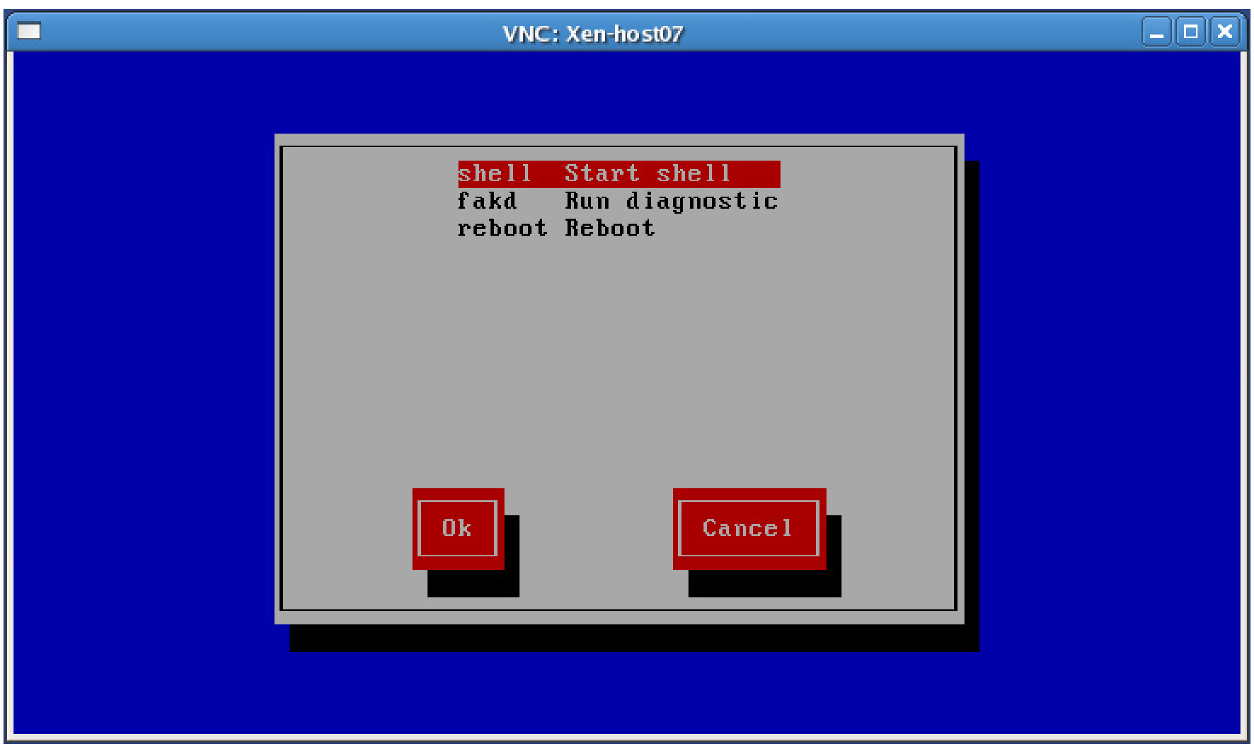
f. Review the information displayed on the following screen. Press **Enter** to continue.



g. Review the information displayed on the following screen. Press **Enter** to continue.



h. Choose the “**shell Start shell**” menu option (as shown). Use the **Tab** key to select **OK** and press **Enter** to display a shell prompt.



- A shell prompt is displayed.

8. Repair the corrupted `/etc/fstab` file.
   a. Use the `df` command to view the mounted file systems.

```
# df
Filesystem                     ...          Mounted on
...
/dev/mapper/vg_host07-lv_root ...          /mnt/sysimage
/dev/xvda1                     ...          /mnt/sysimage/boot
...
```

   - Notice that the file systems are mounted under the `/mnt/sysimage` directory.

   b. Use the `pwd` command to determine the current directory location. Use the `ls` command to view the contents of the current / directory.

```
# pwd
/
# ls
bin  firmware  lib64      mnt      proc  selinux  usr
dev  init      lib64_old  modules  root  sys      usr_old
etc  lib       lib_old    oldtmp   sbin  tmp      var
```

   c. Use the `chroot` command to change the root partition of the rescue mode environment to the root partition of your file system.

```
# chroot /mnt/sysimage
```

   d. Use the `df` command to view the mounted file systems.

```
# df
Filesystem                     ...          Mounted on
/dev/xvda1                     ...          /boot
...
```

   - Notice that the file system mount points are different.

   e. Use the `ls` command to view the contents of the current / directory.

```
# ls
bin    cgroup  etc   lib   lost+found  misc  net   proc
sbin   srv     tmp   var   boot        dev   home  lib64
media  mnt     opt   root  selinux     sys   usr
```

   - Notice that the contents of the / directory are different.

   f. Use the `cp` command to restore `/etc/fstab` from `/etc/fstab.backup`.

```
# cp /etc/fstab.backup /etc/fstab
```

   g. Use the `exit` command to exit the `chroot` environment.

```
# exit
```

   h. Close the window by clicking the `X` in the top right corner of the window.

9. Boot **host07** from the system hard drive.
   a. From **dom0**, use the `vi` editor to change the “`boot`” entry in the **host07** `vm.cfg` file from `boot = 'd'` back to `boot = 'cd'`.

```
# cd /OVS/running_pool/host07
# vi vm.cfg
...
boot = 'd'                              (old entry)
boot = 'cd'                             (new entry)
...
```

   b. Use the `xm destroy host07` command to shut down the **host07** VM.

```
# xm destroy host07
```

   c. Use the `xm create vm.cfg` command to start the **host07** VM.

```
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host07 (id=#)
```

10. Log in to **host07**.

    From **dom0**, use the `ssh` command to connect to `192.0.2.107` (**host07**). The `root` password is `oracle`.

    - Use the IP address because the `/etc/hosts` file on **dom0** does not contain an entry to resolve **host07**.
    - You need to wait a few seconds for the reboot to complete.

```
# ssh 192.0.2.107
root@host07's password: oracle
```

    - Notice that your system successfully boots from the hard drive and you can log in.

11. Remove **host07** and clean up **dom0**.
    a. Use the `shutdown -h now` command to shut down **host07**.

```
# shutdown -h now
...
The system is going down for halt NOW!
Connection to 192.0.2.107 closed by remote host.
Connection to 192.0.2.107 closed
```

    b. From **dom0**, use the `rm -r` command to remove the `/OVS/running_pool/host07/` directory.

```
# cd /OVS/running_pool
# rm -r host07
rm: descent into directory 'host07'? y
rm: remove regular file 'host07/system.img'? y
rm: remove regular file 'host07/vm.cfg'? y
rm: remove directory 'host07'? y
```

c. Use the `/bin/rm -r` command to remove the `/var/www/html/OL6.5/` directory.

```
# cd /var/www/html
# /bin/rm -r OL6.5
```

12. Restart **host01** VM.

a. From **dom0**, change to the `/OVS/running_pool/host01` directory and use the `xm create` command as follows:

```
# cd /OVS/running_pool/host01
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host01 (id=6)
```

b. Use the `xm list` command to verify that **host01**, **host02**, and **host03** are running and that **host07** is not running.

```
# xm list
Name                    ID    Mem  VCPUs       State       Time(s)
Domain-0                 0   2048      2      r-----         758.9
host01                   4   1536      1      -b----          37.4
host02                   5   1536      1      -b----          37.3
host03                   9   1536      1      -b----         109.3
```

Practices for Lesson 6: Samba Services

Chapter 6

Practices for Lesson 6

Practices Overview

In these practices, you configure a Samba server and access the Samba shares on the server from an Oracle Linux client host.

Practice 6-1: Installing the Samba Packages

Overview

In this practice, you install the packages necessary to configure Samba services on the **host03** VM.

Assumptions

You are the `root` user on **dom0**.

Tasks

1. Use the `ssh` command to log in to **host03** from **dom0**.
   - The `root` password is `oracle`.

```
[dom0]# ssh host03
root@host03's password: oracle
Last login: ...
[host03]#
```

2. Install the `samba` packages on **host03**.
   a. From **host03**, use the `rpm -qa` command to list the installed `samba` packages.

```
# rpm -qa | grep samba
samba-common-3.6.9-164.el6.x86_64
samba-winbind-3.6.9-164.el6.x86_64
samba4-libs-4.0.0-58.el6.rc4.x86_64
samba-client-3.6.9-164.el6.x86_64
samba-winbind-clients-3.6.9-164.el6.x86_64
```

   - In this example, several `samba` packages are installed.
   - The `samba-`*`<version>`* package still needs to be installed.

   b. Run the `yum install` command to install the `samba` package.

```
# yum install samba
...
Transaction Summary
==============================================================
Install       1 Package(s)
Total download size: 5.0 M
Installed size: 18 M
Is this ok [y/N]: y
...
Complete!
```

Practice 6-2: Configuring a Samba Server

Overview

In this practice, you configure **host03** to act as a Samba server by performing the following:

- Start the `smb` service.
- Open ports 137 through 139 in the firewall of the Samba server (**host03**).
- Create user `user01` on the Samba server.
- Edit the `smb.conf` file.
- Use the `testparm` command to check the syntax of the `sbm.conf` file.
- Create a password for user `user01`.

Assumptions

You are the `root` user on **host03**.

Tasks

1. Start the `smb` service on **host03**.
   a. Use the `service` command to start the `smb` service.

```
# service smb start
Starting SMB services:                                 [  OK  ]
```

   b. Use `chkconfig` to configure the `smb` service to start at boot time.

```
# chkconfig smb on
# chkconfig smb --list
smb     0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

2. Modify `iptables` to allow access on ports `137` through `139`.
   - You are not using Active Directory integration in the Oracle classroom environment.
   - If you were using Active Directory integration, you would also need to open port `445`.

   a. Use the `vi` editor to edit the `/etc/sysconfig/iptables` file. Use the `:set nu` command to turn on line numbers.

```
# vi /etc/sysconfig/iptables
...
:set nu
```

   b. Add the following entry after line 12:
      - This assumes that you added two lines in “Practice3-6: Configuring an OpenLDAP Server” (step 13). If you did not add two lines in Practice 3-6, add the following entry after line 10.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 137:139 -j
ACCEPT
```

   c. Use the `service` command to restart the `iptables` service.

```
# service iptables restart
...
```

d. Use the `service` command to restart the `smb` service after making changes to `iptables`.

```
# service smb restart
Shutting down SMB services:                         [  OK  ]
Starting SMB services:                              [  OK  ]
```

e. Use the `iptables –L INPUT` command to list the rules in the INPUT chain.

```
# iptables –L INPUT
...
ACCEPT    tcp    --    anywhere    anywhere    state NEW  tcp
dpts:netbios-ns:netbios-ssn
...
```

- Repeat step 2 if you do not see the above line.

3. Add a new user on **host03**.

a. User the `useradd` command to add `user01`.

```
# useradd user01
```

b. Use the `passwd` command to set the password to `oracle` for `user01`.

- Ignore the BAD PASSWORD warning messages.

```
# passwd user01
Changing password for user user01.
New password: oracle
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password: oracle
passwd: all authentication tokens updated successfully.
```

4. Edit the `smb.conf` file.

a. Use the `cd` command to change to the `/etc/samba` directory. Use the `ls` command to list the contents of the directory.

```
# cd /etc/samba
# ls
lmhosts  smb.conf  smbusers
```

b. Use the `vi` editor to edit the `smb.conf` file. Use the `:set nu` command to turn on line numbers.

```
# vi smb.conf
...
:set nu
```

c. At around line number 74, change `workgroup = MYGROUP` to `workgroup = GROUPA`.

```
    workgroup = GROUPA
```

- The `workgroup` parameter defines the workgroup name for your environment. In the classroom environment, this parameter has no effect.

d. At around line number 77, change `netbios name = MYSERVER` to `netbios name = SMB-HOST03`.
   - Remove the semicolon at the beginning of the line.

```
     netbios name = SMB-HOST03
```

   - The `netbios name` parameter is set to the name recognized by your Windows environment for your Samba server. In the classroom environment, this parameter has no effect.

e. At around line number 101, ensure that the `security` parameter is set to `user` and that the security parameter line is uncommented.
   - You do not need to make changes to this line.

```
     security = user
```

f. At around line number 248, examine the `[homes]` stanza.
   - You do not need to make changes to this stanza.
   - The default options for this share definition allow users to access their home directory as Samba shares from a remote location.

```
[homes]
          comment = Home Directories
          browseable = no
          writable = yes
;         valid users = %S
;         valid users = MYDOMAIN\%S
```

g. At around line number 254, immediately following the `[homes]` stanza, add a `[tmp]` stanza for the `/tmp` directory.
   - This stanza allows users to access the `/tmp` directory as a Samba share.

```
[tmp]
          path = /tmp
          writable = yes
          guest ok = yes
```

h. Save the changes to the `smb.conf` file and exit `vi`.

5. Use the `testparm` command to check the syntax of the `sbm.conf` file.
   - Press **Enter** when prompted.

```
# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows
limit (16384)
Processing section "[homes]"
Processing section "[tmp]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
<Press the ENTER key>
```

```
[global]
        workgroup = GROUPA
        netbios name = SMB-HOST03
        server string = Samba Server Version %v
        log file = /var/log/samba/log.%m
        max log size = 50
        idmap config * : backend = tdb
        cups options = raw

[homes]
        comment = Home Directories
        read only = No
        browseable = No

[tmp]
        path = /tmp
        read only = No
        guest ok = Yes

[printers]
        comment = All Printers
        path = /var/spool/samba
        printable = Yes
        print ok = Yes
        browseable = No
```

- If you do not specify a name for the configuration file with the `testparm` command, the command uses the default path name at `/etc/samba/smb.conf`.

6. Reload the `smb.conf` file.
   a. Run the `service smb reload` command to refresh the `smb.conf` file without having to stop and restart the `smb` service.

```
# service smb reload
Reloading smb.conf file:                                [  OK  ]
```

   b. Run the `service` command to view the status of the `smb` service.

```
# service smb status
smbd (pid ...) is running...
```

7. Create a Samba password for the `user01` user.

   Use the `smbpasswd` command to add user `user01` to the local `smbpasswd` file.

   - Set the password for `user01` to `MyOracle1`.

```
# smbpasswd -a user01
New SMB password: MyOracle1
Retype new SMB password: MyOracle1
Added user user01.
```

   - You use this password when accessing a Samba share from another Linux system or a Windows system as `user01`.

Practice 6-3: Accessing Samba Shares from a Client Host

Overview

In this practice, you do the following:

- Access the Samba shares that you set up on **host03** in the previous practice, from **host01**, which acts as an Oracle Linux Samba client.
- Mount and unmount a Samba share on **host01**.

Assumptions

All steps are performed from the **host01** VM except where indicated.

Tasks

1. From **dom0**, log in to **host01**.

   Use the `ssh` command to log in to **host01**.

   - The `root` password is `oracle`.

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[host01]#
```

2. From **host01**, access the Samba shares on **host03** as user `user01`.

   a. Use the `smbclient` command to access the `/tmp` directory on **host03**.

   - The Samba password for `user01` is `MyOracle1`.

```
# smbclient //host03/tmp -U user01
Enter user01's password: MyOracle1
Domain=[GROUPA] OS=[Unix] Server=[Samba 3.6.9-164.el6]
smb: \>
```

   b. If the `smbclient` command returns "`session setup failed: NT_STATUS_LOGON_FAILURE`," use the `service` command on **host03** to restart the `smb` service.

   - After restarting the `smb` service, run the `smbclient` command in step 2a again.

```
[host03]# service smb restart
Shutting down SMB services:                            [  OK  ]
Starting SMB services:                                 [  OK  ]
```

   c. At the `smb:` prompt on **host01**, use the `ls` command to list the files in the `/tmp` directory.

```
smb: \> ls
  .                       D        0  ...
  ..                     DR        0  ...
...
smb: \>
```

   d. Use the `exit` command to exit the `smb` session on **host01**.

```
smb: \> exit
```

e. Use the `smbclient` command to access the home directory for user `user01` on **host03**.
   - The Samba password for `user01` is `MyOracle1`.

```
# smbclient //host03/user01 -U user01
Enter user01's password: MyOracle1
Domain=[GROUPA] OS=[Unix] Server=[Samba 3.6.9-164.el6]
smb: \>
```

f. Use the `ls` command to list the files in the home directory for user `user01`.

```
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
smb: \>
```

   - The error message indicates a problem with SELinux.
   - SELinux is covered in “Lesson 16: SELinux.”

g. Use the `exit` command to exit the `smb` session.

```
smb: \> exit
```

h. To allow Samba users access to their home directories, set SELinux to permissive mode on **host03**.
   - You could configure SELinux to allow Samba users to access their home directories; however, for purposes of this practice, set SELinux to permissive mode.

```
[host03]# getenforce
Enforcing
[host03]# setenforce 0
[host03]# getenforce
Permissive
```

i. On **host01**, re-issue the `smbclient` command to access the home directory for user `user01` on **host03**.
   - The Samba password for `user01` is `MyOracle1`.

```
[host01]# smbclient //host03/user01 -U user01
Enter user01's password: MyOracle1
Domain=[GROUPA] OS=[Unix] Server=[Samba 3.6.9-164.el6]
smb: \>
```

j. Use the `ls` command to list the files in the home directory for user `user01`.

```
smb: \> ls
  .                      D        ...
  ..                     D        ...
  .bash_profile          H        ...
  .bashrc                H        ...
  .gnome2               DH        ...
  .mozilla              DH        ...
  .bash_logout           H        ...
...
smb: \>
```

- Because of the change in the SELinux mode, you can now list and access the files in `user01`'s home directory.

k. Use the `exit` command to exit the `smb` session.

```
smb: \> exit
```

3. On **host01**, mount and unmount a Samba share from your Oracle Linux client.

a. On **host01**, create a mount point for `user01`'s home directory.

```
# mkdir /homedir
```

b. Use the `mount` command to mount `user01`'s home directory on the newly created mount point.

- Use the `-t cifs` option to indicate that you are mounting a Samba or Windows type share.
- Specify read-only in the mount options.
- The Samba password for `user01` is `MyOracle1`.

```
# mount -t cifs -o username=user01,ro  //host03/user01 /homedir
Password: MyOracle1
```

c. Use the `df -hT` command to verify that the mount operation was successful.

```
# df -hT
Filesystem          Type  Size  Used  Avail  Use%  Mounted on
...
//host03/user01     cifs   11G  3.7G   5.9G   39% /homedir
```

- Notice that the file system type for `//host03/user01` is `cifs`.

d. Verify that the `/homedir` directory is read-only by using the `mount` command:

```
# mount | grep homedir
//host03/user01 on /homedir type cifs (ro)
```

e. List the contents of `/homedir`:

```
# ls /homedir
```

- Notice that the directory is empty.

f. On **host03**, use the `touch` command to create the `/home/user01/testfile` file.

```
[host03]# touch /home/user01/testfile
```

g. On **host01**, list the contents of `/homedir`:

```
# ls /homedir
testfile
```

- Notice that the `testfile` can now be seen from **host01**.

h. On **host01**, use the `umount` command to unmount the Samba share.

```
# cd
# umount /homedir
```

i. Use the `exit` command to log off **host01**.

```
# exit
logout
Connection to host01 closed.
```

j. Set SELinux to enforcing mode on **host03**.

```
[host03]# getenforce
Permissive
[host03]# setenforce 1
[host03]# getenforce
Enforcing
```

k. Use the `shutdown -h now` command to shut down **host03**.

```
[host03]# shutdown -h now
...
```

- You are instructed to shut down **host03** in preparation for Practice 7.

Practice 6-4: Accessing a Linux Samba Share from a Windows System

Overview

In this practice, you become familiar with procedures to access a Linux Samba share from a Windows system. You do not have a Windows system in the Oracle classroom environment. All you can do is read through the tasks in this practice to help understand the steps.

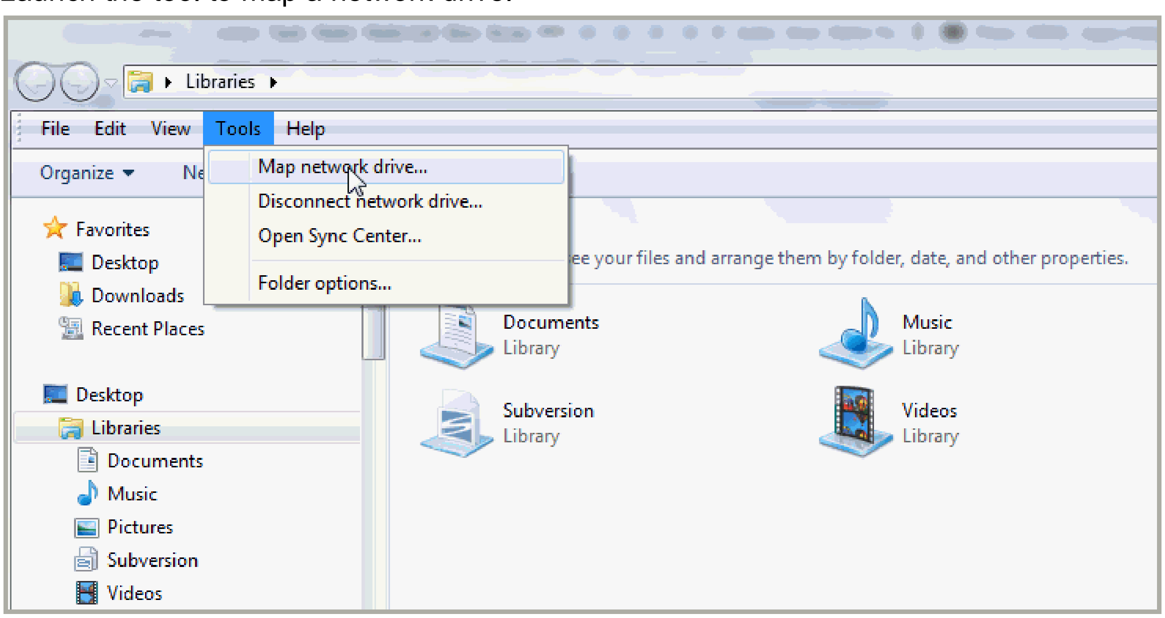
Assumptions

- This practice is not intended to be a hands-on exercise.
- The Linux Samba server is **host01**, IP address is `192.0.2.101`.

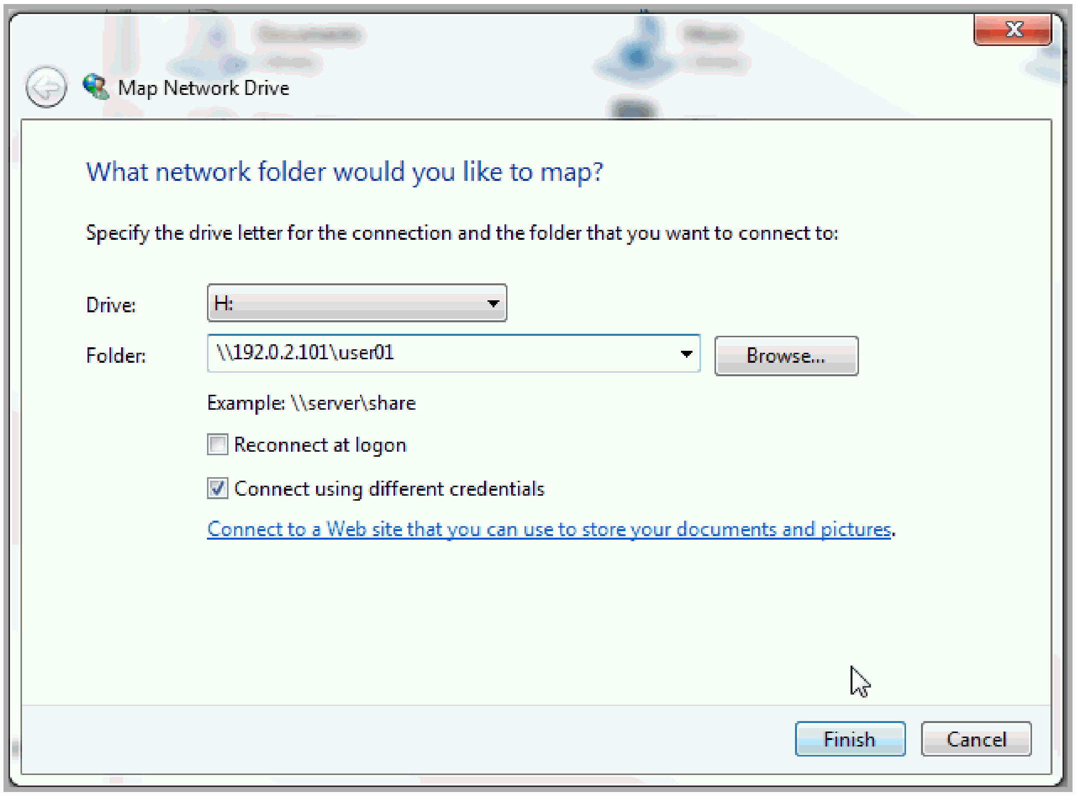
Tasks

1. Access `user01`'s home directory on the **host01** VM from a Windows machine.
   - In this task, you examine the steps to access the home directory for `user01`, residing on the **host01** VM. This home directory is offered as a network share through Samba services running on **host01**. You performed the same task previously, but you accessed the share from an Oracle Linux client.
   - The steps are identical to the steps that are needed to map any Windows network share.
   - You can use your Windows username if the Samba administrator has mapped your Windows domain username to a Samba Linux username on the Linux host providing the Samba services.
   - In this example, you use `user01` as the username, and provide the Samba password set up for this username.

   a. Launch the tool to map a network drive.

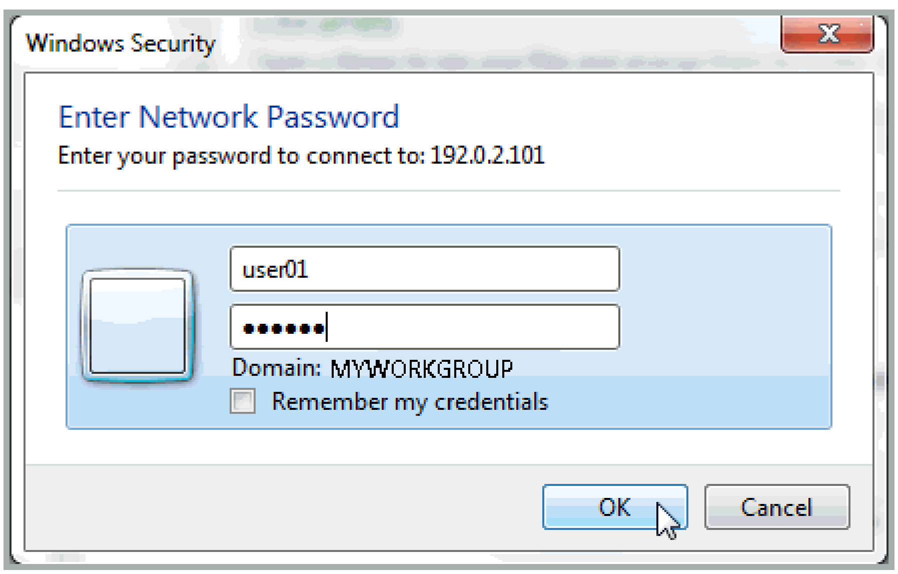


b. Provide the name of the share as \\<`server name`>\<`share name`>.

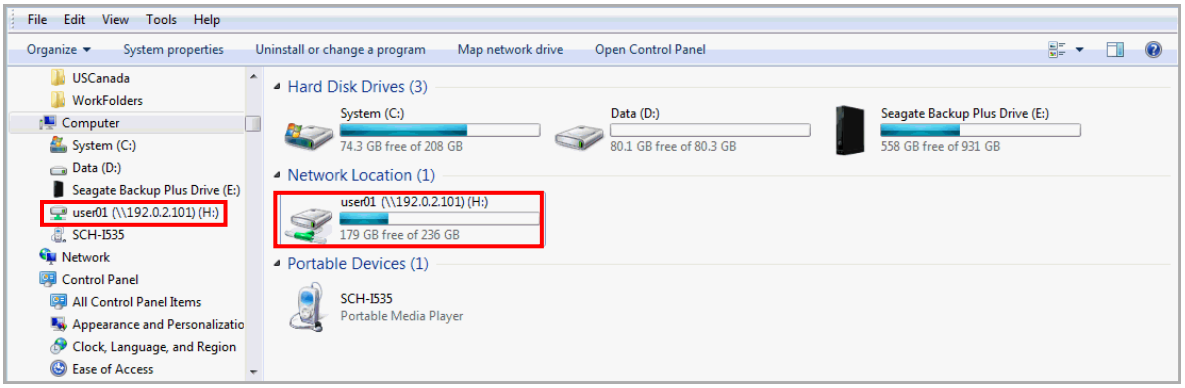


- Select “Connect using different credentials” to provide your Linux username and its associated Samba password.

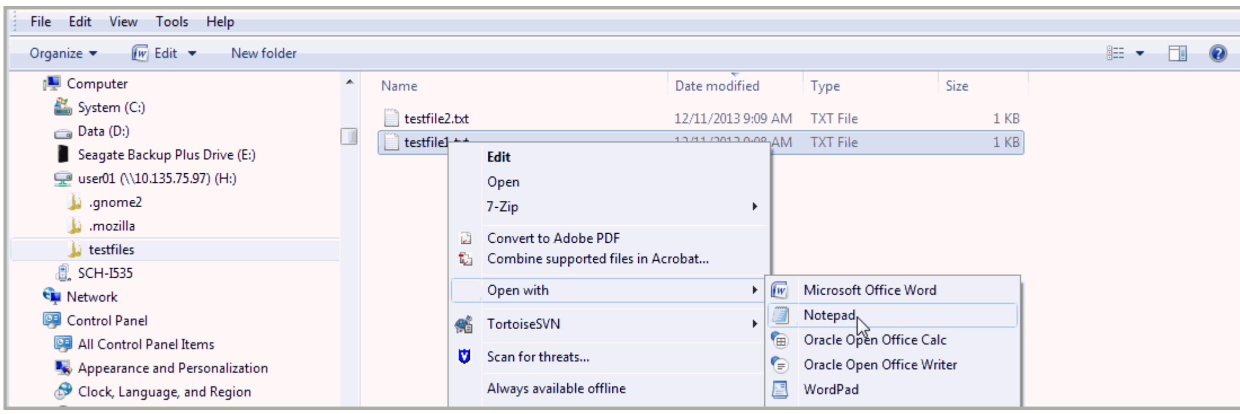
c. In the Windows Security window, enter the credentials for the share as `user01` and the Samba password as `MyOracle1`.



- Click OK to access and map the drive.
- After successful completion of the mapping operation, the home directory for `user01` on **host01** is mapped to drive `H:`.



d. You can view and manipulate the files in the `H:` drive:



e. Use Disconnect to release the network share.

Practices for Lesson 7: Advanced Software Package Management

Chapter 7

Practices for Lesson 7: Advanced Software Package Management

Practices Overview

In these practices, you:

- Learn to manage Yum plug-ins
- Create a binary RPM package
- Manage software updates with PackageKit's Software Update program
- Work with Yum history and Yum cache

Practice 7-1: Managing Yum Plug-Ins

Overview

In this practice, you do the following:

- Start the **host04** VM.
- View Public Yum Server configuration on **host04**.
- Use the `security` plug-in.
- Install and use the `downloadonly` plug-in.
- Use Yum utilities.

Assumptions

- You are the `root` user on **dom0**.
- You use **host04** exclusively in this practice.
- The **host04** VM is pre-configured to access Oracle's Public Yum Server.

Tasks

1. Start the **host04** VM.
   a. From **dom0**, run the `xm list` command to list the running VMs.

```
# xm list
Name                    ID    Mem  VCPUs      State      Time(s)
Domain-0                 0   2048      2      r-----       758.9
host01                   4   1536      1      -b----        37.4
host02                   5   1536      1      -b----        37.3
```

   - You were instructed to shut down **host03** at the end of Practice 6.
   - Therefore, in this example, only **host01** and **host02** VMs are running.

   b. If **host03** is running on your system, use the `xm shutdown` command to shut it down.

   - The available memory on **dom0** allows a maximum of only three VMs to be running.
   - Therefore, it is necessary to shut down one VM before starting the **host04** VM.

```
# xm shutdown –w host03
Domain host03 terminated
All domains terminated
```

   c. If the `xm shutdown` command is taking more than a few seconds to complete, use `CTRL-C` to kill the command and run the following `xm destroy` command.

```
# xm destroy host03
```

   d. Use the `cd` command to change to the `/OVS/running_pool/host04` directory.

```
# cd /OVS/running_pool/host04
```

   e. Run the `xm create` command to create the **host04** VM.

```
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host04 (id=...)
```

2. Log in to **host04**.
   a. Determine the VNC port number for **host04** by running the `xm list -l host04 | grep location` command.

```
# xm list -l host04 | grep location
                (location 0.0.0.0:5903)
                (location 3)
```

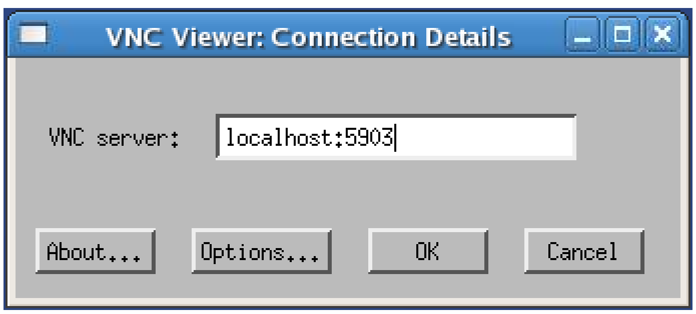
   - The sample shown indicates that the port number is **5903**. Your port number might be different.

   b. Run the `vncviewer&` command.

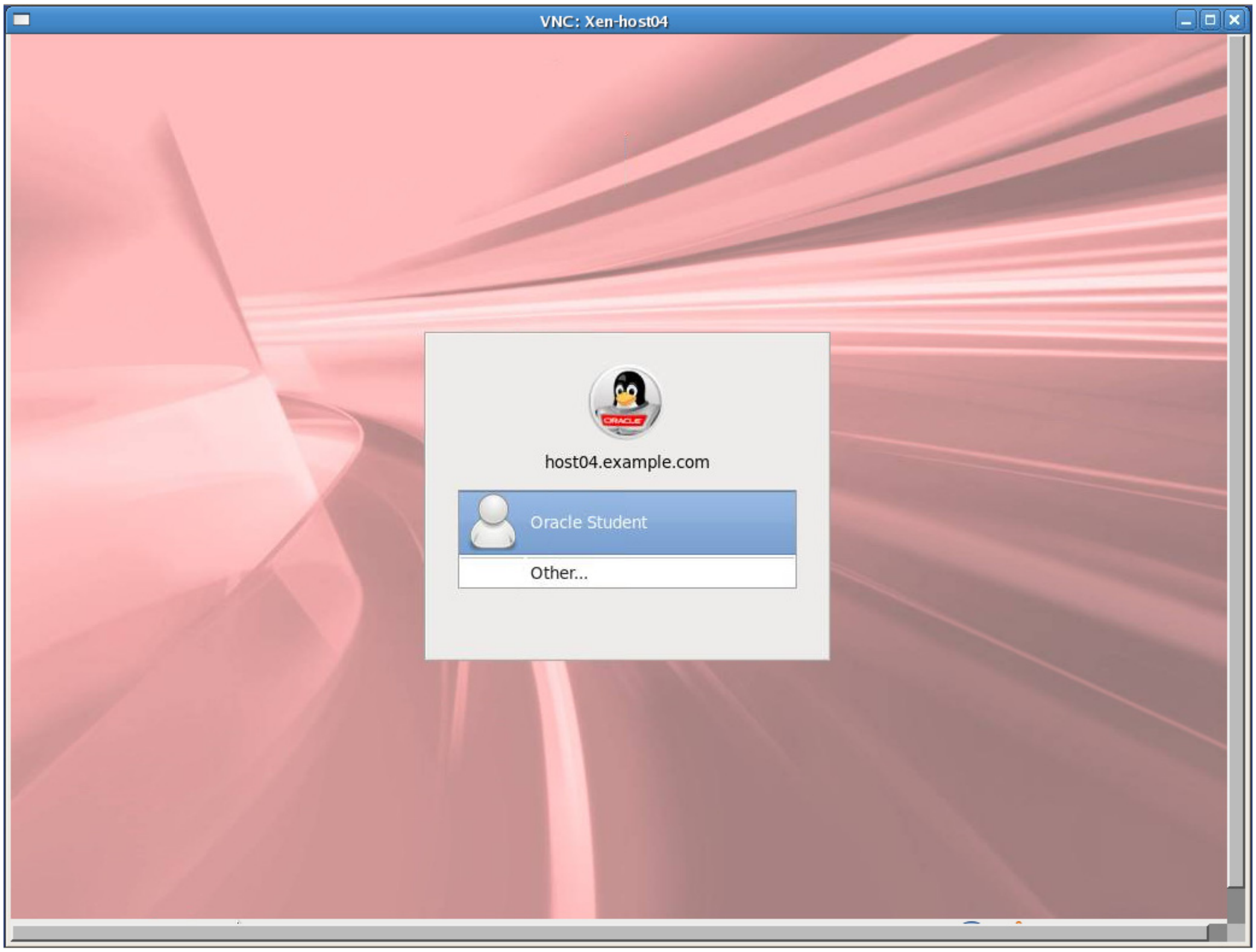
```
# vncviewer&
```

   - The **VNC Viewer: Connection Details** dialog box is displayed.

   c. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list -l host04 | grep location` command. For example, if the port number is **5903**, enter `localhost:5903` and click **OK**.



   - The GNOME login screen appears:

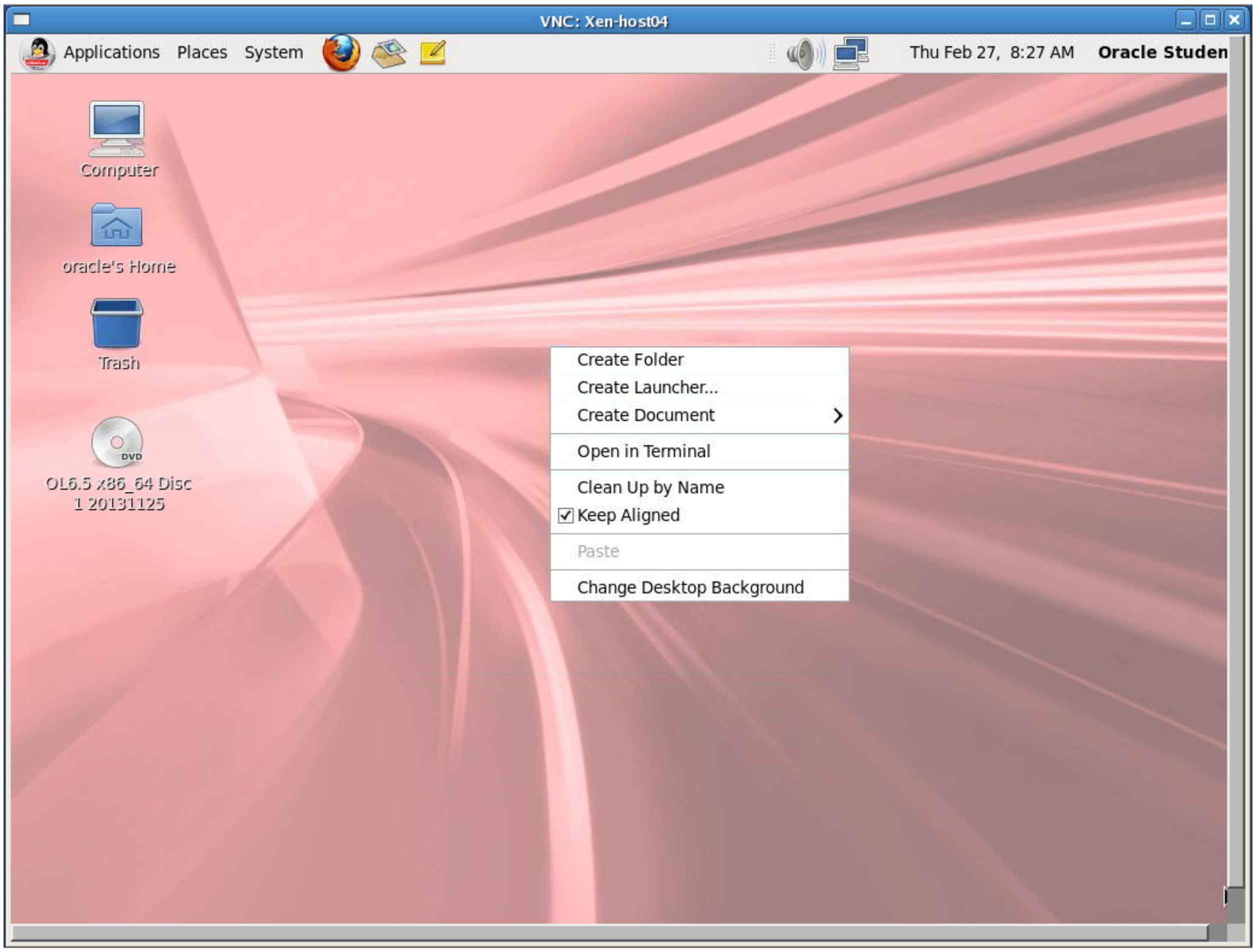


d. Click **Oracle Student** in the list of users. You are prompted for the **Password**.

e. Enter `oracle` for the Password and click **Log In**.

- The GNOME desktop appears.

f. Right-click on the desktop to display the pop-up menu:



g. From the pop-up menu, click **Open in Terminal**.
   - A terminal window appears.

h. In the terminal window, use the `su -` command to become the `root` user.
   - The `root` password is `oracle`.

```
$ su –
Password: oracle
#
```

3. View the Public Yum Server configuration on **host04**.
   - The **host04** VM is pre-configured to access Oracle's Public Yum Server.
   - There are four files that provide access to Public Yum:
     - `/etc/sysconfig/network-scripts/ifcfg-eth0`
     - `/etc/resolv.conf`
     - `/etc/profile`
     - `/etc/yum.repos.d/public-yum-ol6.repo`
   - The DNS and proxy configurations are specific to the Oracle University environment.

a. Use the `cat` command to view the contents of the `/etc/sysconfig/network-scripts/ifcfg-eth0` file.

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
...
DOMAIN="example.com us.oracle.com"
DNS1=192.0.2.1
DNS2=152.68.154.3
DNS3=10.216.106.3
DNS4=193.32.3.252
```

b. Use the `cat` command to view the contents of the `/etc/resolv.conf` file.

- The content of this file is automatically generated by **NetworkManager** from the `DOMAIN` and `DNS*` entries in the `ifcfg-eth0` file.

```
# vi /etc/resolv.conf
# Generated by NetworkManager
search us.oracle.com example.com
nameserver 192.0.2.1
nameserver 152.68.154.3
nameserver 10.216.106.3
# NOTE: the libc resolver may not support more than 3
nameservers.
# The nameservers listed below may not be recognized.
nameserver 193.32.3.252
```

c. Use the `tail` command to view the last five lines in the `/etc/profile` file.

- The HTTP proxy server variable is set in the last line of this file.

```
# tail /etc/profile
...
export http_proxy=http://ges-proxy.us.oracle.com:80
```

d. Use the `cat` command to view the `public-yum-ol6.repo` file in the `/etc/yum.repos.d` directory.

- The following three Public Yum repositories are enabled:
  - **public_ol6_latest** (`enabled=1`)
  - **public_ol6_UEK_latest** (`enabled=$uek`)
  - **public_ol6_UEKR3_latest** (`enabled=$uekr3`)

```
# cat /etc/yum.repos.d/public-yum-ol6.repo
[public_ol6_latest]
...
enabled=1
...
[public_ol6_UEK_latest]
...
enabled=$uek
```

```
...
[public_ol6_UEKR3_latest]
...
enabled=$uekr3
...
```

4. On **host04**, display currently available and enabled Yum plug-ins.
   - Sample output is provided throughout this practice. Your output might be different.

   a. Use a Yum command. For example, run `yum info yum`.
      - Many `yum` commands display the plug-ins; this is just one example.
      - The first `yum` command takes a few minutes to complete because the Public Yum repositories need to initialize. Subsequent `yum` commands do not require this initialization process.

```
# yum info yum
Loaded plugins: refresh-packagekit, security
public_ol6_UEKR3_latest ...
public_ol6_latest ...
public_ol6_latest/primary ...
public_ol6_latest ...
Installed Packages
Name        : yum
Arch        : noarch
Version     : 3.2.29
Release     : 40.0.1.el6
Size        : 4.5 M
Repo        : installed
From repo   : anaconda-OracleLinuxServer-201311252058.x86_64
Summary     : RPM package installer/updater/manager
URL         : http://yum.baseurl.org/
License     : GPLv2+
Description : Yum is a utility that can check for and ...

Available Packages
Name        : yum
Arch        : noarch
Version     : 3.2.29
Release     : 43.0.1.el6_5
Size        : 996 k
Repo        : public_ol6_latest
Summary     : RPM package installer/updater/manager
URL         : http://yum.baseurl.org/
License     : GPLv2+
Description : Yum is a utility that can check for and ...
```

- Each time you execute the `yum` command, the currently enabled Yum plug-ins are listed immediately, before the output of the `yum` command.
- In this example, the currently enabled Yum plug-ins are:
  - `refresh-packagekit`
  - `security`

b. Use the `rpm` and `grep` commands to list the Yum plug-ins currently installed.

```
# rpm -qa | grep plugin | grep yum
yum-rhn-plugin-0.9.1-49.0.1.el6.noarch
PackageKit-yum-plugin-0.5.8-21.0.1.el6.x86_64
yum-plugin-security-1.1.30-14.0.1.el6.noarch
```

- Most Yum plug-in names contain the `yum-plugin` string, but there are exceptions. For example, the `yum-rhn-plugin` does not follow this naming pattern. The `yum-rhn-plugin` is used to connect to the Red Hat Network (RHN).
- Notice that there are three plug-ins listed in the output of step 2b but that there are only two plug-ins showing as enabled when you run the `yum info yum` command in step 2a. This is because the `yum-rhn-plugin` plug-in is not enabled when running Oracle Linux.

5. Yum plug-ins have an associated configuration file. Access and examine these configuration files.

a. Use the `cd` command to change to the `/etc/yum/pluginconf.d` directory. Use the `ls` command to list the contents of the directory.

```
# cd /etc/yum/pluginconf.d
# ls -l
total 12
-rw-r--r--. ... refresh-packagekit.conf
-rw-r--r--. ... rhnplugin.conf
-rw-r--r--. ... security.conf
```

b. Use the `cat` command to view the contents of the `security.conf` file.

```
# cat security.conf
[main]
enabled=1
```

- The contents of a Yum plug-in configuration file vary from one plug-in to another.

6. Exercise the `security` plug-in.

a. Run the `yum updateinfo list` to list all the errata that are available for your system.

- Sample output is provided. New errata exist since this example was created.

```
# yum updateinfo list
Loaded plugins: refresh-packagekit, security
ELSA-2014-0043 Moderate/Sec.  bind-libs-32:9.8.2-0.23.rc1.el...
ELSA-2014-0043 Moderate/Sec.  bind-utils-32:9.8.2-0.23.rc1.e...
ELSA-2013-1866 Moderate/Sec.  ca-certificates-2013.1.95-65.1...
...
ELBA-2014-0138 bugfix         yum-utils-1.1.30-17.0.1.el6_5...
updateinfo list done
```

- This errata list provides the errata ID for each entry in the errata.
- Errata fall into three categories:
  - Bug fixes
  - Security fixes listed by priority (important, moderate, low)
  - Enhancements

b. Use the `cves` option with the `yum updateinfo list` command to display only the security patches.

- This list provides the CVE ID instead of the errata ID.

```
# yum updateinfo list cves
Loaded plugins: refresh-packagekit, security
 CVE-2014-0591 Moderate/Sec.  bind-libs-32:9.8.2-0.23.rc1...
 CVE-2014-0591 Moderate/Sec.  bind-utils-32:9.8.2-0.23.rc1...
 CVE-2012-4453 Moderate/Sec.  dracut-004-336.0.1.el6.noarch
...
 CVE-2013-6424 Important/Sec. xorg-x11-server-common-1.13...
updateinfo list done
```

c. Correlate a published CVE to its errata ID. The following example selects the last CVE in the previous output.

- Use the `--cve <CVE>` option to the `yum updateinfo list` command.

```
# yum updateinfo list --cve CVE-2013-6424
Loaded plugins: refresh-packagekit, security
ELSA-2013-1868 Important/Sec. xorg-x11-server-Xorg-1.13...
ELSA-2013-1868 Important/Sec. xorg-x11-server-common-1.13...
updateinfo list done
```

- The list for this CVE includes the security patches by errata ID for the particular CVE ID. This CVE affects two packages in this example.
- Your output differs if you choose a different CVE.

d. Display additional information about a specific CVE.

- Use the `info` argument instead of the `list` argument.

```
# yum updateinfo info --cve CVE-2013-6424
Loaded plugins: refresh-packagekit, security
==============================================================
   xorg-x11-server security update
==============================================================
  Update ID : ELSA-2013-1868
    Release : Oracle Linux 6
       Type : security
     Status : final
     Issued : 2013-12-20
       CVEs : CVE-2013-6424
Description : [1.13.0-23.1]
            : - Fix root window damage reports when Xinerama
```

```
             :   is active (#919165)
   Severity : Important
updateinfo info done
```

- Your output differs if you choose a different CVE.

e. Update the packages affected by the specific CVE.
   - Answer “`y`” when prompted “**Is this ok [y/N]:**”.
   - You are asked about the GPG key only the first time you use the `yum update` (or `yum install`) command.

```
# yum update --cve CVE-2013-6424
Loaded plugins: refresh-packagekit, security
Setting up Update Process
Resolving Dependencies
...
Transaction Summary
==============================================================
Upgrade       2 Package(s)
Total download size: 1.3 M
Is this ok [y/N]: y
...
Retrieving key from http://192.0.2.1/repo/OracleLinux/OL6/5/...
...
Importing GPG key ...
...
Is this ok [y/N]: y
...
Complete!
```

7. Install the `downloadonly` Yum plug-in.

a. Get information about the `downloadonly` plug-in.

```
# yum info yum-plugin-downloadonly
Loaded plugins: refresh-packagekit, security
Available Packages
Name        : yum-plugin-downloadonly
Arch        : noarch
Version     : 1.1.30
Release     : 17.0.1.el6_5
Size        : 20 k
Repo        : public_ol6_latest
Summary     : Yum plugin to add downloadonly command option
URL         : http://yum.baseurl.org/download/yum-utils/
License     : GPLv2+
Description : This plugin adds a --downloadonly flag to yum ...
```

- The plug-in information specifies that this plug-in introduces a new flag, `--downloadonly`, for the `yum` command that allows you to download a package without installing it.
- Note that the package shows as available, not installed.

b. Install the plug-in by using the `yum` command.

```
# yum install yum-plugin-downloadonly
Loaded plugins: refresh-packagekit, security
Setting up Install Process
Resolving Dependencies
...
Transaction Summary
=======================================================================
Install       1 Package(s)
Total download size: 20 k
Installed size: 21 k
Is this ok [y/N]: y
...
Complete!
```

8. Use the newly installed `downloadonly` Yum plug-in.

a. List the Oracle Database pre-installation packages (`oracle-rdbms`) that are available for installation.

```
# yum list available | grep oracle-rdbms
oracle-rdbms-server-11gR2-preinstall.x86_64
oracle-rdbms-server-12cR1-preinstall.x86_64
```

b. View more information for the Oracle Database pre-installation packages.

```
# yum info oracle-rdbms*
Loaded plugins: downloadonly, refresh-packagekit, security
Available Packages
Name        : oracle-rdbms-server-11gR2-preinstall
Arch        : x86_64
Version     : 1.0
Release     : 9.el6
Size        : 17 k
Repo        : public_ol6_latest
Summary     : Sets the system for Oracle single instance and ...
License     : GPLv2
Description : This package installs software packages and ...

Name        : oracle-rdbms-server-12cR1-preinstall
Arch        : x86_64
Version     : 1.0
Release     : 11.el6
Size        : 15 k
```

```
Repo        : public_ol6_latest
Summary     : Sets the system for Oracle single instance and ...
License     : GPLv2
Description : This package installs software packages and ...
```

- In this example, there are two releases of this package. You select to download the latest release of the package, which, in this example, is `oracle-rdbms-server-12cR1-preinstall`.
- Be careful when using wildcards with the `yum` command. They are very useful to list packages, but you can get unexpected results when using wildcards to install or remove packages.

c. Check the dependencies for the target package by using the `repoquery` command.

- The `repoquery` utility is part of the `yum-utils` package and is useful for querying information from Yum repositories.
- The `--requires` option lists package dependencies.

```
# repoquery --requires oracle-rdbms-server-12cR1-preinstall
/bin/bash
/bin/sh
/etc/redhat-release
bc
bind-utils
...
xorg-x11-xauth
```

- If a dependency package is missing, it is downloaded along with the `oracle-rdbms-server-12cR1-preinstall` package in the next step.

d. Use the new flag introduced by the `downloadonly` Yum plug-in to download the `oracle-rdbms-server-12cR1-preinstall` package and any missing dependent packages.

```
# yum install oracle-rdbms-server-12cR1-preinstall --
downloadonly
Loaded plugins: downloadonly, refresh-packagekit, security
Setting up Install Process
Resolving Dependencies
...
Transaction Summary
==============================================================
Install       7 Package(s)

Total download size: 7.3 M
Installed size: 23 M
Is this ok [y/N]: y
Downloading Packages:
...
exiting because --downloadonly specified
```

- In this example, six packages are downloaded in addition to the `oracle-rdbms-server-12cR1-preinstall-1.0-11.el6.x86_64.rpm` package.

e. Verify that the package and its dependency packages are downloaded by examining the content of the `/var/cache/yum/x86_64/6Server/public_ol6_latest/packages` directory.

```
# cd /var/cache/yum/x86_64/6Server/public_ol6_latest/packages
# ls
compat-libcap1-1.10-1.x86_64.rpm
compat-libstdc++-33-3.2.3-69.el6.x86_64.rpm
gcc-c++-4.4.7-4.el6.x86_64.rpm
ksh-20120801-10.el6.x86_64.rpm
libaio-devel-0.3.107-10.el6.x86_64.rpm
libstdc++-devel-4.4.7-4.el6.x86_64.rpm
oracle-rdbms-server-12cR1-preinstall-1.0-11.el6.x86_64.rpm
```

- You can also specify an alternative directory for the downloaded packages with `--downloaddir=`<*directory path*>.
- If the package that you want to download is already installed, it is not downloaded and its dependencies are not downloaded. In the next step, you use a different technique to download a package if the package is already installed on your system.

9. Using the Yum utilities.

- In this task, you examine the Yum utilities available and use the `yumdownloader` utility to download a package.

a. Use the `rpm -ql` command to examine the files that make up the `yum-utils` package.

```
# rpm -ql yum-utils
...
/usr/bin/repoquery
...
/usr/bin/yumdownloader
...
```

- Notice that `yumdownloader` and `repoquery` are included in the `yum-utils` package.

b. Use the `--downloadonly` option of the `downloadonly` plug-in to attempt to download the `xorg-x11-server-Xorg` program.

```
# yum install xorg-x11-server-Xorg --downloadonly
Loaded plugins: downloadonly, refresh-packagekit, security
Setting up Install Process
Package xorg-x11-server-Xorg-1.13.0-23.1.el6_5.x86_64 already
installed and latest version
Nothing to do
```

- The package is not downloaded because it is already installed.

c. Use the `yumdownloader` command to download the `xorg-x11-server-Xorg` package.

```
# yumdownloader xorg-x11-server-Xorg
Loaded plugins: refresh-packagekit
xorg-x11-server-Xorg-1.13.0-23.1.el6_5.x86_64.rpm    | ...
```

- The command downloads the package in the current directory.
- The command does not download the dependencies for the `xorg-x11-server-Xorg` program.

d. Use the `yum deplist` command to display the dependencies for the `xorg-x11-server-Xorg` program:

```
# yum deplist xorg-x11-server-Xorg
Loaded plugins: downloadonly, refresh-packagekit, security
Finding dependencies:
package: xorg-x11-server-Xorg.x86_64 1.10.4-6.el6
  dependency: libdbus-1.so.3()(64bit)
   provider: dbus-libs.x86_64 1:1.2.24-5.el6_1
   provider: dbus-libs.x86_64 1:1.2.24-7.0.1.el6_3
...
  dependency: libc.so.6(GLIBC_2.3.4)(64bit)
   provider: glibc.x86_64 2.12-1.80.el6_3.7
   provider: glibc.x86_64 2.12-1.7.el6_0.5
...
  dependency: libdrm.so.2()(64bit)
   provider: libdrm.x86_64 2.4.23-1.el6
   provider: libdrm.x86_64 2.4.25-2.el6
...
```

- If you download a package by using the `yumdownloader` utility, you have to determine the dependencies manually. You can use the `rpm` command to let you know which packages are missing and install those packages.
- A dependency package is different than a dependent package. When you use the `yum deplist <package name>` command, you list the packages that the `<package name>` package needs to operate.
- A dependent package is a package that needs the `<package name>` package to operate. Knowing whether a package is dependent is important when trying to remove a package. By default, the `yum` command and the `rpm` command do not allow you to remove a package that is needed by other packages. To find out which packages depend on a package, use the `repoquery --whatrequires <package name>` command.

e. Use the `repoquery --whatrequires` command for the `xorg-x11-server-Xorg` program to find out which packages depend on `xorg-x11-server-Xorg`.

- This command takes a few minutes to run.

```
# repoquery --whatrequires xorg-x11-server-Xorg
xorg-x11-server-Xorg-0:1.13.0-23.1.el6_5.x86_64
tigervnc-server-module-0:1.0.90-0.10.20100115svn3945.el6.x86_64
tigervnc-server-module-0:1.0.90-0.15.20110314svn4359.el6.x86_64
...
```

- Compare this list to the list obtained with the `yum deplist` command in step 9d.

Practice 7-2: Creating an RPM Package

Overview

In this practice, you prepare to build an RPM package. The steps for this preparation are:

- Create a non-privileged user to perform the build.
- Check for the required packages to perform the build and install them if necessary.
- Create the directory infrastructure for the build.
- Create the program for the package.
- Create the compressed TAR file and store it in the appropriate build directory.
- Create the spec file.

After performing the steps to prepare for the RPM package build, you perform the build by using the `rpmbuild` command.

In the last task, you install the new RPM package as `root` to verify that the program gets installed as you expected.

Assumptions

You are the `root` user on **host04**.

Tasks

1. Create a non-privileged user `rpmbuilder` to perform the build.
   a. Use the `useradd` command to add the `rpmbuild` user.

```
# useradd -d /home/rpmbuilder -m rpmbuilder
```

   b. Use the `ls –ld` command to view the home directory for the `rpmbuilder` user.

```
# ls -ld /home/rpmbuilder
drwx------. 4 rpmbuilder rpmbuilder ... /home/rpmbuilder
```

   c. Use the `passwd` command to create a password of `oracle` for the `rpmbuilder` user.
   - Ignore the BAD PASSWORD warnings.

```
# passwd rpmbuilder
Changing password for user rpmbuilder.
New password: oracle
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password: oracle
passwd: all authentication tokens updated successfully.
```

2. Verify the presence of the required `rpmdevtools` package and install it if it is not installed.
   a. Run the `rpm` command to search for the `rmpdevtools` command.

```
# rpm -qa | grep rpmdevtools
```

   - In this example, the `rpmdevtools` package is not installed.

   b. Use the `yum` command to install the `rpmdevtools` package.

```
# yum install rpmdevtools
...
```

```
Transaction Summary
==============================================================
Install       4 Package(s)

Total download size: 330 k
Installed size: 779 k
Is this ok [y/N]: y
...
Complete!
```

- The `rpm-build` package is a dependency for the `rpmdevtools` package and is installed at the same time as `rpmdevtools`. The `rpm-build` package contains the `rpmbuild` command, which you use to build the RPM package in this practice. The `rpmdevtools` package contains several commands that are useful when creating RPM packages, including the following two commands that you use later in this practice:
  - `rpmdev-setuptree`: Creates the build directory structure
  - `rpmdev-newspec`: Creates a skeleton `spec` file

3. Create the directory infrastructure for the RPM build.
   a. Use the `su -` command to become the `rpmbuilder` user.
      - Use the `whoami` command to confirm you are the `rpmbuilder` user.

```
# su – rpmbuilder
$ whoami
rpmbuilder
```

   b. Use the `ls -la` command to list the contents of the `rpmbuilder` user's home directory.

```
$ ls -la
...
-rw-r--r--. ... .bash_logout
-rw-r--r--. ... .bash_profile
-rw-r--r--. ... .bashrc
drwxr-xr-x. ... .gnome2
drwxr-xr-x. ... .mozilla
```

   c. Run the `rpmdev-setuptree` command, and then use the `ls -la` command to verify the presence of new entries in the home directory.

```
$ rpmdev-setuptree
$ ls -la
...
-rw-r--r--. ... .bash_logout
-rw-r--r--. ... .bash_profile
-rw-r--r--. ... .bashrc
drwxr-xr-x. ... .gnome2
drwxr-xr-x. ... .mozilla
drwxrwxr-x. ... rpmbuild
-rw-rw-r--. ... .rpmmacros
```

- Notice the new `rpmbuild` directory and the new `.rpmmacros` file.

d. Use the `ls -lR` command to view the directory structure in the new `rpmbuild` directory.

```
$ ls -lR rpmbuild
...
drwxrwxr-x. ... BUILD
drwxrwxr-x. ... RPMS
drwxrwxr-x. ... SOURCES
drwxrwxr-x. ... SPECS
drwxrwxr-x. ... SRPMS
...
```

4. Create the program that is going to be part of the RPM package

a. Use the `cd` command to change to the `rpmbuild` directory.

```
$ cd rpmbuild
```

b. Use the `vi` editor to create the following `hello.c` file.

```
$ vi hello.c
#include <stdio.h>

main() {
   printf("Hello World!\n");
   return(0);
}
```

c. Use the `gcc` command to compile the program.

- Name the output file `hello`.

```
$ gcc hello.c -o hello
```

d. Run the `hello` program.

```
$ ./hello
Hello World!
```

5. Create the compressed TAR file with the build directory structure and the compiled program, and store it in the `rpmbuild/SOURCES` directory.

- The build directory name must reflect the correct name and version for the package that you are building.

a. Use the `pwd` command to ensure you are in the `/home/rpmbuilder/rpmbuild` directory.

- From this directory, use the `mkdir` command to create the `hello-1.0` directory.
- Use the `mv` command to move the `hello` program to the new directory.

```
$ pwd
/home/rpmbuilder/rpmbuild
$ mkdir hello-1.0
$ mv hello hello-1.0/
```

b. Use the `tar` command to create a compressed TAR file of the `hello-1.0` directory structure and store the resulting `.tar.gz` file in the `rpmbuild/SOURCES` directory.

```
$ tar cvzf SOURCES/hello-1.0.tar.gz hello-1.0/
hello-1.0/
hello-1.0/hello
```

c. Use the `ls` command to verify that the new `.tar.gz` file is in the `SOURCES` directory.

```
$ ls SOURCES
hello-1.0.tar.gz
```

6. Create and populate the `spec` file.

a. From the `rpmbuild` directory, use the `rpmdev-newspec` to create a skeleton `spec` file.

```
$ rpmdev-newspec SPECS/hello.spec
Skeleton specfile (minimal) has been created to
"SPECS/hello.spec".
```

b. Use the `cat` command to view the contents of the new `spec` file.

```
$ cat SPECS/hello.spec
Name:           hello
Version:
Release:        1%{?dist}
Summary:
...
%changelog
```

c. Use the `cd` command to change to the `SPECS` directory.

```
$ cd SPECS
```

d. Use the `vi` editor to edit the `hello.spec` file and populate the header section by making the following changes:

Note: A pre-configured `hello.spec` file exists on **dom0** in the `/OVS/seed_pool/sfws` directory.

- You can edit the `hello.spec` file as follows using the `vi` command, or you can use the `sftp` command (as the `root` user) and copy `/OVS/seed_pool/sfws/hello.spec` from **dom0** to `/home/rpmbuilder/rpmbuild/SPECS/hello.spec` on **host04**.
- If you use this `hello.spec` file on **dom0**, you do not need to edit the file as instructed in the following steps. You can go immediately to step 6k.
  - Leave `hello` as the `Name` tag.
  - Specify `1.0` for the `Version` tag.
  - Leave the `Release` information as is.
  - Specify `Test for the hello program` for the `Summary` tag.
  - Specify `Applications/Communications` for the `Group` tag. In this example, the group selection is a subjective choice. See the list of available groups in this file: `/usr/share/doc/rpm-4.8.0/GROUPS`.

- Specify GPL for the License tag.
- Comment out the URL tag by inserting # at the beginning of the line.
- Specify hello-1.0.tar.gz for the Source0 tag.
- Comment out the BuildRequires and Requires tags.
- Add this line: A program that display Hello World as a new line following the %description directive.
- After making the changes, the header section looks like this:

```
Name:           hello
Version:        1.0
Release:        1%{?dist}
Summary:        Test for the hello program

Group:          Applications/Communications
License:        GPL
#URL:
Source0:        hello-1.0.tar.gz

#BuildRequires:
#Requires:

%description
A program that displays Hello World
```

e. Leave the %prep section as is.
   - The %prep macro is a section where you get the files ready for the build section. This might involve patching some files. The %setup macro in this section unpacks the source files in the SOURCES directory into the BUILD directory. The -q option indicates a quiet action.
   - In this example, the only necessary step for this section is the unpacking step.

f. Use the vi editor to remove the entries in the %build section but leave the %build macro.

```
%build
%configure                                    delete this line
make %{?_smp_mflags}                          delete this line
```

   - Generally, this section contains the steps to build the software. A command such as the make command is allowed. In this example, the software is already built.

g. Use the vi editor to make the following changes to the %install section of the hello.spec file:
   - Leave the rm -rf $RPM_BUILD_ROOT line as is. This line cleans the BUILDROOT directory before performing the build.
   - Comment out the make install DESTDIR=$RPM_BUILD_ROOT line. The next line creates the required directory.

- Add a line to create the build directory structure in the BUILDROOT directory by using the `install -d` command. This line is followed by an `install` command that copies the built program into its build directory.
- After making the changes, the `%install` section looks like this:

```
%install
rm -rf $RPM_BUILD_ROOT
#make install DESTDIR=$RPM_BUILD_ROOT
install -d $RPM_BUILD_ROOT/usr/local/bin
install hello $RPM_BUILD_ROOT/usr/local/bin/hello
```

- As seen in this example, this section "installs" the software, which means that the necessary directories are created and the package files are copied to their respective directory.

h. The `%clean` section remains unchanged.

i. Use the `vi` editor to make the following changes to the `%files` section:

- Change the `%doc` line to `/usr/local/bin/hello`.
- After making the changes, the `%files` section looks like this:

```
%files
%defattr(-,root,root,-)
/usr/local/bin/hello
```

- In the `%files` section, you list the files and their location for the binary RPM package. This section can also trigger the creation of directories.
- The `%defattr` directive sets the default attributes for the files listed in this section. It is a pattern of the form (`mode`, `owner`, `group`). A `-` means "default to the installed values." In this example, the ownership is set to `root` for both the owner and group settings.

j. Leave the `%changelog` section unchanged.

k. Use the `cat` command to view the `hello.spec` file. Ensure that the contents of the `hello.spec` file match the following.

- Edit the file again if necessary to ensure the contents of `hello.spec` looks like this:

```
$ cat hello.spec
Name:           hello
Version:        1.0
Release:        1%{?dist}
Summary:        Test for the hello program

Group:          Applications/Communications
License:        GPL
#URL:
Source0:        hello-1.0.tar.gz

#BuildRequires:
#Requires:
```

```
%description
A program that displays Hello World

%prep
%setup -q


%build


%install
rm -rf $RPM_BUILD_ROOT
#make install DESTDIR=$RPM_BUILD_ROOT
install -d $RPM_BUILD_ROOT/usr/local/bin
install hello $RPM_BUILD_ROOT/usr/local/bin/hello

%clean
rm -rf $RPM_BUILD_ROOT


%files
%defattr(-,root,root,-)
/usr/local/bin/hello


%changelog
```

7. Perform the build of the binary RPM package.
   a. Use the `cd` command to change to the `/home/rpmbuilder/rpmbuild` directory.

```
$ cd /home/rpmbuilder/rpmbuild
```

   b. Run the `rpmbuild` command, specifying the following options and `spec` file parameter: `rpmbuild -bb -v SPECS/hello.spec`
      - The `-bb` option indicates that you want to build only the binary package.
      - The `-v` option requests verbose information.
      - The `SPECS/hello.spec` parameter specifies the location of the `spec` file for this RPM binary build.
      - The four major sections during the build process, `%prep`, `%build,` `%install`, and `%clean`, are shown in bold format in this example.

```
$ rpmbuild -bb -v SPECS/hello.spec
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
```

```
+ cd /home/rpmbuilder/rpmbuild/BUILD
+ rm -rf hello-1.0
+ /bin/tar -xf -
+ /usr/bin/gzip -dc /home/rpmbuilder/rpmbuild/SOURCES/hello-
1.0.tar.gz
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd hello-1.0
+ /bin/chmod -Rf a+rX,u+w,g-w,o-w .
+ exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
+ cd hello-1.0
+ exit 0
Executing(%install): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
+ cd hello-1.0
+ rm -rf /home/rpmbuilder/rpmbuild/BUILDROOT/hello-1.0-
1.el6.x86_64
+ install -d /home/rpmbuilder/rpmbuild/BUILDROOT/hello-1.0-
1.el6.x86_64/usr/local/bin
+ install hello /home/rpmbuilder/rpmbuild/BUILDROOT/hello-1.0-
1.el6.x86_64/usr/local/bin/hello
+ /usr/lib/rpm/check-rpaths /usr/lib/rpm/check-buildroot
+ /usr/lib/rpm/brp-compress
+ /usr/lib/rpm/brp-strip
+ /usr/lib/rpm/brp-strip-static-archive
+ /usr/lib/rpm/brp-strip-comment-note
Processing files: hello-1.0-1.el6.x86_64
Requires(rpmlib): rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
Requires: libc.so.6()(64bit) libc.so.6(GLIBC_2.2.5)(64bit)
rtld(GNU_HASH)
Checking for unpackaged file(s): /usr/lib/rpm/check-files
/home/rpmbuilder/rpmbuild/BUILDROOT/hello-1.0-1.el6.x86_64
Wrote: /home/rpmbuilder/rpmbuild/RPMS/x86_64/hello-1.0-
1.el6.x86_64.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
+ cd hello-1.0
```

```
+ rm -rf /home/rpmbuilder/rpmbuild/BUILDROOT/hello-1.0-
1.el6.x86_64
+ exit 0
```

- If you see a "`warning: Could not canonicalize hostname:`" message, this can be ignored. This is a DNS resolution error and can be fixed by adding the host name to `/etc/hosts`.

c. View the new RPM package in the RPMS directory.

```
$ cd RPMS
$ ls
x86_64
$ cd x86_64
$ ls
hello-1.0-1.el6.x86_64.rpm
```

- The package appears with the version and release specified in the `hello.spec` file.

8. Install the newly built package.

a. Use the `exit` command to log off as the `rpmbuilder` user.

- Use the `whoami` command to verify you are the `root` user.

```
$ exit
logout
# whoami
root
```

b. Use the `cd` command to change to the directory where the new package resides.

```
# cd /home/rpmbuilder/rpmbuild/RPMS/x86_64
# ls
hello-1.0-1.el6.x86_64.rpm
```

c. Use the `rpm` command to install the package:

```
# rpm -ivh hello-1.0-1.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:hello                  ########################################### [100%]
```

d. Run the `which hello` command to display the path of the command.

```
# which hello
/usr/local/bin/hello
```

e. Run the `hello` program.

```
# hello
Hello World!
```

f. Use the `ls -l` command to display the file and its permissions in its target directory.

```
# ls -l /usr/local/bin
total 8
-rwxr-xr-x. 1 root root ... hello
```

- The permissions are those specified by the `%defattr` directive in the `spec` file.

Practice 7-3: Managing Software Updates with PackageKit

Overview

PackageKit is a software program that provides graphical tools to install software and software updates on your Linux systems. PackageKit is available for several Linux distributions.

In this practice you use the Software Update program that is part of PackageKit to manage software updates on your Oracle Linux system.

PackageKit also includes the Add/Remove Software graphical tool, but this program is not used in this practice.

Assumptions

- You are the `root` user on **host04**.
- Ensure that you logged in to **host04** using `vncviewer` and not `ssh`.

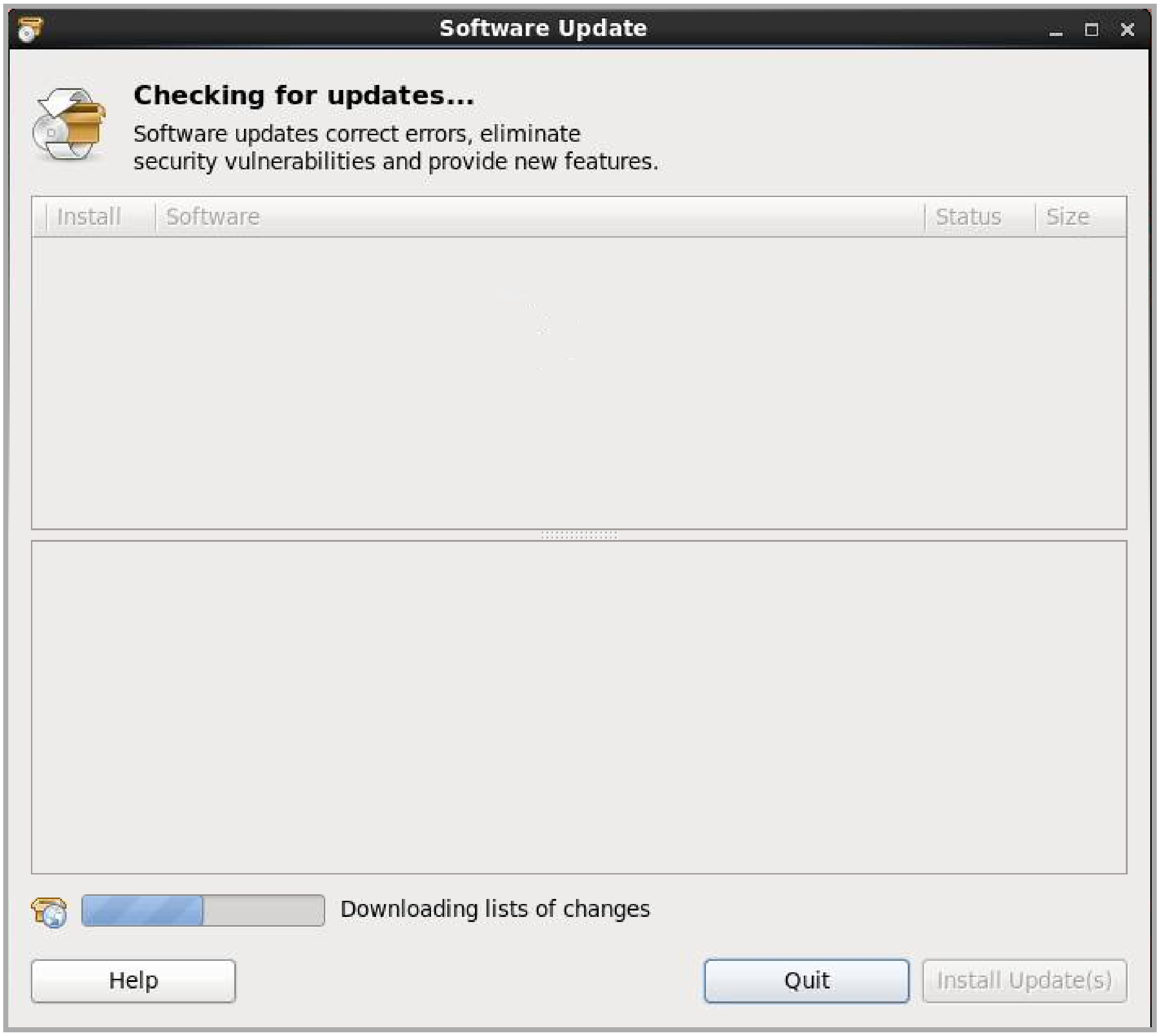
Tasks

1. When using the PackageKit Software Update program, the proxy set through an environment variable does not work. You need to set the proxy directly in the Yum configuration file.
   a. As the `root` user on **host04**, use the `vi` editor to edit the `/etc/yum.conf` file and add the following line.

```
# vi /etc/yum.conf
proxy=http://ges-proxy.us.oracle.com:80
```

2. Launch Software Update.
   a. In the GNOME task bar, select **System > Administration > Software Update**.

- The Software Update window appears.



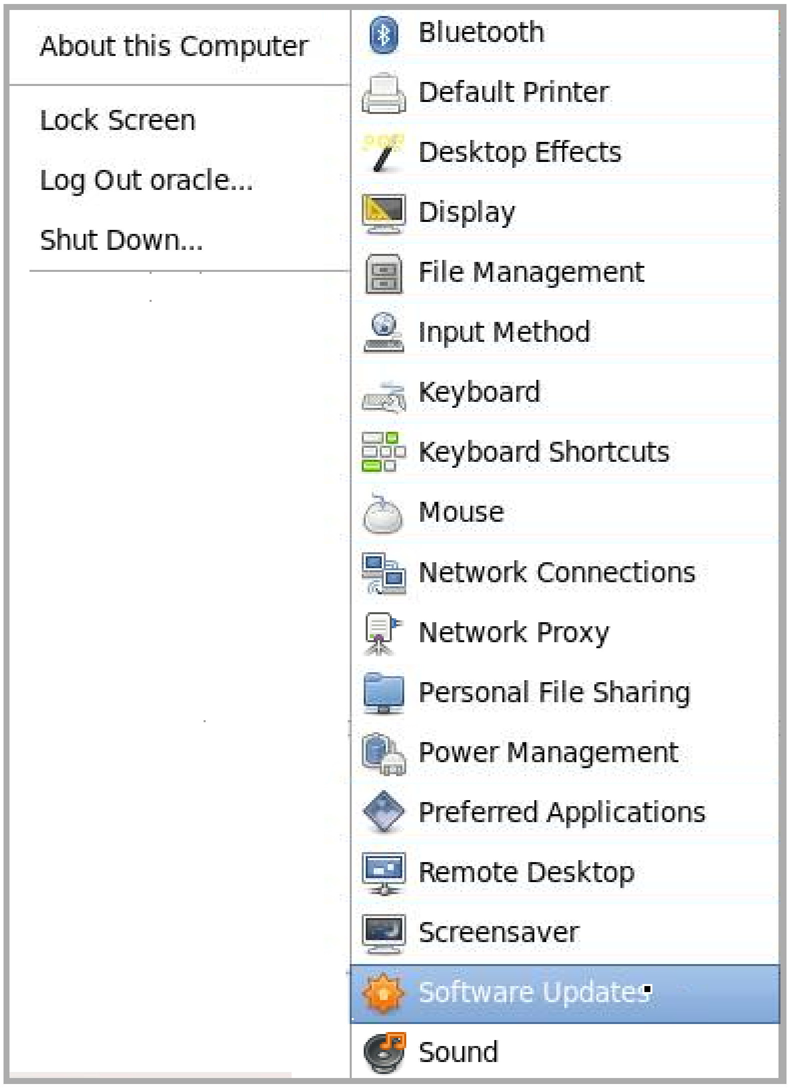
- "Checking for updates" takes several minutes to complete.
- Continue with step 2b and step 3 while waiting for the update to complete.

b. While the list of changes is being created, open a terminal window on the desktop, and examine the process that is running to obtain the lists of updates, called changes in the Software Update program:

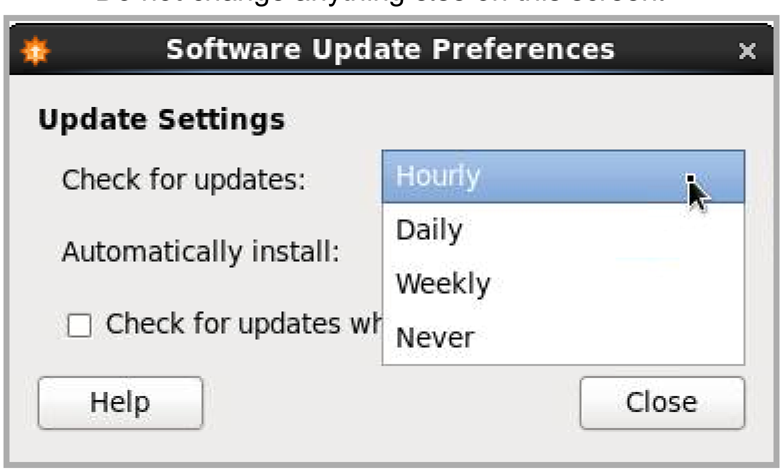
```
$ ps -ef | grep yum
root ... /usr/bin/python
/usr/share/PackageKit/helpers/yum/yumBackend.py get-updates
newest
...
```

- The `yumBackend.py get-updates` is the PackageKit program that checks for updates.

3. Change the frequency at which the Software Update program checks for updates.
   a. Select **System > Preferences > Software Updates** from the task bar.

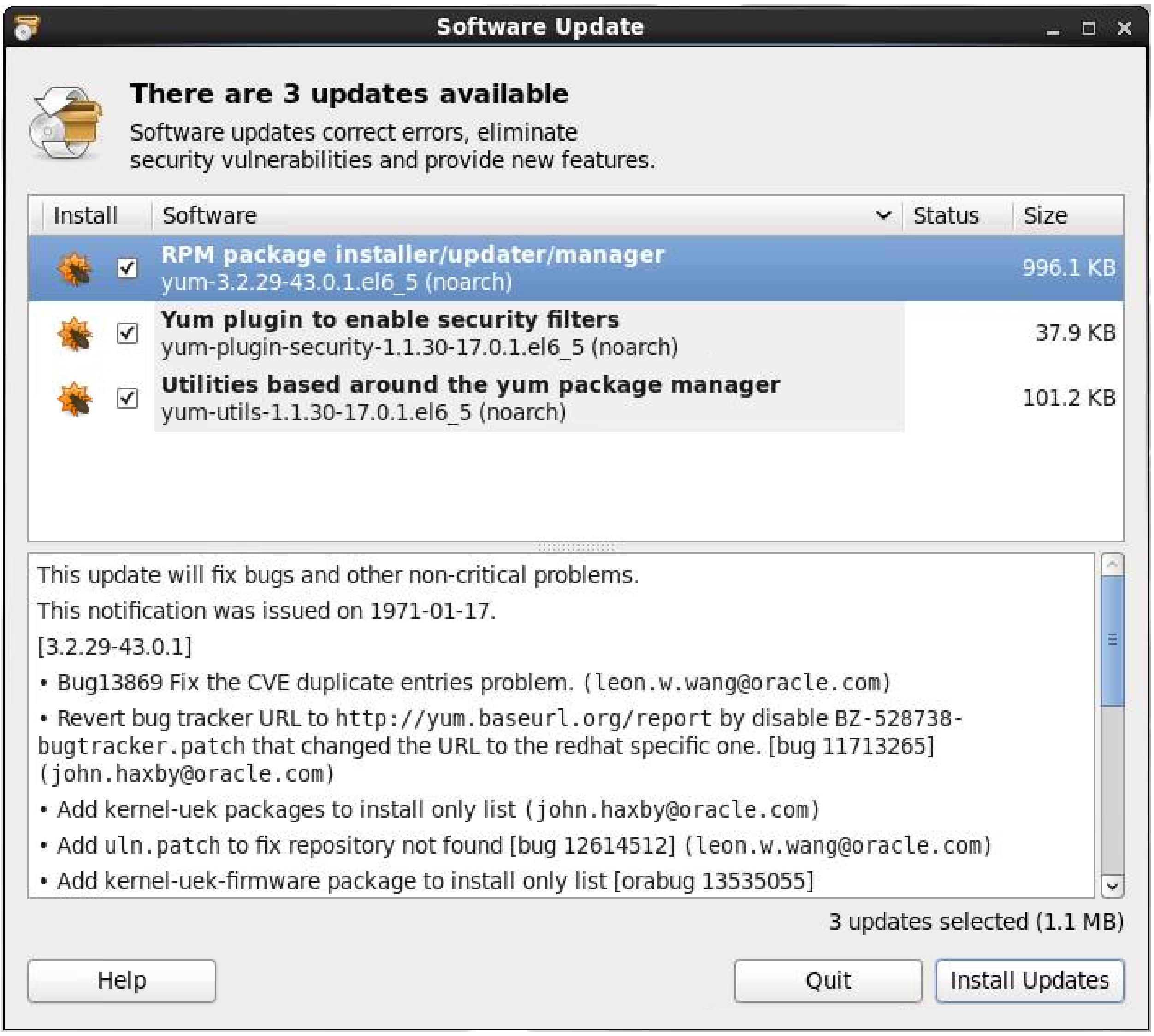


   b. From the "**Check for updates**" drop-down list, select **Hourly**.
      - Do not change anything else on this screen.



      - Click **Close** to trigger the change.

4. Install the update(s) flagged by the Software Update program.
   a. Return to the Software Update program. In this example, the program has found three updates.



- This is sample output. Your environment might be different because updates have been added since this example was captured.
- In this first pass, the Software Update program looks for updates to the `rpm`, `yum`, and PackageKit programs. If it finds updates, you are prompted to install those before proceeding.

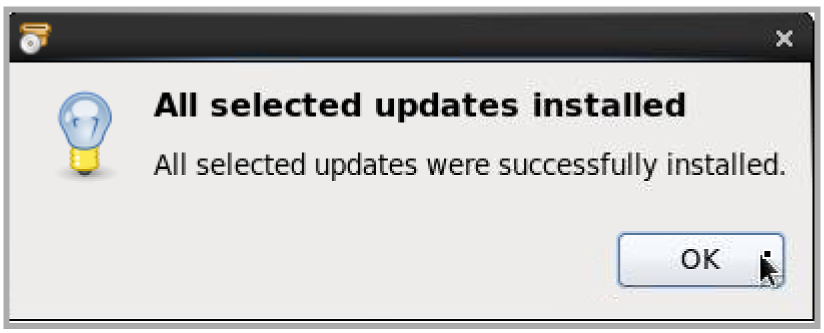
   b. Click the **Install Updates** button to install the updates to the `rpm`, `yum`, or PackageKit programs.

- The status area is located above the Help button. The “Resolving dependencies” message appears, followed by a window requesting authentication for the `root` user.

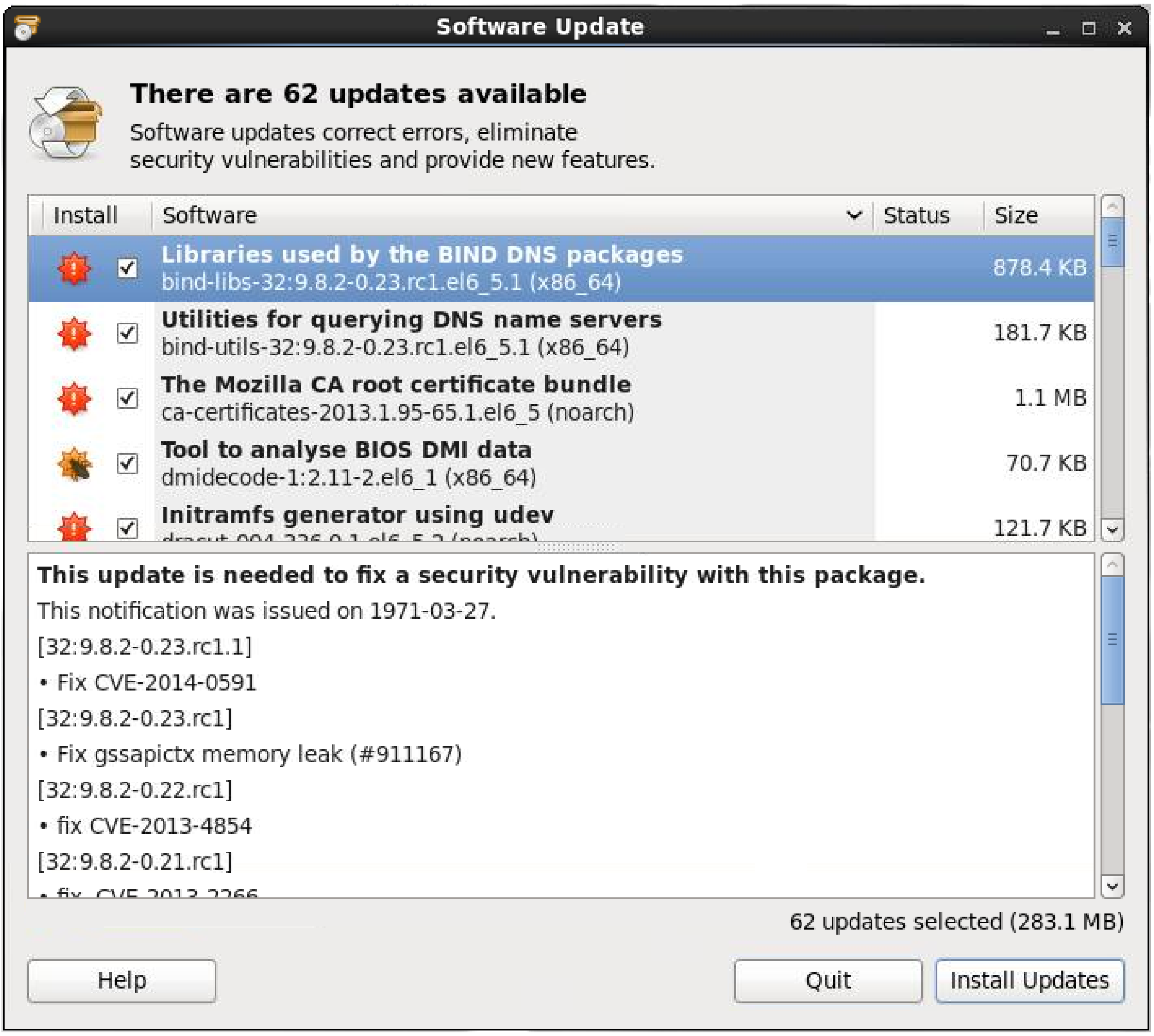
c. Enter `oracle` and click the **Authenticate** button.

- The program displays several messages as it installs the updates.

d. Click **OK** when the updates are installed successfully.



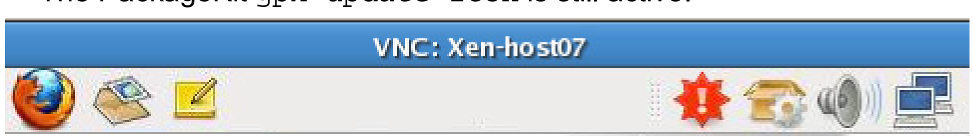
- **Note:** You might have to restart the Software Update program after the updates are installed.
- After installing these initial updates, all the remaining update packages show up in the packages area of the Software Update program.



- The number of updates available is different for your environment.
- **Do not click the Install Updates button** because it takes too long to install all of the updates.
- Use the scroll bar to view the list of updates.
- Note the various types for the updates:
  - An orange starburst with a bug for bug fixes
  - A red starburst with an exclamation point for security updates
  - A green starburst for enhancements

e. Click **Quit** to exit the Software Update program.

- The PackageKit `gpk-update-icon` is still active.

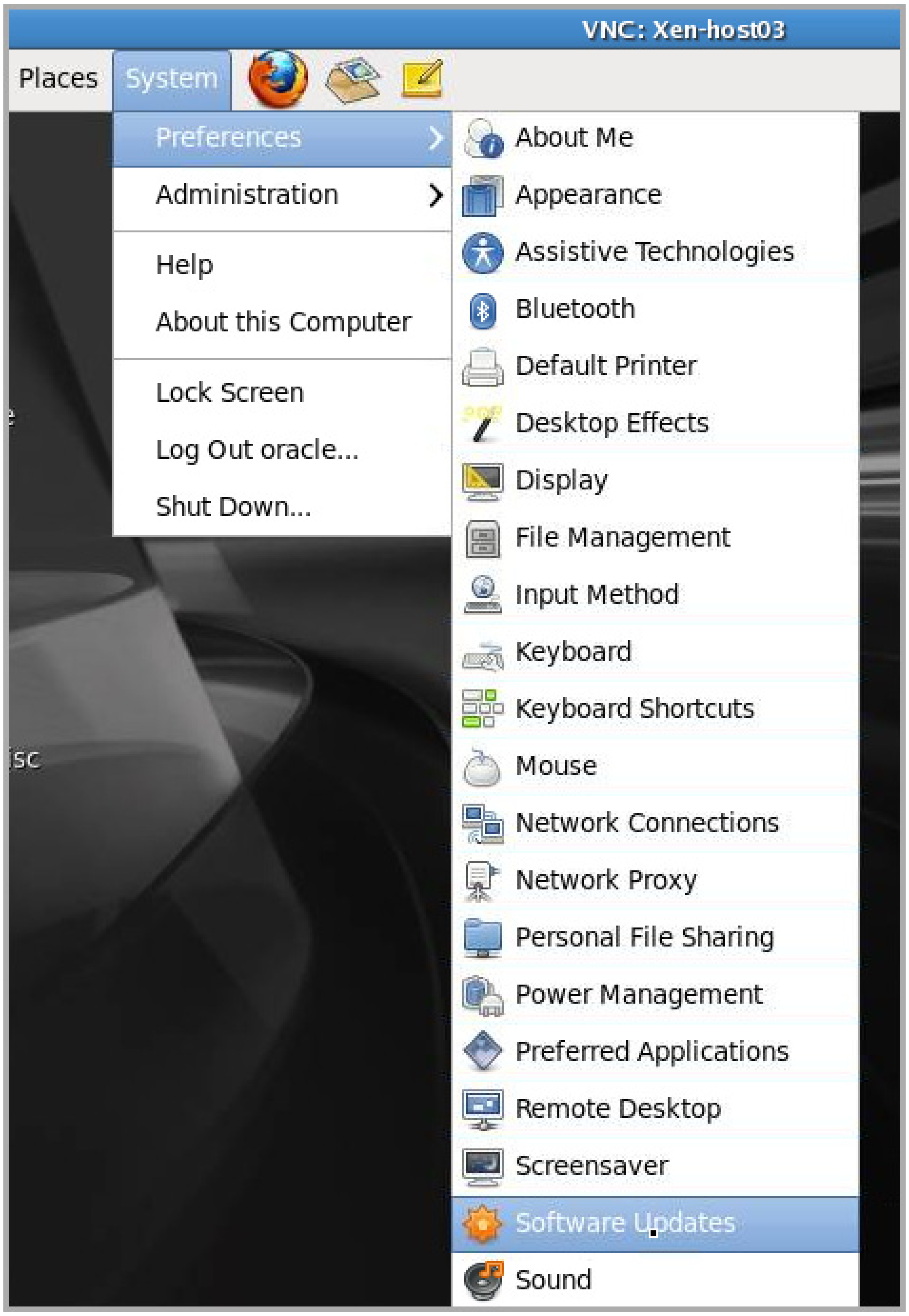


- This program displays an icon in the notification area when there are updates to install. In the next step, you disable this notification.

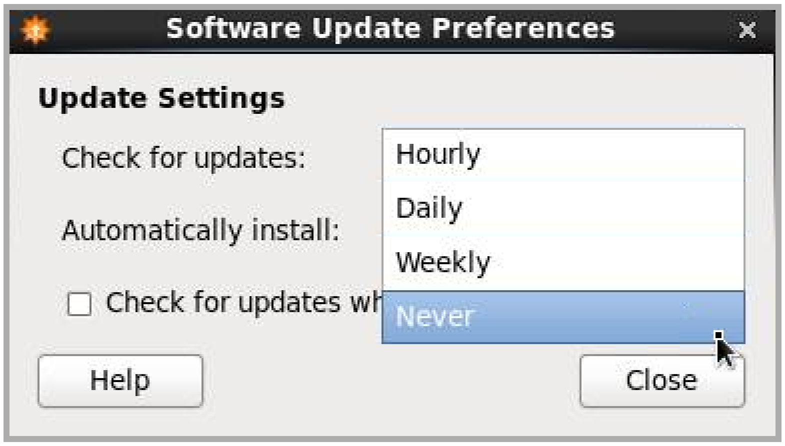
5. Disable the Software Update Automatic Notification.

- Many Linux users prefer using the Yum utilities directly rather than using the Software Update program to keep their system up to date. In this task, you disable the Software Update notification, but do not remove the ability to use the Software Update program.

a. From the GNOME desktop, select **System > Preferences > Software Updates**.



b. Select **Never** from the **Check for updates** drop-down list:



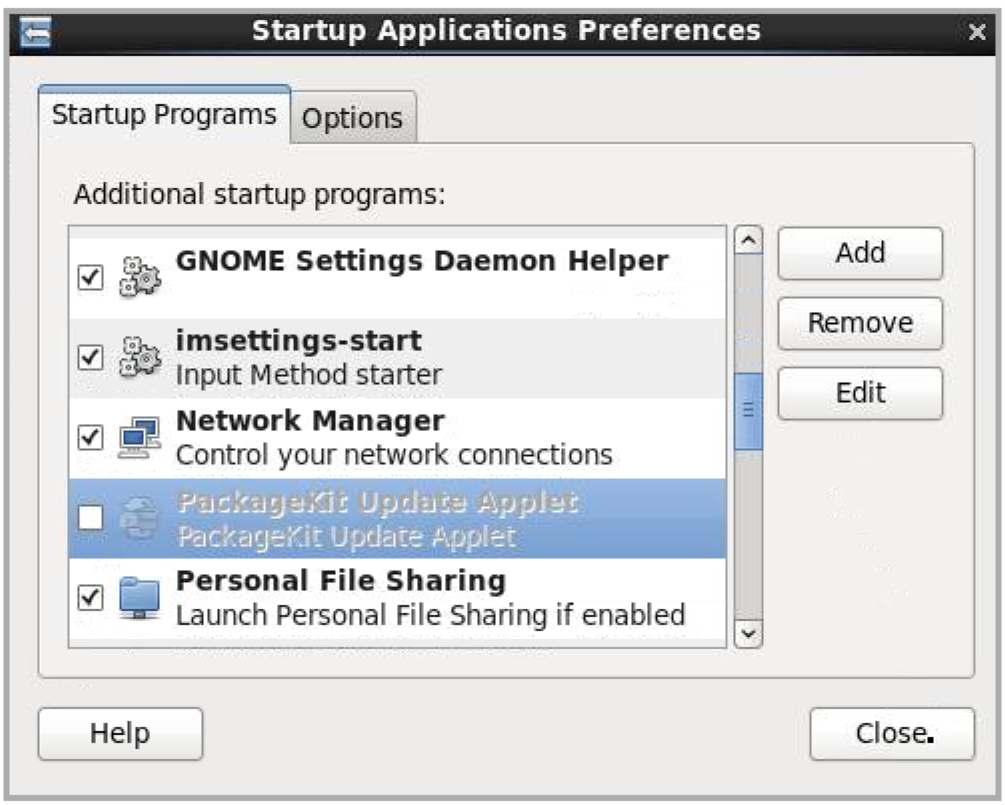
- Click **Close** to trigger the change.
- The Software Update icon is still present in the notification area. The icon disappears if you log out and log back in.

c. From the GNOME desktop, select **System > Preferences > Startup Applications**.

- The **Startup Applications Preferences** window appears.

d. In the **Startup Applications Preferences** window, scroll down, find the **PackageKit Update Applet** entry and deselect it.

- Click **Close** to trigger the change.



- When you log off and log back on, the `gpk-update-icon` application is no longer present.

Practice 7-4: Working with Yum History and Yum Cache

Overview

In this practice, you become familiar with:

- The history of transactions kept by Yum

  The history contains information about Yum transactions, such as date and time of occurrence, whether the transactions were successful, and the number of packages affected in the RPM database. You can use the history kept by Yum to undo a given transaction or to redo a transaction.

- Cache information kept by Yum

  Yum caches a variety of information to allow faster operations and, in some cases, to allow you to perform package management without a network connection. Information cached by Yum operations includes packages, header information for packages, and metadata for enabled repositories.

Assumptions

You are the `root` user on **host04**.

Tasks

1. Display Yum history information.
   a. As the `root` user on **host04**, use the `yum history` command to list transactions.
      - The following is sample output.

```
# yum history list
Loaded plugins: downloadonly, refresh-packagekit, security
ID     | Login user       | Date and time    | Action(s)      | Altered
-----------------------------------------------------------------------
     5 | System <unset>   | 2014-01-27 09:31 | Update         |    1  <
     4 | root <root>      | 2014-01-27 08:10 | Install        |    4 >
     3 | root <root>      | 2014-01-27 07:46 | Install        |    1
     2 | root <root       | 2014-01-27 07:45 | Update         |    2
     1 | System <unset>   | 2014-01-27 06:57 | Install        | 1131
history list
```

   b. Select the most recent transaction ID and display detailed information for that transaction.
      - In this example, the most recent transaction ID is `5`.

```
# yum history info 5
Loaded plugins: downloadonly, refresh-packagekit, security
Transaction ID : 5
Begin time     : ...
Begin rpmdb    : ...
...
Return-Code    : Success
Transaction performed with:
    Installed     rpm-4.8.0-37.el6.x86_64
@anaconda-OracleLinuxServer-201311252058.x86_64/6.5
```

```
    Updated       yum-3.2.29-40.0.1.el6.noarch
@anaconda-OracleLinuxServer-201311252058.x86_64/6.5
    Installed     yum-metadata-parser-1.1.2-16.el6.x86_64
@anaconda-OracleLinuxServer-201311252058.x86_64/6.5
Packages Altered:
    Updated yum-3.2.29-40.0.1.el6.noarch   @anaconda-
OracleLinuxServer-201311252058.x86_64/6.5
    Update      3.2.29-43.0.1.el6_5.noarch @public_ol6_latest
history info
#
```

2. Install the `presto` Yum plug-in and de-install it by using information in the Yum history.
   - You install this plug-in package and you de-install it in this task.
   - The `presto` plug-in allows you to download delta RPM packages instead of the full packages. This results in faster downloads but some processing is necessary on the receiving system to install the delta package, because you must rebuild the full package before you install it.

   a. Install the `yum-presto` package by using the `yum install` command.

```
# yum install yum-presto
...
Transaction Summary
==============================================================
Install       2 Package(s)

Total download size: 102 k
Installed size: 252 k
Is this ok [y/N]: y
...
Complete!
```

   b. List the Yum history to display the latest transaction.

```
# yum history list
Loaded plugins: downloadonly, presto, refresh-packagekit,
security
ID     | Login user     | Date and time    | Action(s)|Altered
--------------------------------------------------------------
     6 | Oracle Student <oracle>  ...      | Install  |   2
...
history list
```

   - The most recent transaction reflects the action taken when installing the `yum-presto` package. Two packages were installed as part of that transaction.

c. Undo the most recent transaction by using the `yum history undo <ID number>` command.
   - Replace `<ID number>` with the `ID` number obtained from your previous history listing.

```
# yum history undo 6
...
Transaction Summary
==============================================================
Remove        2 Package(s)

Installed size: 252 k
Is this ok [y/N]: y
...
Complete!
```

d. List the history again to examine the latest transaction information.

```
# yum history list
Loaded plugins: downloadonly, refresh-packagekit, security
ID     | Login user      | Date and time    | Action(s)| Altered
-------------------------------------------------------------
     7 | Oracle Student <oracle>  ...       | Erase    |    2
     6 | Oracle Student <oracle>  ...       | Install  |    2
...
history list
```

   - The packages installed by installing the `yum-presto` package are de-installed when you used the `yum history undo` command.

3. Examine Yum cache information.

a. Use the `cd` command to change to the `/var/cache/yum` directory.

```
# cd /var/cache/yum
```

b. Access each subdirectory until you reach the `6Server` directory. Use the `ls –l` command to display the contents of this directory.

```
# ls
x86_64
# cd x86_64/
# ls
6Server
# cd 6Server/
# ls -l
drwxr-xr-x. ...  InstallMedia
drwxr-xr-x. ...  public_ol6_latest
drwxr-xr-x. ...  public_ol6_UEKR3_latest
```

   - In the `/var/cache/yum/x86_64/6Server` directory, there is a subdirectory for each enabled repository.

c. Use the `cd` command to change to the `public_ol6_latest` directory. Use the `ls -l` command to display the contents of the directory.

- This directory contains the metadata for the `http://public-yum.oracle.com/repo/ OracleLinux/OL6/latest/` repository.

```
# cd public_ol6_latest
# ls -l
-rw-r--r--. ...  cachecookie
-rw-r--r--. ...  comps.xml
-rw-r--r--. ...  filelists.xml.gz
-rw-r--r--. ...  filelists.xml.gz.sqlite
drwxr-xr-x. ...  gen
-rw-r--r--. ...  other.xml.gz
-rw-r--r--. ...  other.xml.gz.sqlite
drwxr-xr-x. ...  packages
-rw-r--r--. ...  primary.xml.gz
-rw-r--r--. ...  primary.xml.gz.sqlite
-rw-r--r--. ...  repomd.xml
-rw-r--r--. ...  updateinfo.xml.gz
```

- The metadata for this repository consists of several compressed XML files that were downloaded from the Oracle Public Yum site.
- The `gen` directory contains the uncompressed `updateinfo.xml.gz` file.
- The `packages` directory contains cached packages when caching is enabled in the `/etc/yum.conf` file or if you have used the `--downloadonly` flag when using the `yum install` command.

d. Use the `ls –l` command to list the contents of the `packages` directory.

```
# ls -l packages
-rw-r--r--. ...  compat-libcap1-1.10-1.x86_64.rpm
-rw-r--r--. ...  compat-libstdc++-33-3.2.3-69.el6.x86_64.rpm
-rw-r--r--. ...  gcc-c++-4.4.7-4.el6.x86_64.rpm
-rw-r--r--. ...  ksh-20120801-10.el6.x86_64.rpm
-rw-r--r--. ...  libaio-devel-0.3.107-10.el6.x86_64.rpm
-rw-r--r--. ...  libstdc++-devel-4.4.7-4.el6.x86_64.rpm
-rw-r--r--. ...  oracle-rdbms-server-12cR1-preinstall-1.0...
-rw-r--r--. ...  xorg-x11-server-Xorg-1.13.0-23.1.el6_5...
```

- In your environment, package caching is disabled but the packages that you downloaded in Practice 7-1 are still present because you have not installed these packages. Packages are deleted after they are installed when package caching is disabled.

4. Clean the Yum cache.
   a. Use the `yum clean packages` command to clean the packages in the Yum cache.

```
# yum clean packages
Loaded plugins: downloadonly, refresh-packagekit, security
Cleaning repos: public_ol6_UEKR3_latest public_ol6_latest
8 package files removed
```

   b. Use the `ls` command to list the contents of the `packages` directory.

```
# ls packages
```

   - The packages are no longer present.

   c. Use the `ls -l` command to list the contents of the `gen` directory.

```
# ls -l gen
-rw-r--r--. ...  updateinfo.xml
```

   - This directory contains the uncompressed data from `updateinfo.xml.gz`.

   d. Use the `yum clean metadata` command to clean the metadata in the Yum cache.

```
# yum clean metadata
Loaded plugins: downloadonly, refresh-packagekit, security
Cleaning repos: public_ol6_UEKR3_latest public_ol6_latest
15 metadata files removed
6 sqlite files removed
0 metadata files removed
```

   - The number of files removed might differ in your environment.

   e. Use the `ls –l` command to list the contents of the current directory, `/var/cache/yum/x86_64/6Server/ public_ol6_latest` and the `gen` subdirectory, and note the effect of the `yum clean metadata` command.

```
# ls -l
drwxr-xr-x. ...  gen
drwxr-xr-x. ...  packages
# ls -l gen
total 0
```

   - The directories are empty.
   - The metadata files are gone not only in this directory but in each directory corresponding to an enabled Oracle Public Yum repository.
   - There are other variations of the `yum clean` command. Consult the `yum` man page for more information about cleaning the Yum cache.
   - You can also use the `yum clean all` command to clean all cached information.
   - If you experience problems accessing packages and package information from the Oracle Public Yum or from the Oracle Unbreakable Linux Network (ULN) site, it is often helpful to issue the `yum clean metadata` command. This forces `yum` to download the latest metadata the next time it is invoked.

5. Shut down **host04** and start **host03**.
   a. Use the `shutdown -h now` command to shut down **host04**.

```
# shutdown -h now
```

   - Your VNC window closes.

   b. From a terminal window on **dom0**, use the `cd` command to change to the `/OVS/running_pool/host03` directory.

```
# cd /OVS/running_pool/host03
```

   c. Run the `xm create vm.cfg` command to start the **host03** VM.

```
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host03 (id=...)
```

   d. Run the `xm list` command to list the running VMs.

```
# xm list
Name                     ID     Mem  VCPUs       State      Time(s)
Domain-0                  0    2048      2      r-----       758.9
host01                    4    1536      1      -b----        37.4
host02                    5    1536      1      -b----        37.3
host03                   15    1536      1      -b----        37.3
```

   Only the **host01**, **host02**, and **host03** VMs are running.

Practices for Lesson 8: Advanced Storage Administration

Chapter 8

Practices for Lesson 8: Advanced Storage Administration

Practices Overview

In these practices, you:

- Create and mount a file system on `/dev/xvdb`
- Set access control lists (ACLs) on a file system
- Set quotas on a directory
- Encrypt a file system
- Use the `kpartx` utility
- Explore and configure Udev

Practice 8-1: Creating and Mounting a File System

Overview

In this practice, you:

- Create a partition on a storage device
- Create an ext4 file system on the partition
- Mount the file system on `/Dev`
- Update the file system mount table

Assumptions

- You are the `root` user on **dom0**.

Tasks

1. Connect to **host03** using `vncviewer`.
   - Refer to Practice 3-1 for instructions if necessary.
2. Log on to **host03**.
   a. Select **Oracle Student** from the GNOME desktop login window to login as `oracle` user. Password is `oracle`.
   b. Open a terminal window on the GNOME desktop
   c. Use the `su –` command to become the `root` user. The password is `oracle`.
3. Partition a storage device using `fdisk`.
   a. As the `root` user on **host03**, use the `fdisk` command to display the partition table.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *   1     64     512000   83  Linux
/dev/xvda2      64    1567  12069888   8e  Linux LVM
Disk /dev/xvdb: 10.7 GB,  10737418240 bytes
Disk /dev/xvdd: 10.7 GB,  10737418240 bytes
Disk /dev/mapper/vg_host03-lv_root: 11.1 GB, 11068768256 bytes
Disk /dev/mapper/vg_host03-lv_swap: 1287 MB, 1287651328 bytes
```

   - This lists the following three storage devices:
     - `/dev/xvda`, approximately 12 GB in size
     - `/dev/xvdb`, approximately 10 GB in size
     - `/dev/xvdd`, approximately 10 GB in size
   - The operating system is installed on the `/dev/xvda` device.
   - The `/dev/xvdb` and `/dev/xvdd` devices are unused.

b. Use the `fdisk` command to partition `/dev/xvdb`.

```
# fdisk /dev/xvdb
...
Command (m for help):
```

c. Create a 1 GB primary partition as follows.

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1): ENTER
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1305, default
1305): +1G
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

d. Use the `fdisk` command to list the partition table on `/dev/xvdb`.

```
# fdisk –l /dev/xvdb
Disk /dev/xvdb: 10.7 GB,  10737418240 bytes
...
    Device Boot   Start      End     Blocks     Id    System
/dev/xvdb1            1      132    1060258+    83    Linux
```

4. Create a file system on `/dev/xvdb1`.

   Use the `mkfs` command to make an **ext4** file system on `/dev/xvdb1`.

```
# mkfs –t ext4 /dev/xvdb1
mke2fs 1.43-WIP (20-June-2013)
Filesystem label=
OS type: Linux
...
```

5. Mount the file system.

a. Use the `mkdir` command to create a mount point.

```
# mkdir /Dev
```

b. Use the `mount` command to mount `/dev/xvdb1` on `/Dev` with ACL support.

- Include the `-o acl` mount option for ACL support.

```
# mount –t ext4 –o acl /dev/xvdb1 /Dev
```

c. Use the `df` command to display the mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/xvdb1    988M  1.3M   935M    1%  /Dev
```

6. Update the file systems mount table.

   Use the `vi` editor to add the following entry to `/etc/fstab`.

```
# vi /etc/fstab
/dev/xvdb1   /Dev   ext4  acl  0  0
```

Practice 8-2: Implementing Access Control Lists

Overview

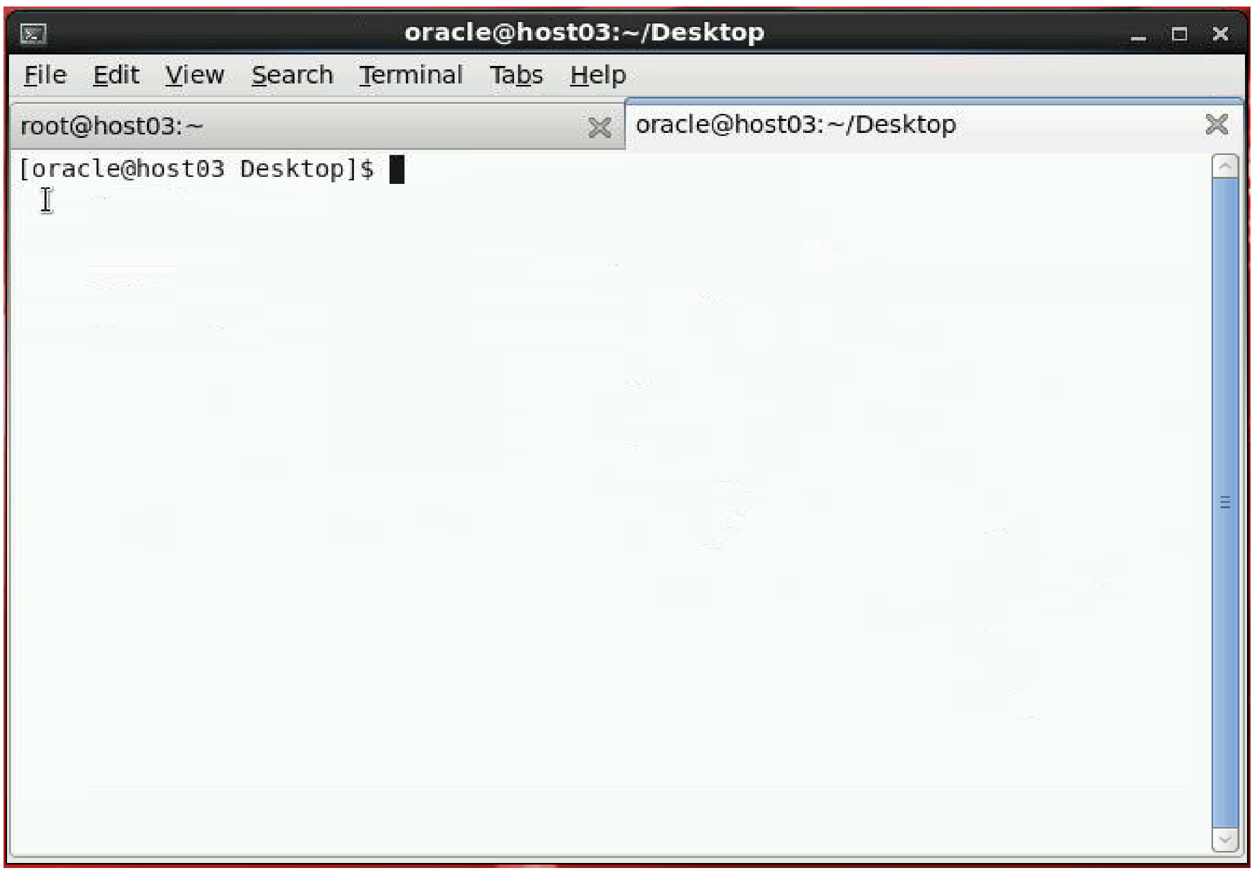
In this practice, you set ACLs on a directory.

Assumptions

- Ensure that you are using `vncviewer` to connect to **host03** and not using `ssh`.
- You are the `root` user on **host03** VM.
- You will switch between the `root` user and the `oracle` user for this practice.

Tasks

1. Open a tab in the current window.
   From the terminal window menu bar, select **File > Open Tab**, or press **Shift + Ctrl + T**.
   - Your window should look similar to the following screenshot.
   - You are the `root` user in one tab and you are the `oracle` user in the other.



2. As the `oracle` user, use the `touch` command to create the `test` file in the `/Dev` directory.

```
[oracle@host03]$ touch /Dev/test
touch: cannot touch ‘Dev/test’: Permission denied
```

- Notice that you do not have permission to create files in the `/Dev` directory.

3. As the `root` user, use the `getfacl` command to display the `/Dev` directory's ACL.
   - Click the **root@host03:** tab to enter commands as the `root` user.

```
[root@host03]# getfacl /Dev
getfacl: Removing leading '/' from absolute path names
# file: Dev
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

4. As the `root` user, use the `setfacl` command to add a rule to the ACL giving the `oracle` user read, write, and execute permissions to the `/Dev` directory.

```
[root@host03]# setfacl –m u:oracle:rwx /Dev
```

5. As the `root` user, use the `getfacl` command to display the `/Dev` directory's ACL.

```
[root@host03]# getfacl /Dev
getfacl: Removing leading '/' from absolute path names
# file: Dev
# owner: root
# group: root
user::rwx
user:oracle:rwx
group::r-x
mask::rwx
other::r-x
```

   - Notice the new `user:oracle:rwx` line in the output of the `getfacl` command.

6. As the `root` user, use the `ls –ld` command to display the permissions for the `/Dev` directory.

```
[root@host03]# ls -ld /Dev
drwxrwxr-x+ ... /Dev
```

   - Notice the plus sign (`+`), indicating that the directory has an ACL.

7. As the `oracle` user, use the `touch` command to create the `test` file in the `/Dev` directory.
   - Click the **oracle@host03:** tab to enter commands as the `oracle` user.

```
[oracle@host03]$ touch /Dev/test
```

   - Notice that the command succeeded this time.

8. As the `oracle` user, use the `ls` command to display a long listing of the `/Dev` directory.

```
[oracle@host03]$ ls -l /Dev
drwx------. 2 root    root ...   lost+found
-rw-rw-r--. 1 oracle  oracle ... test
```

   - Notice that the `test` file is owned by the `oracle` user.

Practice 8-3: Setting Disk Quotas

Overview

In this practice, you set quotas on a directory for the `oracle` user. You also remove the quotas and the ACL on the directory.

Assumptions

You switch between the `root` user and the `oracle` user for this practice.

Tasks

1. As the `root` user, use the `vi` editor to change the entry in `/etc/fstab` for `/Dev` and include ACL support and the user and group quota options.
   - Click the **root@host03:** tab to enter commands as the `root` user.

```
[root@host03]# vi /etc/fstab
/dev/xvdb1 /Dev  ext4  acl                    0  0 (old entry)
/dev/xvdb1 /Dev  ext4  acl,usrquota,grpquota  0  0 (new entry)
```

2. As the `root` user, use the `mount` command with the `–o remount` option to remount `/Dev`.

```
[root@host03]# mount –o remount /Dev
```

3. As the `root` user, use the `mount` command to display the new mount options on `/Dev`.

```
[root@host03]# mount | grep /Dev
/dev/xvdb1 on /Dev type ext4 (rw,acl,usrquota,grpquota)
```

4. As the `root` user, use the `quotacheck` command to create disk usage tables for `/Dev`.

```
[root@host03]# quotacheck –cug /Dev
quotacheck: Cannot create new quotafile /Dev/aquota.user.new
(Permission denied
...
```

   - Notice the “Permission denied” message.
   - This is caused by an SELinux policy enforcement.
   - You could configure SELinux to allow the creation of disk usage tables.
   - But for purposes of this practice, set SELinux to permissive mode.

```
[root@host03]# getenforce
Enforcing
[root@host03]# setenforce 0
[root@host03]# getenforce
Permissive
```

   - SELinux is covered in “Lesson 16: SELinux.”

5. As the `root` user, use the `quotacheck` command to create disk usage tables for `/Dev`.

```
[root@host03]# quotacheck –cug /Dev
```

   - Notice that the command succeeded this time.

6. As the `root` user, use the `ls` command to display the files created in `/Dev`.

```
[root@host03]# ls -l /Dev
-rw-------. root root ... aquota.group
-rw-------. root root ... aquota.user
...
```

7. As the `root` user, use the `quotaon` command to enable quotas on `/Dev`.

```
[root@host03]# quotaon /Dev
```

8. As the `root` user, use the `repquota` command to report disk usage on `/Dev`.

```
[root@host03]# repquota /Dev
*** Report for user quotas on device /dev/xvdb1
Block grace time: 7days; Inode grace time: 7days
                  Block limits             File limits
User       used  soft    hard  grace  used  soft  hard  grace
-------------------------------------------------------------
root   --    20    0       0             2     0     0
oracle --     0    0       0             1     0     0
```

9. As the `root` user, use the `edquota` command to limit the `oracle` user.
   - This command invokes the `vi` editor.

   Change the block quota to set a hard limit of 2048 blocks (2 MB) for the `oracle` user.

```
[root@host03]# edquota oracle
Disk quotas for user oracle (uid 500):
Filesystem  blocks  soft  hard  inodes  soft  hard
/dev/xvdb1       0     0     0       1     0     0 (old entry)
/dev/xvdb1       0     0  2048       1     0     0 (new entry)
```

   - Alternatively, you could use the `setquota oracle 0 2048 0 0 /Dev` command.

10. As the `root` user, use the `repquota` command to report disk usage on `/Dev`.

```
[root@host03]# repquota /Dev
*** Report for user quotas on device /dev/xvdb1
Block grace time: 7days; Inode grace time: 7days
                  Block limits             File limits
User       used  soft    hard  grace  used  soft  hard  grace
-------------------------------------------------------------
root   --    20    0       0             2     0     0
oracle --     0    0    2048             1     0     0
```

11. As the `oracle` user, use the `dd if=/dev/zero of=bigfile bs=1M count=4096` command to attempt to create a 4 MB file on `/Dev`.
    - Click the **oracle@host03:** tab to enter commands as the `oracle` user.

```
[oracle@host03]$ cd /Dev
[oracle@host03]$ dd if=/dev/zero of=bigfile bs=1M count=4096
xvdb1: write failed, user block limit reached.
```

```
dd: writing ‘bigfile’: Disk quota exceeded
3+0 records in
1+0 records out
2088960 bytes (2.1 MB) copied, ...
```

- Notice the “Disk quota exceeded” error message.

12. As the `oracle` user, use the `ls` command to display a long listing of the `/Dev` directory.

```
[oracle@host03]$ ls -l /Dev
...
-rw-rw-r--. 1 oracle  oracle 2097152 ... bigfile
...
```

- Notice that the `bigfile` is not 4 MB, but was truncated after quota limits were reached.

13. As the `oracle` user, use the `quota` command to display quota information.

```
[oracle@host03]$ quota
Disk quotas for user oracle (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/xvdb1   2048*    0  2048            2     0     0
```

14. As the `oracle` user, use the `rm` command to delete the `bigfile` file in the `/Dev` directory.

```
[oracle@host03]$ rm /Dev/bigfile
```

15. As the `oracle` user, use the `quota` command to display quota information.

```
[oracle@host03]$ quota
Disk quotas for user oracle (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/xvdb1      0     0  2048            1     0     0
```

- Notice the difference in the number of blocks and number of files from step 13.

16. As the `oracle` user, use the `rm` command to delete the `test` file in the `/Dev` directory.

```
[oracle@host03]$ rm /Dev/test
```

17. As the `root` user, use the `setquota oracle 0 0 0 0 /Dev` command to reset the disk quota for the `oracle` user.
    - Click the **root@host03:** tab to enter commands as the `root` user.

```
[root@host03]# setquota oracle 0 0 0 0 /Dev
```

18. Remove ACL from `/Dev`.
    a. As the `root` user, use the `setfacl` command to remove the ACL from the `/Dev` directory.

```
[root@host03]# setfacl -b /Dev
```

    b. As the `root` user, use the `getfacl` command to display the `/Dev` directory’s ACL.

```
[root@host03]# getfacl /Dev
getfacl: Removing leading ‘/’ from absolute path names
# file: Dev
# owner: root
```

```
# group: root
user::rwx
group::r-x
other::r-x
```

- Notice that the `user:oracle:rwx` line in the output has been removed.

c. As the `root` user, use the `ls -ld` command to display the permissions for the `/Dev` directory.

```
[root@host03]# ls -ld /Dev
drwxr-xr-x ... /Dev
```

- Notice that there is no plus sign (`+`), indicating that the directory has no ACL.

19. As the `root` user, use the `vi` editor to change the entry in `/etc/fstab` for `/Dev` to remove the ACL and the user and group quota options.

```
[root@host03]# vi /etc/fstab
/dev/xvdb1  /Dev  ext4  acl,usrquota,grpquota 0  0 (old entry)
/dev/xvdb1  /Dev  ext4  defaults              0  0 (new entry)
```

20. As the `root` user, use the `mount` command with the `–o remount` option to remount `/Dev`.

```
[root@host03]# mount –o remount /Dev
```

21. As the `root` user, use the `mount` command to display the new mount options on `/Dev`.

```
[root@host03]# mount | grep /Dev
/dev/xvdb1 on /Dev type ext4 (rw)
```

22. Click the **X** on the **oracle@host03:** tab to close the tab.
23. Set SELinux to enforcing mode on **host03**.

```
[host03]# getenforce
Permissive
[host03]# setenforce 1
[host03]# getenforce
Enforcing
```

Practice 8-4: Encrypting a File System

Overview

In this practice, you create an encrypted file system, create a file system on the encrypted volume, reboot your system and provide the passphrase to mount the encrypted file system, and remove the encrypted file system.

Assumptions

You are the `root` user on **host03** VM.

Tasks

1. Unmount the file system on `/Dev`.

   Use the `umount` command to unmount /Dev.

   ```
   # umount /Dev
   ```

2. Set up a cryptographic volume.

   a. Use the `cryptsetup` command with `luksFormat` to initialize the `/dev/xvdb1` volume and set an initial key of `pass`.

   - The "`pass`" entries are not displayed for security reasons.

   ```
   # cryptsetup luksFormat /dev/xvdb1
   WARNING!
   ========
   This will overwrite data on /dev/xvdb1 irrevocably.
   Are you sure? (Type uppercase yes): YES
   Enter LUKS passphrase: pass
   Verify passphrase: pass
   ```

   b. Use the `cryptsetup` command with `luksOpen` to open the partition and create the device mapping of `cryptfs`.

   ```
   # cryptsetup luksOpen /dev/xvdb1 cryptfs
   Enter passphrase for /dev/xvdb1: pass
   ```

   c. Use the `cryptsetup` command to check the status of the encrypted volume.

   ```
   # cryptsetup status cryptfs
   /dev/mapper/cryptfs is active.
     type:  LUKS1
     cipher:  aes-cbs-essiv:sha256
     keysize: 256 bits
     device:  /dev/xvdb1
     offset:  4096 sectors
     size:    2116421 sectors
     mode:    read/write
   ```

   d. Use the `blkid` command to view the attributes of the `/dev/xvdb1` block device.

   ```
   # blkid /dev/xvdb1
   /dev/xvdb1: UUID=... TYPE="crypto_LUKS"
   ```

e. Use the `ls -l` command to list the `/dev` entry for the `cryptfs` encrypted volume.

```
# ls -l /dev/mapper
...
lrwxrwxrwx.  cryptfs -> ../dm-2
lrwxrwxrwx.  vg_host03-lv_root -> ../dm-0
lrwxrwxrwx.  vg_host03-lv_swap -> ../dm-1
```

3. Create a file system on the encrypted volume.

a. Use the `mkfs.ext4` command to create an **ext4** file system.

```
# mkfs.ext4 /dev/mapper/cryptfs
mke2fs 1.43-WIP (20-Jun-2013)
Filesystem label=
OS type: Linux
...
Writing inode tables: done
Creating journal (8192 blocks): done
...
```

b. Use the `mkdir` command to create a mount point named `/cryptfs`.

```
# mkdir /cryptfs
```

c. Use the `mount` command to mount the file system.

```
# mount /dev/mapper/cryptfs /cryptfs
```

d. Display the mounted file systems.

```
# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/cryptfs
               986M  1.3M   933M    1%  /cryptfs
```

4. Update the `/etc/crypttab` configuration file and the file system mount table.

a. Use the `vi` editor to create `/etc/crypttab` and to add the following entry.

```
# vi /etc/crypttab
cryptfs /dev/xvdb1 none luks
```

b. Use the `vi` editor to create the following line to `/etc/fstab`.

- Edit the existing entry for the `/dev/xvdb1` device.

```
# vi /etc/fstab
/dev/xvdb1          /Dev      ext4  defaults  0  0 (old entry)
/dev/mapper/cryptfs /cryptfs  ext4  defaults  0  0 (new entry)
```

- After updating these files, the encrypted file system is properly set up and mounted at reboot.

5. Reboot your system and enter the passphrase to mount the encrypted file system.
   a. Use the `reboot` command to reboot your system.

```
# reboot
```

   - After you reboot your system, your VNC session closes.

   b. From **dom0**, connect to **host03** guest by using `vncviewer`.

```
# vncviewer&
```

   - The **VNC Viewer: Connection Details** window appears.

   c. Enter the command, `localhost:<port_number>`, substituting the correct port number for the **host03** guests. For example, if the port number is `5903`, enter the following and click **OK**.

```
localhost:5903
```

   d. Provide the passphrase, `pass`, when prompted for the encrypted file system passphrase during reboot.

```
/cryptfs is password protected: pass
```

   - The boot process continues after providing the correct passphrase.

6. Remove the encrypted file system.
   a. Log in as `Oracle Student` with password `oracle`.
   b. Open a terminal window.
   c. Become the `root` user. The password is `oracle`.

```
$ su -
Password: oracle
# whoami
root
```

   d. Display the mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/cryptfs
              986M  1.3M   933M    1%  /cryptfs
...
```

   - Notice that the encrypted file system is mounted.

   e. Unmount the encrypted file system, `/cryptfs`.

```
# umount /cryptfs
```

   f. Using the `vi` editor, remove the following entry from `/etc/crypttab`.

```
# vi /etc/crypttab
cryptfs /dev/xvdb2 none luks
```

   g. Using the `vi` editor, remove the following entry from `/etc/fstab`.

```
# vi /etc/fstab
/dev/mapper/cryptfs /cryptfs  ext4  defaults  0  0
```

h. Use the `cryptsetup` command with `luksOpen` to remove the device mapping.

```
# cryptsetup luksClose /dev/mapper/cryptfs
```

i. Verify that the `cryptfs` device mapping has been removed.

```
# ls /dev/mapper
```

Practice 8-5: Using `kpartx`

Overview

In this practice, you use the `kpartx` utility to create device maps from partitions tables.

Assumptions

- This practice is performed on **dom0** and on **host03** VM.
- You are logged in as the `root` user on **dom0** and **host03**.

Tasks

1. Review the **host03** virtual disk configuration.
   a. From **dom0**, use the `cd` command to change to the `/OVS/running_pool/host03` directory on **dom0**.

   ```
   [dom0]# cd /OVS/running_pool/host03
   ```

   b. Use the `ls -l` command to list the contents of the directory.

   ```
   [dom0]# ls -l
   -rw-r--r-- 12884901888 system.img
   -rw-r--r-- 10737418240 u01.img
   -rw-r--r-- 10737418240 u02.img
   -rw-r--r--         733 vm.cfg
   ```

   - The `system.img` file is represented by `/dev/xvda`.
   - The `u01.img` file is represented by `/dev/xvdb`.
   - The `u02.img` file is represented by `/dev/xvdd`.

   c. Use the `cat` command to view the `vm.cfg` file.

   ```
   [dom0]# cat vm.cfg
   name = "host03"
   builder = "hvm"
   memory = "1536"
   boot = 'cd'
   disk = [ 'file:/OVS/running_pool/host03/system.img,hda,w',
            'file:/OVS/running_pool/host03/u01.img,hdb,w',
            'file:/OVS/running_pool/host03/u02.img,hdd,w',
   ...
   ```

   - Notice that the `system.img` file is mapped to `hda` or `/dev/xvda`.
   - Notice that the `u01.img` file is mapped to `hdb` or `/dev/xvdb`.
   - Notice that the `u02.img` file is mapped to `hdd` or `/dev/xvdd`.

2. Review the partition information on the `system.img` file.
   a. From **dom0**, use the `kpartx -l` command to list the partitions on the `system.img` disk image file.

```
[dom0]# kpartx -l system.img
loop2p1 : 0 1024000 /dev/loop2 2048
loop2p2 : 0 24139776 /dev/loop2 1026048
```

   - The output shows that the `system.img` disk image file contains two partitions.

   b. From **host03** VM, use the `fdisk` command to list the partition table for `/dev/xvda`.

```
[host03]# fdisk -l | grep /dev/xvda
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *   1     64    512000   83  Linux
/dev/xvda2      64   1567  12069888   83  Linux LVM
```

   - Notice that `/dev/xvda` has two partitions.
   - This confirms that the `system.img` file is mapped to `/dev/xvda`.

3. Review the partition information on the `u01.img` file.
   a. From **dom0**, use the `kpartx -l` command to list the partitions on the `u01.img` disk image file.

```
[dom0]# kpartx -l u01.img
loop3p1 : 0 2120517 /dev/loop3 63
```

   - The output shows one partition.

   b. From **host03** VM, use the `fdisk` command to list the partition table for `/dev/xvdb`.

```
[host03]# fdisk -l | grep /dev/xvdb
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
/dev/xvdb1       1    132  1060258+  83  Linux
```

   - The output shows one partition.
   - This confirms that the `u01.img` file is mapped to `/dev/xvdb`.

4. Review the partition information on the `u02.img` file.
   a. From **dom0**, use the `kpartx -l` command to list the partitions on the `u02.img` disk image file.

```
[dom0]# kpartx -l u02.img
```

   - The output shows no partitions.

   b. From **host03** VM, use the `fdisk` command to list the partition table on `/dev/xvdd`.

```
[host03]# fdisk -l | grep /dev/xvdd
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
```

   - The output shows no partitions on `/dev/xvdd`.
   - This confirms that the `u02.img` file is mapped to `/dev/xvdd`.

5. Create and mount a file system on `/dev/xvdb1`.
   a. From **host03**, use the `mkfs` command to make an **ext3** file system on `/dev/xvdb1`.

```
[host03]# mkfs -t ext3 /dev/xvdb1
mke2fs 1.43-WIP (20-June-2013)
Filesystem label=
OS type: Linux
...
```

   b. Use the `mount` command to mount `/dev/xvdb1` on `/Dev`.

```
[host03]# mount /dev/xvdb1 /Dev
```

   c. Use the `df` command to display the mounted file systems.

```
[host03]# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/xvdb1   1020M   34M   935M    4%  /Dev
```

   d. Use the `cp` command to copy the `init*` files from `/boot` to `/Dev`.
      - These files are viewed later in this practice to confirm the success of the `kpartx` command.

```
[host03]# cp /boot/init* /Dev
[host03]# ls /Dev
initramfs-2.6.32-431.el6.x86_64.img           lost+found
initramfs-3.8.13-16.2.1.el6uek.x86_64.img
initramfs-3.8.13-26.1.1.el6uek.x86_64.img
```

The remaining commands in this practice are entered from **dom0**.

6. Create device maps from the partition table on `u01.img`.
   a. From **dom0**, use the `ls` command to list the `/dev/mapper` directory.

```
[dom0]# ls /dev/mapper
control
```

      - Before adding the device files, a listing of `/dev/mapper` shows only the `control` file.

   b. Use the `kpartx -l` command to list the partitions on the `u01.img` disk image file.
      - Recall that `u01.img` maps to `/dev/xvdb`.

```
[dom0]# kpartx -l u01.img
loop3p1 : 0 2120517 /dev/loop3 63
```

      - This confirms that there is one partition on `/dev/xvdb`.

   c. Use the `kpartx -a` command to add the device mappings for the detected partitions.
      - To save time, this practice is not requiring you to shut down the **host03** VM before using the `kpartx -a` command.
      - A best practice would be to shut down **host03** before creating device mappings and before mounting the devices on **dom0**.

```
[dom0]# kpartx -a u01.img
```

d. Use the `ls` command to list the `/dev/mapper` directory.

```
[dom0]# ls /dev/mapper
control   loop3p1
```

- Notice that a file was created for the partition on `/dev/xvdb`.

7. Mount the device created by the `kpartx` command.

a. From **dom0**, use the `mkdir` command to create a mount point, `/mnt/map1`.

```
[dom0]# mkdir /mnt/map1
```

b. Use the mount command to mount `/dev/mapper/loop3p1` on `/mnt/map1`.

```
[dom0]# mount /dev/mapper/loop3p1 /mnt/map1
```

c. Use the `ls` command to view the files on `/mnt/map1`.

```
[dom0]# ls /mnt/map1
initramfs-2.6.32-431.el6.x86_64.img          lost+found
initramfs-3.8.13-16.2.1.el6uek.x86_64.img
initramfs-3.8.13-26.1.1.el6uek.x86_64.img
```

- Notice that these are the same files that you copied to `/Dev` in step 5d.

8. Remove the `kpartx` device mapping on **dom0**.

a. From **dom0**, use the `umount` command to unmount `/mnt/map1`.

```
[dom0]# umount /mnt/map1
```

b. Use the `rmdir` command to delete `/mnt/map1`.

```
[dom0]# rmdir /mnt/map1
```

c. Use the `kpartx -d` command to disconnect the device.

```
[dom0]# # kpartx -d u01.img
loop deleted : /dev/loop3
```

d. Use the `ls` command to list the contents of `/dev/mapper`.

```
[domo]# ls /dev/mapper
control
```

- Notice that the device mapping no longer exists in `/dev/mapper`.

Practice 8-6: Exploring and Configuring Udev

Overview

In this practice, you:

- Explore Udev files and directories
- Query the Udev database
- Create a Udev rule to change the name of a device

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Explore Udev.
   a. Use the `ls` command to view existing Udev rules files in the `/lib/udev/rules.d` and `/etc/udev/rules.d` directories.

```
# ls /lib/udev/rules.d
01-log-block.rules      71-biosdevname.rules
10-console.rules        75-cd-aliases-generator.rules
10-dm.rules             75-net-description.rules
11-dm-lvm.rules         75-persistent-net-generator.rules
...
# ls /etc/udev/rules.d
60-fprint-autosuspend.rules     70-persistent-cd.rules
60-pcmcia.rules                 90-alsa.rules
...
```

   b. Use the `less` command to view the `/lib/udev/rules.d/50-udev-default.rules` file.
      - Page through the file. Press `q` to return to the command prompt.

```
# less /lib/udev/rules.d/50-udev-default.rules
# do not edit this file, it will be overwritten on update
SUBSYSTEM==“block”, SYMLINK{unique}+=“block/%M:%m”
SUBSYSTEM!=“block”, SYMLINK{unique}+=“char/%M:%m”
KERNEL==“ptty[pqrstuvwxyzabcdef][0123456789abcdef]”,
GROUP=“tty”, MODE=“0660”
KERNEL==“tty[pqrstuvwxyzabcdef][0123456789abcdef]”, GROUP=“tty”,
MODE=“0660”
...
```

2. Query the Udev database.
   - Sample output is shown. Your output might be different.
   a. Use the `udevadm` command to query the Udev database for all device information for `/dev/xvdd`.

```
# udevadm info --query=all --name=/dev/xvdd
P: /devices/vbd-5696/block/xvdd
```

```
N: xvdd
W: 21
S: block/202:48
S: disk/by-path/xen-vbd-5696
E: UDEV_LOG=3
E: DEVPATH=/devices/vbd-5696/block/xvdd
E: MAJOR=202
E: MINOR=48
E: DEVNAME=/dev/xvdd
E: DEVTYPE=disk
E: SUBSYSTEM=block
E: ID_PATH=xen-vbd-5696
E: ID_PART_TABLE_TYPE=dos
E: LVM_SBIN_PATH=/sbin
E: DEVLINKS=/dev/block/202:48 /dev/disk/by-path/xen-vbd-5696
E: UDISKS_PRESENTATION_NOPOLICY=1
E: UDISKS_PARTITION_TABLE=1
E: UDISKS_PARTITION_TABLE_SCHEME=mbr
E: UDISKS_PARTITION_TABLE_COUNT=0
```

b. Use the `udevadm` command to query the Udev database for the device path of `/dev/xvdd`.

```
# udevadm info --query=path --name=/dev/xvdd
/devices/vbd-5696/block/xvdd
```

c. Use the `udevadm` command to print all `sysfs` properties of `/dev/xvdd`.

```
# udevadm info --attribute-walk --name=/dev/xvdd
...
  looking at device '/devices/vbd-5696/block/xvdd':
    KERNEL=="xvdd"
    SUBSYSTEM=="block"
    DRIVER==""
    ATTR{range}=="16"
...
  looking at parent device '/devices/vbd-5696':
    KERNELS=="vbd-5696"
    SUBSYSTEMS=="xen"
    DRIVERS=="vbd"
    ATTR{nodename}=="device/vbd/5696"
    ATTR{devtype}=="vbd"
    ATTR{modalias}=="xen:vbd"
```

3. Change the name of a device.
   a. Use the `vi` editor to create the `/etc/udev/rules.d/10-local.rules` file as follows:
      - Use the `KERNEL` and `SUBSYSTEM` values from the previous "`udevadm info --attribute-walk`" command.

```
# vi /etc/udev/rules.d/10-local.rules
KERNEL=="xvdd", SUBSYSTEM=="block", NAME="my_disk"
```

      - The `NAME` directive provides the new name for the device.

   b. Use the `ls` command to list the `/dev/xv*` devices.

```
# ls /dev/xv*
/dev/xvda /dev/xvda1 /dev/xvda2 /dev/xvdb /dev/xvdb1 /dev/xvdd
```

   c. Run the `start_udev` command to process the rules files.

```
# start_udev
Starting udev:                                    [  OK  ]
```

   d. Use the `ls` command to list the `/dev/xv*` devices.

```
# ls /dev/xv*
/dev/xvda /dev/xvda1 /dev/xvda2 /dev/xvdb /dev/xvdb1
```

      - Notice that `/dev/xvdd` no longer exists.
      - It has been renamed to `/dev/my_disk`.

   e. Use the `ls` command to list the `/dev/my*` devices.

```
# ls /dev/my*
/dev/my_disk
```

4. Restore the original name of a device.
   a. Use the `rm` command to remove the `/etc/udev/rules.d/10-local.rules` file.

```
# rm /etc/udev/rules.d/10-local.rules
rm: remove regular file '/etc/udev/rules.d/10-local.rules'? y
```

   b. Run the `start_udev` command to process the rules files.

```
# start_udev
Starting udev:                                    [  OK  ]
```

   c. Use the `ls` command to list the `/dev/xv*` devices.

```
# ls /dev/xv*
/dev/xvda /dev/xvda1 /dev/xvda2 /dev/xvdb /dev/xvdb1 /dev/xvdd
```

      - Notice that the `/dev/xvdb` file now exists.
      - The `/dev/my_disk` still exists but is removed after a reboot.

5. Use the `shutdown -h now` command to shut down **host03**.
   - The next practice expects **host03** to be stopped.

Practices for Lesson 9: OCFS2 and Oracle Clusterware

Chapter 9

Practices for Lesson 9: Clustering with OCFS2

Practices Overview

In these practices, you perform the following:

- Prepare for an OCFS2 configuration.
- Install or upgrade the software required for OCFS2.
- Configure the cluster layout.
- Configure and start the O2CB cluster stack service.
- Create an OCFS2 volume.
- Mount an OCFS2 volume.
- Perform OCFS2 tuning and debugging.

Practice 9-1: Preparing for an OCFS2 Configuration

Overview

In this practice, you perform the following:

- Reconfigure VMs to use shared storage.
- View a private network for VM guests.

Assumptions

- You are the `root` user on **dom0**.
- You completed Practice 2-1 and Practice 2-2 that assigned a static IP address to `eth1` on **host01**.
  - If you did not complete these practices, perform step 1 to assign an IP address to `eth1` on **host01**.
  - If you did complete these practices, go directly to step 2.

Tasks

1. If necessary, assign a static IP address to `eth1` on **host01**.
   a. From **dom0**, use the `ssh` command to log in to **host01**.
      - The `root` password is `oracle`.

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[root@host01 ~]#
```

   b. From **host01**, use the `cd` command to change to the `/etc/sysconfig/network-scripts` directory.

```
# cd /etc/sysconfig/network-scripts
```

   c. Use the `vi` editor to edit the `ifcfg-eth1` file as follows:
      - Only the bold lines should need to be edited, but verify that all lines are as shown.

```
# vi ifcfg-eth1
DEVICE=eth1
HWADDR=00:16:3e:00:02:01
TYPE=Ethernet
UUID=... (leave this as is)
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
IPV6INIT=no
IPADDR=192.168.1.200
NETMASK=255.255.255.0
BROADCAST=192.168.1.255
```

d. Use the `ifup eth1` command to start the `eth1` network interface.

```
# ifup eth1
Determining if ip address 192.168.1.200 is already in use for
device eth1...
```

e. Use the `ifconfig eth1` command to verify that `eth1` has an IP address.

```
# ifconfig eth1
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:01
          inet addr:192.168.1.200 ...
          inet6 addr: ...
          UP BROADCAST RUNNING ...
...
```

f. Use the `exit` command to log off **host01**.

```
# exit
logout
Connection to host01 closed.
```

2. Shut down all VMs.

a. From **dom0**, run the `xm destroy <VM name>` command on all VMs that are currently running.

- The following example runs this command on all three VMs.
- The **host03** VM should already be shut down from Practice 8.

```
# xm destroy host01
# xm destroy host02
# xm destroy host03
```

b. Use the `xm list` command to confirm that all VM domains are shut down.

- When all VMs are shut down, only **dom0** appears in the output of the `xm list` command.

```
# xm list
Name                    ID    Mem  VCPUs      State      Time(s)
Domain-0                 0   2048      2     r-----        281.1
```

3. Change the “hdb” entries in the VM configuration files to shared storage for all three VMs.

a. From **dom0**, use the `grep` command to list the `hdb` entries in the VM guest `vm.cfg` files.

- The quotes in the following example are normal single quotes, not back quotes.

```
[dom0]# grep hdb /OVS/running_pool/host0[123]/vm.cfg
host01/vm.cfg: 'file:/OVS/running_pool/host01/u01.img,hdb,w',
host02/vm.cfg: 'file:/OVS/sharedDisk/physDisk1.img,hdb,w!',
host03/vm.cfg: 'file:/OVS/running_pool/host03/u01.img,hdb,w',
```

- Notice that **host01** and **host03** use `u01.img` in the `/OVS/running_pool/host0[123]/vm.cfg` file for the `hdb` virtual disk.
- Notice that **host02** uses the `/OVS/sharedDisk/physDisk1.img` file for the `hdb` virtual disk.

- Also notice that the **host02** entry uses `w!` to indicate that this is shared storage.
- The `hdb` translates to `xvdb` (`/dev/xvdb`) for the VM guests.

b. Use the `vi` editor to change the "hdb" entry in the **host01** `vm.cfg` file to match the "hdb" entry in the **host02** `vm.cfg` file.

- Make a backup copy of the file beforehand.

```
# cd /OVS/running_pool/host01
# cp vm.cfg vm.cfg.backup
# vi vm.cfg
...
'file:/OVS/running_pool/host01/u01.img,hdb,w',     (old entry)
'file:/OVS/sharedDisk/physDisk1.img,hdb,w!',       (new entry)
...
```

c. Use the `vi` editor to change the "hdb" entry in the **host03** `vm.cfg` file to match the "hdb" entry in the **host02** `vm.cfg` file.

- Make a backup copy of the file beforehand.

```
# cd /OVS/running_pool/host03
# cp vm.cfg vm.cfg.backup
# vi vm.cfg
...
'file:/OVS/running_pool/host03/u01.img,hdb,w',     (old entry)
'file:/OVS/sharedDisk/physDisk1.img,hdb,w!',       (new entry)
...
```

d. Use the `grep` command to list the `hdb` entries in the VM guest `vm.cfg` files.

```
[dom0]# grep hdb /OVS/running_pool/host0[123]/vm.cfg
host01/vm.cfg: 'file:/OVS/sharedDisk/physDisk1.img,hdb,w!',
host02/vm.cfg: 'file:/OVS/sharedDisk/physDisk1.img,hdb,w!',
host03/vm.cfg: 'file:/OVS/sharedDisk/physDisk1.img,hdb,w!',
```

- Do not continue until the **host01** and **host03** entries are identical to the **host02** entry.

4. Start all VMs: **host01**, **host02**, and **host03**.

a. From the `/OVS/running_pool/host01` directory on **dom0**, run the `xm create vm.cfg` command to start the **host01** VM.

```
# cd /OVS/running_pool/host01
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host01 (id=#)
```

b. From the `/OVS/running_pool/host02` directory on **dom0**, run the `xm create vm.cfg` command to start the **host02** VM.

```
# cd /OVS/running_pool/host02
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host02 (id=#)
```

c. From the `/OVS/running_pool/host03` directory on **dom0**, run the `xm create vm.cfg` command to start the **host03** VM.

```
# cd /OVS/running_pool/host03
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host03 (id=#)
```

d. Use the `xm list` command to confirm whether all VM domains are running.

```
# xm list
Name                    ID    Mem  VCPUs      State      Time(s)
Domain-0                 0   2048      2     r-----        304.5
host01                   4   1536      1     -b----         18.7
host02                   2   1536      1     -b----        159.0
host03                   3   1536      1     -b----         13.2
```

- In this example, all VMs are running. The `ID` and `Time(s)` values are examples.
- Do not be concerned if the state of the VMs is '`r`' or '`b`'. Continue to step 5.

5. Log in and view a private network configuration on the VM guests.

a. From **dom0**, use the `ssh` command to log in to **host01**.

- The `root` password is `oracle`.

```
[dom0]# ssh host01
root@host01's password: oracle
Last login...
[root@host01 ~]#
```

b. From **host01**, use the `ifconfig` command to display the network configuration.

```
[host01]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:01
          inet addr:192.0.2.101 ...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:01
          inet addr:192.168.1.200 ...
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 ...
...
```

- Notice that `eth1` is on the private network (`192.168.1`).
- The `eth1` interface was configured to use DHCP in "Practices for Lesson 2: Network Addressing and Name Services."
- In this example, the IP address for `eth1` is `192.168.1.200`, but because DHCP is used, the IP address for your `eth1` interface might be different.
- This practice assumes the IP address for `eth1` on **host01** is `192.168.1.200`. Substitute the correct IP address, if different, as necessary in this practice.

c. From **dom0**, open a new terminal window, `su -` to become the `root` user (password is `oracle`) on **dom0**, and use the `ssh` command to log in to **host02**.

- The `root` password on **host02** is `oracle`.

```
[dom0]$ su -
Password: oracle
[dom0]# ssh host02
root@host02's password: oracle
Last login...
[root@host02 ~]#
```

d. From **host02**, use the `ifconfig` command to display the network configuration.

```
[host02]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:02
          inet addr:192.0.2.102 ...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:02
          inet addr:192.168.1.102 ...
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 ...
...
```

- Notice that `eth1` is on the private network (IP address is `192.168.1.102`).

e. From **dom0**, open a third terminal window, `su -` to become the `root` user (password is `oracle`) on **dom0**, and use the `ssh` command to log in to **host03**.

- The `root` password on **host03** is `oracle`.

```
[dom0]$ su -
Password: oracle
[dom0]# ssh host03
root@host03's password: oracle
Last login...
[root@host03 ~]#
```

f. From **host03**, use the `ifconfig` command to display the network configuration.

```
[host03]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:16:3E:00:01:03
          inet addr:192.0.2.103 ...
...
eth1      Link encap:Ethernet HWaddr 00:16:3E:00:02:03
          inet addr:192.0.2.104 ...
...
eth2      Link encap:Ethernet HWaddr 00:16:3E:00:03:03
          inet addr:192.168.1.103 ...
...
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 ...
...
```

- Notice that `eth2` is on the private network (IP address is `192.168.1.103`).

The O2CB cluster also requires `iptables` to be disabled or modified to allow network traffic on the private network interface.

- This step is performed in "Practice 9-6: Mounting an OCFS2 Volume."

Practice 9-2: Verifying That the Required Software Is Installed

Overview

In this practice, you perform the following:

- Ensure that the OCFS2 package is installed.
- Ensure that the version of the UEK is the same on all VMs.
- Enable kernel settings for O2CB.
- Perform each step on all VMs: **host01**, **host02**, and **host03**.

Assumptions

You are the `root` user on all the VM guests.

Tasks

1. Install the `ocfs2-tools` package on the VM guests.

   Use the `yum install` command to install the `ocfs2-tools` package.

```
# yum install ocfs2-tools
...
Transaction Summary
================================================================
Install       1 Package(s)
Total download size: 488 k
Installed size: 2.5 M
Is this ok [y/N]: y
...
Complete!
```

2. Ensure that the same version of the UEK kernel is installed on the VM guests.

   Use the `uname -r` command to determine the UEK kernel version.

```
# uname -r
3.8.13-16.2.1.el6uek.x86_64
```

   - The `kernel-uek` version is `3.8.13-16.2.1.el6uek` on all VMs.

3. Enable kernel settings for O2CB on the VM guests.

   a. Use the `vi` editor to add the following two entries to the end of the `/etc/sysctl.conf` file.

```
# vi /etc/sysctl.conf
...
kernel.panic_on_oops = 1
kernel.panic = 30
```

b. Use the `sysctl -p` command to cause the changes made to `/etc/sysctl.conf` to take effect immediately.

```
# sysctl -p
...
kernel.panic_on_oops = 1
kernel.panic = 30
```

4. Repeat steps 1, 2, and 3 for the remaining VM guests.

Practice 9-3: Configuring the Cluster Layout

Overview

In this practice, you perform the following:

- Perform all steps from **host01**.
- Create the cluster a layout configuration file on **host01**.
- Copy the cluster layout configuration file from **host01** to **host02** and to **host03**.

Assumptions

You are the `root` user on the **host01** VM guest.

Tasks

1. From **host01**, configure the cluster layout as follows:
   - Cluster name = `mycluster`
   - Number of nodes in the cluster = `3`
   - Heartbeat mode = local
   - First node name = `host01`, IP address = `192.168.1.200`
   - Second node name = `host02`, IP address = `192.168.1.102`
   - Third node name = `host03`, IP address = `192.168.1.103`

   a. Use the `o2cb` command to add the `mycluster` cluster.

   ```
   # o2cb add-cluster mycluster
   ```

   b. Use the `o2cb` command to list the `mycluster` information in the cluster layout configuration file, `/etc/ocfs2/cluster.conf`.

   - You could also use the `cat` command or any shell command (`less`, `more`) that displays the content of a text file.

   ```
   # o2cb list-cluster mycluster
   cluster:
       name = mycluster
       heartbeat_mode = local
       node_count = 0
   ```

   - Notice that the default heartbeat mode is local.

   c. Use the `o2cb` command to add the `host01` node to the `mycluster` cluster.

   ```
   # o2cb add-node mycluster host01
   ```

   d. Use the `o2cb` command to list `mycluster` information.

   - The order of your output might be different from the following example. The `cluster` section might be listed before the `node` section. This is not a problem.

   ```
   # o2cb list-cluster mycluster
   node:
       number = 0
       name = host01
       ip_address = 192.0.2.101
       ip_port = 7777
   ```

```
    cluster = mycluster

cluster:
    name = mycluster
    heartbeat_mode = local
    node_count = 1
```

- Notice that by default, the IP address for the node is obtained from the `/etc/hosts` file.
- This entry needs to be removed and re-added; it is not the address for the private network interface.

e. Use the `o2cb` command to remove the `host01` node from the `mycluster` cluster.

```
# o2cb remove-node mycluster host01
```

f. Use the `o2cb` command to add the `host01` node with IP address `192.168.1.200` to the `mycluster` cluster.

```
# o2cb add-node mycluster host01 --ip 192.168.1.200
```

g. Use the `o2cb` command to add the `host02` node with IP address `192.168.1.102` to the `mycluster` cluster.

```
# o2cb add-node mycluster host02 --ip 192.168.1.102
```

h. Use the `o2cb` command to add the `host03` node with IP address `192.168.1.103` to the `mycluster` cluster.

```
# o2cb add-node mycluster host03 --ip 192.168.1.103
```

i. Use the `o2cb` command to list `mycluster` information.

- The order of your output might be different.

```
# o2cb list-cluster mycluster
node:
    number = 0
    name = host01
    ip_address = 192.168.1.200
    ip_port = 7777
    cluster = mycluster

node:
    number = 1
    name = host02
    ip_address = 192.168.1.102
    ip_port = 7777
    cluster = mycluster

node:
    number = 2
    name = host03
```

```
     ip_address = 192.168.1.103
     ip_port = 7777
     cluster = mycluster

cluster:
     name = mycluster
     heartbeat_mode = local
     node_count = 3
```

j. Use the `o2cb` command to list `mycluster` information with the `--oneline`, argument.

```
# o2cb list-cluster mycluster --oneline
node: 0 host01 192.168.1.200:7777 mycluster
node: 1 host02 192.168.1.102:7777 mycluster
node: 2 host03 192.168.1.103:7777 mycluster
cluster: 3 local mycluster
```

- The cluster layout configuration is complete.

2. Copy the cluster layout configuration file to all nodes in the cluster.
   a. Use the `scp` command to copy `/etc/ocfs2/cluster.conf` from **host01** to **host02**. The `root` user password is `oracle`.

```
# scp /etc/ocfs2/cluster.conf host02:/etc/ocfs2/cluster.conf
The authenticity of host 'host02 (192.0.2.102)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host02,192.0,2,102' (RSA) to the
list of known hosts.
root@host02's password: oracle
scp: /etc/ocfs2/cluster.conf: no such file or directory
```

   - Notice that the copy failed because the `/etc/ocfs2` directory does not exist on the remote host, **host02**.
   - If you are logged on to **host02** in another terminal window, use the `mkdir` command to make the `/etc/ocfs2` directory on **host02**. Then repeat the preceding `scp` command to copy the file.
   - If you are not logged on to **host02**, perform the following step (2b).

   b. Use the `sftp` command to establish a connection from **host01** to **host02**.
   - After connecting to **host02**, make the `/etc/ocfs2` directory on **host02**.
   - After making the directory, use the `put` command to upload the file.
   - After transferring the file, use the `exit` command to close the `sftp` connection.

```
# sftp host02
Connecting to host02...
root@host02's password: oracle
sftp> mkdir /etc/ocfs2
sftp> put /etc/ocfs2/cluster.conf /etc/ocfs2/cluster.conf
```

```
Uploading /etc/ocfs2/cluster.conf to /etc/ocfs2/cluster.conf
/etc/ocfs2/cluster.conf          100% ...
sftp> exit
```

3. Repeat step 2 to copy the cluster layout configuration file from **host01** to **host03**.
   - Substitute **host03** for **host02** in the commands.
   - The cluster layout configuration file now exists on all nodes.

Practice 9-4: Configuring and Starting the O2CB Cluster Stack Service

Overview

In this practice, you will perform the specified step on each VM: **host01**, **host02**, and **host03**.

Assumptions

You are the `root` user on all VM guests.

Tasks

1. Configure the cluster stack service.
   a. Use the `service o2cb` command without any arguments to display usage.

```
# service o2cb
Usage: /etc/init.d/o2cb {start|stop|restart|force-reload|
enable|disable|configure|load|unload|online|offline|force-
offline|status|online-status}
```

   b. Use the `service` command to run the `/etc/init.d/o2cb` initialization script to configure the cluster stack.
      - Enter yes (`y`) to load the O2CB driver on boot.
      - Enter `mycluster` as the cluster to start on boot.
      - Accept the defaults (press Enter) for all other queries.

```
# service o2cb configure
Configuring the O2CB driver.

This will configure the on-boot properties of the O2CB driver.
The following questions will determine whether the driver is
loaded on boot. The current values will be shown in brackets
('[]'). Hitting <ENTER> without typing an answer will keep that
current value. Ctrl-C will abort.
Load O2CB driver on boot (y/n) [n]: y
Cluster stack backing O2CB [o2cb]: ENTER
Cluster to start on boot (Enter "none" to clear) [ocfs2]:
mycluster
Specify heartbeat dead threshold (>=7) [31]: ENTER
Specify network idle timeout in ms (>=5000) [30000]: ENTER
Specify network keepalive delay in ms (>=1000) [2000]: ENTER
Specify network reconnect delay in ms (>=2000) [2000]: ENTER
Writing O2CB configuration: OK
Loading filesystem “configfs”” OK
Mounting configfs filesystem at /sys/kernel/config: OK
Loading stack plugin “o2cb”: OK
Loading filesystem “ocfs2_dlmfs”: OK
Creating directory ‘/dlm’: OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Setting cluster stack "o2cb": OK
```

```
Starting O2CB cluster "mycluster": OK
Setting O2CB cluster timeouts : OK
```

- Notice that each step of the configuration passed (each message is followed by an `OK`).

c. Use the `service o2cb status` command to view the status and settings of the cluster stack.

```
# service o2cb status
Driver for "configfs": Loaded
Filesystem "configfs": Mounted
Stack glue driver: Loaded
Stack plugin "o2cb": Loaded
Driver for "ocfs2_dlmfs": Loaded
Filesystem "ocfs2_dlmfs": Mounted
Checking O2CB cluster mycluster: Online
  Heartbeat dead threshold = 31
  Network idle timeout: 30000
  Network keepalive delay: 2000
  Network reconnect delay: 2000
  Heartbeat mode: Local
Checking O2CB heartbeat: Not active
```

- Notice that the cluster status is `Online`.

2. Repeat step 1b for the remaining VM guests.
   - The O2CB cluster stack needs the same configuration on each VM guest.

Practice 9-5: Creating an OCFS2 Volume

Overview

In this practice, you perform the following:

- Perform all steps from **host01**.
- Create a single partition on the shared disk.
- Create OCFS2 volumes with default settings.
- Create OCFS2 volumes with settings for `mail`, `datafiles`, and `vmstore` usage.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. From **host01**, create a partition on the shared disk.
   - Though it is not required, it is recommended that you create OCFS2 volumes only on partitions because only partitioned volumes can be mounted by label.

   Use the `fdisk` command to create a single partition on `/dev/xvdb`. Use the entire disk for the partition.

```
# fdisk /dev/xvdb
...
WARNING: DOS-compatible mode is deprecated. It’s strongly
recommended to switch off the mode (command ‘c’) and change
display units to sectors (command ‘u’).

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1): ENTER
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1305, default
1305): ENTER
Using default value 1305

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks
```

2. Create different types of OCFS2 volumes on the partitioned disk, `/dev/xvdb1`.
   a. Use the `mkfs.ocfs2` command to create a file system without any options (all defaults).

```
# mkfs.ocfs2 /dev/xvdb1
mkfs.ocfs2 1.8.0
Cluster stack: classic o2cb
Label:
Features: sparse extended-slotmap backup-super unwritten
inline-data strict-journal-super xattr indexed-dirs refcount
discontig-bg
Block size: 4096 (12 bits)
Cluster size: 4096 (12 bits)
Volume size: 10733957120 (2620595 clusters) (2620595 blocks)
Cluster groups: 82 (tail covers 7859 clusters, rest cover 32256
clusters)
Extent allocator size: 4194304 (1 groups)
Journal size: 67108864
Node slots: 4
Creating bitmaps: done
Initializing superblock: done
Writing system files: done
Writing backup superblock: 2 block(s)
Formatting Journals: done
Growing extent allocator: done
Formatting slot map: done
Formatting quota files: done
Writing lost_found: done
mkfs.ocfs2 successful
```

   - Notice the default features: `sparse extended-slotmap backup-super unwritten inline-data strict-journal-super xattr indexed-dirs refcount discontig-bg`
   - Notice the default block size and cluster size: `4 KB`
   - Notice the default number of node slots: `4`
   - Notice the default journal size: `67108864`

   b. Use the `mkfs.ocfs2` command to create a file system with the `-T mail` option.
   - Specify this type when you intend to use the file system as a mail server.
   - Mail servers perform many metadata changes to many small files, which require the use of a large journal.
   - Answer `y` to “Proceed (y/N)”.

```
# mkfs.ocfs2 –T mail /dev/xvdb1
mkfs.ocfs2 1.8.0
Cluster stack: classic o2cb
```

```
Overwriting existing ocfs2 partition.
Proceed (y/N): y
Filesystem Type of mail
...
Journal size: 268435456
...
mkfs.ocfs2 successful
```

- Notice the larger journal size: `268435456`
- All other settings are the defaults.

c. Use the `mkfs.ocfs2` command to create a file system with the `-T datafiles` option.

- Specify this type when you intend to use the file system for database files.
- These file types use fewer fully allocated large files, with fewer metadata changes, and do not benefit from a large journal.
- Answer `y` to “Proceed (y/N)”.

```
# mkfs.ocfs2 -T datafiles /dev/xvdb1
mkfs.ocfs2 1.8.0
Cluster stack: classic o2cb
Overwriting existing ocfs2 partition.
Proceed (y/N): y
Filesystem Type of datafiles
...
Cluster size: 131072 (17 bits)
...
Cluster groups: 3 (tail covers 17381 clusters, rest cover 32256
clusters)
Extent allocator size: 8388608 (2 groups)
Journal size: 33554432
...
mkfs.ocfs2 successful
```

- Notice the differences from the default file system:
  - Cluster size: `131072 (17 bits)`. Default is `4096 (12 bits)`.
  - Cluster groups: `3`. Default is `82`.
  - Extent allocator size: `8388608 (2 groups)`. Default is `4194304 (1 groups)`.
  - Journal size: `33554432`. Default is `67108864`.

d. Use the `mkfs.ocfs2` command to create a file system with the `-T vmstore` option.

- Specify this type when you intend to store virtual machine images.
- These file types are sparsely allocated large files and require moderate metadata updates.
- Answer `y` to “Proceed (y/N)”.

```
# mkfs.ocfs2 –T vmstore /dev/xvdb1
mkfs.ocfs2 1.8.0
Cluster stack: classic o2cb
Overwriting existing ocfs2 partition.
Proceed (y/N): y
Filesystem Type of vmstore
...
Cluster size: 131072 (17 bits)
...
Cluster groups: 3 (tail covers 17381 clusters, rest cover 32256 clusters)
Extent allocator size: 8388608 (2 groups)
Journal size: 134217728
...
mkfs.ocfs2 successful
```

- Notice that the differences from the default file system are similar to the settings for the `datafiles` file system type.
- The exception is the journal size: `134217728`

e. Use the `mkfs.ocfs2` command to create a file system with the label `myvolume`.

- Answer `y` to "Proceed (y/N)".

```
# mkfs.ocfs2 -L "myvolume" /dev/xvdb1
mkfs.ocfs2 1.8.0
Cluster stack: classic o2cb
Overwriting existing ocfs2 partition.
Proceed (y/N): y
Label: myvolume
...
mkfs.ocfs2 successful
```

- Notice that a label is assigned: `myvolume`
- All other settings are defaults.

Practice 9-6: Mounting an OCFS2 Volume

Overview

In this practice, you perform the following:

- Mount the shared OCFS2 volume on each VM: **host01**, **host02**, and **host03**.
- Disable `iptables` to allow the heartbeat between the nodes.
- Copy files to the shared volume from one host and remove the copied files from another host.
- Create a file on the shared volume from one host and edit the file from another host.

Assumptions

You are the `root` user on all VM guests.

Tasks

1. From **host01**, mount the OCFS2 volume.
   a. Use the `mdkir` command to make a mount point, `/u01`, for the OCFS2 volume.

```
[host01]# mkdir /u01
```

   b. Use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.
      - There is a delay in the clustered mount operation.

```
[host01]# mount -L myvolume /u01
```

   c. Use the `service o2cb status` command to display the status of the O2CB heartbeat.

```
[host01]# service o2cb status
...
Checking O2CB heartbeat: Active
```

      - Notice that the heartbeat is active after the volume is mounted.

   d. Use the `cp` command to copy `/boot/vmlinuz*` files to `/u01`. Use the `ls` command to display the contents of `/u01`.

```
[host01]# cp /boot/vmlinuz* /u01
[host01]# ls /u01
lost+found
vmlinuz...
```

      - The `mount` operation is successful on **host01**.
      - You are able to copy files to the OCFS2 volume.

2. From **host02**, mount the OCFS2 volume.
   a. Use the `mdkir` command to make a mount point, `/u01`, for the OCFS2 volume.

```
[host02]# mkdir /u01
```

   b. Use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.

```
[host02]# mount -L myvolume /u01
mount: no such partition found
```

- Notice that the mount fails on **host02** with a “no such partition found” error message.

c. Use the `cat` command to display the `/proc/partitions` file.

```
[host02]# cat /proc/partitions
major minor  #blocks name
 202     0  20971520 xvda
 202     1    512000 xvda1
 202     2  20458496 xvda2
 202    16  10485760 xvdb
 202    48  20971520 xvdd
 252     0  16326656 dm-0
 252     1   4128768 dm-1
```

- Notice that the `xvdb1` partition is not listed.

d. Use the `partprobe` command on `/dev/xvdb` to inform the OS of partition changes.

```
# partprobe /dev/xvdb
```

e. Use the `cat` command to display the `/proc/partitions` file.

```
[host02]# cat /proc/partitions
major minor  #blocks name
 202     0  20971520 xvda
 202     1    512000 xvda1
 202     2  20458496 xvda2
 202    16  10485760 xvdb
 202    17  10485760 xvdb1
 202    48  20971520 xvdd
 252     0  16326656 dm-0
 252     1   4128768 dm-1
```

- Notice that the `xvdb1` partition is now listed.

f. Use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.

- There is a delay in the clustered mount operation.

```
[host02]# mount –L myvolume /u01
mount.ocfs2: Transport endpoint is not connected while mounting
/dev/xvdb1 on /u01. Check ‘dmesg’ for more information on this
error.
```

- Notice that the mount fails on **host02** but with a different error message.

g. Use the `dmesg` command to determine the cause of the failed mount.

```
[host02]# dmesg
...
o2net: Connection to node host01 (num 0) at 192.168.1.200:7777
shutdown, state 7
...
```

- A number of `o2net` and `ocfs2` messages appear in the output of `dmesg`.
- The `o2net` error indicates that the nodes are unable to establish a heartbeat on port `7777`.

h. Use the `service iptables stop` command on both **host01** and **host02** to disable `iptables` and allow communication over port `7777`.

```
[host01]# service iptables stop
iptables: Flushing firewall rules:                        [  OK  ]
iptables: Setting chains to policy ACCEPT: filter         [  OK  ]
iptables: Unloading modules:                              [  OK  ]
[host02]# service iptables stop
iptables: Flushing firewall rules:                        [  OK  ]
iptables: Setting chains to policy ACCEPT: filter         [  OK  ]
iptables: Unloading modules:                              [  OK  ]
```

i. From **host02**, use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.

```
[host02]# mount –L myvolume /u01
```

- The `mount` command succeeded.

j. Use the `service o2cb status` command to display the status of the O2CB heartbeat.

```
[host02]# service o2cb status
...
Checking O2CB heartbeat: Active
```

- Notice that the heartbeat is active after the volume is mounted.

k. Use the `ls` command to display the contents of `/u01`.

```
[host02]# ls /u01
lost+found
vmlinuz...
```

- Notice that the `vmlinuz*` files, which were copied to the volume from **host01**, are present on the volume mounted on **host02**.

3. From **host03**, mount the OCFS2 volume.
   a. Use the `mdkir` command to make a mount point, `/u01`, for the OCFS2 volume.

```
[host03]# mkdir /u01
```

   b. Use the `partprobe` command on `/dev/xvdb` to inform the OS of partition changes.

```
# partprobe /dev/xvdb
```

   c. Use the `service iptables stop` command to disable `iptables` and allow communication over port `7777`.

```
[host03]# service iptables stop
iptables: Flushing firewall rules:                  [  OK  ]
iptables: Setting chains to policy ACCEPT: filter   [  OK  ]
iptables: Unloading modules:                        [  OK  ]
```

   d. Use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.

```
[host03]# mount –L myvolume /u01
```

   - The `mount` command succeeded.

4. Test `write-read` properties of the shared OCFS2 volume.
   a. From **host02**, use the `rm` command to remove the `vmlinuz*` files on `/u01`.

```
[host02]# rm /u01/vmlinuz*
rm: remove regular file '/u01/vmlinuz...? y
...
```

   b. From **host02**, use the `vi` editor to create a file on `/u01` named `host02_test`.
   - Enter the following contents in the `host02_test` file:

```
[host02]# vi /u01/host02_test
This is a test file created from host02.
```

   c. From **host01**, use the `ls` command to list the contents of `/u01`.

```
[host01]# ls /u01
lost+found
host02_test
```

   d. From **host01**, use the `vi` editor to edit the `host02_test` file on `/u01`.
   - Add the second line as follows:

```
[host01]# vi /u01/host02_test
This is a test file created from host02.
This file was edited from host01.
```

e. From **host02**, use the `cat` command to display the contents of the `/u01/host02_test` file.

```
[host02]# cat /u01/host02_test
This is a test file created from host02.
This file was edited from host01.
```

f. From **host03**, use the `cat` command to display the contents of the `/u01/host02_test` file.

```
[host03]# cat /u01/host02_test
This is a test file created from host02.
This file was edited from host01.
```

- Notice that a file created by one host on a shared volume can be edited from another host.

Practice 9-7: Tuning and Debugging OCFS2

Overview

In this practice, you perform the following:

- Query the file system attributes of an OCFS2 volume.
- Enable and disable file system features.
- Detect and fix errors on the file system.
- List OCFS2 volumes and the nodes mounting the OCFS2 volumes.

Assumptions

You are the `root` user on all VM guests.

Tasks

1. Query file system attributes.

   From **host01**, use the `tunefs.ocfs2` command to display the following OCFS2 file system attributes for `/dev/xvdb1`. Format the command to include the attribute names. Print each attribute on a new line.

   - `Block Size (%B)`
   - `Cluster Size (%T)`
   - `Number of Node Slots (%N)`
   - `Volume Label (%V)`
   - `Volume UUID (%U)`
   - `Compatible Features (%M)`
   - `Incompatible Features (%H)`
   - `Read-only Compatible Features (%O)`

   Note: The "\" character at the end of each line is optional. Inserting "\" and pressing **ENTER** provides a new line for the same command-line command. The new line begins with the ">" character which does not need to be typed by you. You can eliminate the "\" character and continuing typing the entire command on a single line if desired.

```
[host01]# tunefs.ocfs2 –Q "Block Size: %B\n\
> Cluster Size: %T\nNumber of Node Slots: %N\n\
> Volume Label: %V\nVolume UUID: %U\n\
> Compatible Features: %M\nIncompatible Features: %H\n\
> Read-Only Compatible Features: %O\n" /dev/xvdb1
Block Size: 4096
Cluster Size: 4096
Number of Node Slots: 4
Volume Label: myvolume
Volume UUID: ...
Compatible Features: backup-super strict-journal-super
Incompatible Features: sparse extended-slotmap inline-data xattr
indexed-dirs refcount discontig-bg
Read-Only Compatible Features: unwritten
```

- The file system features are split up into three categories:
  - `Compatible` is a feature that the file system does not need to fully understand to safely read/write to the volume. An example of this is the `backup-super` feature. Because the `backup-super` blocks are typically not read or written, an older file system can safely mount a volume with this feature enabled.
  - `Incompatible` is a feature that the file system needs to fully understand to read/write to the volume. Most features fall under this category.
  - `Read-only Compatible` is a feature that the file system needs to fully understand to write to the volume. Older software can safely read a volume with this feature enabled. An example of this would be user and group quotas and the `unwritten` feature.
- The following table lists the features, feature category, kernel version, and the `ocfs2-tools` version in which each feature became available.

Features	Category	Kernel Version	Tools Version
`backup-super` – Indicates that the volume has backups of the super block	Compatible	All	ocfs2-tools 1.2
`strict-journal-super` – Indicates that the file system is using version 2 of the JBD super block	Compatible	All	All
`local` – Is enabled when a volume is mounted without a cluster stack. It is also referred to as a local mount.	Incompatible	Linux 2.6.20	ocfs2-tools 1.2
`sparse` – Allows the file system to be efficient in terms of both performance and space usage	Incompatible	Linux 2.6.22	ocfs2-tools 1.4
`inline-data` – Allows the file system to store small files and directories in the inode block itself	Incompatible	Linux 2.6.24	ocfs2-tools 1.4
`extended-slotmap` – Is used to map mounted nodes to system file resources	Incompatible	Linux 2.6.27	ocfs2-tools 1.6
`xattr` – Is an extended attribute.	Incompatible	Linux 2.6.29	ocfs2-tools 1.6

Extended attributes are name:value pairs that can be associated within a file system			
`indexed-dirs` – Is a file system that creates indexes in directories to improve directory entry lookup performance	Incompatible	Linux 2.6.30	ocfs2-tools 1.6
`metaecc` – Enables the file system to compute and validate the checksums for all metadata blocks. It allows for better metadata integrity.	Incompatible	Linux 2.6.29	ocfs2-tools 1.6
`refcount` – Enables the creation of reference counted (refcount) trees that are required to support reflinks	Incompatible	Linux 2.6.32	ocfs2-tools 1.6
`discontig-bg` – Allows the file system to grow the inode and the extent allocators even when there is no large contiguous free chunk available	Incompatible	Linux 2.6.35	ocfs2-tools 1.6
`clusterinfo` – Enables storing the cluster stack information in the super block	Incompatible	Linux 2.6.37	ocfs2-tools 1.8
`unwritten` – Allows applications to instantly pre-allocate large extents within a file	Read-Only Compatible	Linux 2.6.23	ocfs2-tools 1.4
`usrquota` – Allows the file system to track the amount of space and number of inodes for each user	Read-Only Compatible	Linux 2.6.29	ocfs2-tools 1.6
`grpquota` – Allows the file system to track	Read-Only Compatible	Linux 2.6.29	ocfs2-tools 1.6

the amount of space and number of inodes for each group			

- The file system has all the available file system features enabled with the following exceptions: `local`, `metaecc`, `usrquota`, and `grpquota`.

2. Enable and disable features on the file system.
   a. Use the `tunefs.ocfs2` command to enable the `metaecc` and `usrquota` features on `/dev/xvdb1`.

```
[host01]# tunefs.ocfs2 --fs-features=metaecc,usrquota /dev/xvdb1
tunefs.ocfs2: Trylock failed while opening device "/dev/xvdb1"
```

   - Notice that the command fails.

   b. Use the `umount` command to unmount `/u01`, and then repeat the previous command.
   - There is a slight delay in unmounting the file system.

```
[host01]# umount /u01
[host01]# tunefs.ocfs2 --fs-features=metaecc,usrquota /dev/xvdb1
tunefs.ocfs2: Trylock failed while opening device "/dev/xvdb1"
```

   - Notice that the command still fails.

   c. Use the `umount` command to unmount `/u01` on the other nodes—**host02** and **host03**—and then repeat the previous `tunefs.ocfs2` command on **host01**.

```
[host02]# umount /u01
[host03]# umount /u01
[host01]# tunefs.ocfs2 --fs-features=metaecc,usrquota /dev/xvdb1
```

   - The command succeeds when the volume was unmounted on all nodes.
   - The volume needs to be unmounted across the cluster before enabling or disabling any features.

   d. Repeat the `tunefs.ocfs2` command that was executed in step 1.

```
[host01]# tunefs.ocfs2 –Q "Block Size: %B\nCluster Size: %T\nNumber of Node Slots: %N\nVolume Label: %V\nVolume UUID: %U\nCompatible Features: %M\nIncompatible Features: %H\nRead-Only Compatible Features: %O\n" /dev/xvdb1
Block Size: 4096
Cluster Size: 4096
Node Slots: 4
Volume Label: myvolume
Volume UUID: ...
Compatible Features: backup-super strict-journal-super
Incompatible Features: sparse extended-slotmap inline-data metaecc xattr indexed-dirs refcount discontig-bg
Read-Only Compatible Features: unwritten usrquota
```

   - Notice that the new features, `metaecc` and `usrquota`, are enabled (in **bold**).

e. Use the `tunefs.ocfs2` command to disable the `metaecc` and `usrquota` features on `/dev/xvdb1`.

```
[host01]# tunefs.ocfs2 --fs-features=nometaecc,nousrquota
/dev/xvdb1
tunefs.ocfs2: I/O error on channel while closing device
"/dev/xvdb1"
```

- Notice that an error occurs but the features are removed. This is a known bug.

f. Repeat the `tunefs.ocfs2` command that was executed in step 1.

```
[host01]# tunefs.ocfs2 –Q "Block Size: %B\nCluster Size:
%T\nNumber of Node Slots: %N\nVolume Label: %V\nVolume UUID:
%U\nCompatible Features: %M\nIncompatible Features: %H\nRead-
Only Compatible Features: %O\n" /dev/xvdb1
Block Size: 4096
Cluster Size: 4096
Node Slots: 4
Volume Label: myvolume
Volume UUID: ...
Compatible Features: backup-super strict-journal-super
Incompatible Features: sparse extended-slotmap inline-data xattr
indexed-dirs refcount discontig-bg
Read-Only Compatible Features: unwritten
```

3. Detect any errors on the file system.

Use the `fsck.ocfs2` command to detect and fix errors on `/dev/xvdb1`.

```
[host01]# fsck.ocfs2 –f /dev/xvdb1
fsck.ocfs2 1.8.0
Checking OCFS2 filesystem in /dev/xvdb1:
Label: myvolume
UUID: ...
Number of blocks: 2620595
Block size: 4096
Number of clusters: 2620595
Cluster size: 4096
Number of slots: 4
/dev/xvdb1 was run with -f, check forced.
Pass 0a: Checking cluster allocation chains
Pass 0b: Checking inode allocation chains
Pass 0c: Checking extent block allocation chains
Pass 1: Checking inodes and blocks.
Pass 2: Checking directory entries.
Pass 3: Checking directory connectivity.
Pass 4a: checking for orphaned inodes
Pass 4b: Checking inodes link counts.
All passes succeeded.
```

4. List mounted volumes.
   a. Use the `mounted.ocfs2` command to list all OCFS2 volumes.

```
[host01]# mounted.ocfs2 -d
Device       Stack  Cluster  F  UUID                 Label
/dev/xvdb1   o2cb               ...                  myvolume
```

   b. Use the `mounted.ocfs2` command to list the nodes that are currently mounting each OCFS2 volume.

```
[host01]# mounted.ocfs2 -f
Device       Stack  Cluster  F  Nodes
/dev/xvdb1   o2cb               Not mounted
```

   c. Use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.

```
[host01]# mount -L myvolume /u01
```

   d. Use the `mounted.ocfs2` command to list the nodes that are currently mounting each OCFS2 volume.

```
[host01]# mounted.ocfs2 -f
Device       Stack  Cluster  F  Nodes
/dev/xvdb1   o2cb               host01
```

   e. From **host02**, use the `mount` command to mount the OCFS2 volume by label `myvolume` on the `/u01` mount point.

```
[host02]# mount -L myvolume /u01
```

   f. Use the `mounted.ocfs2` command, from either **host01** or **host02**, to list the nodes that are currently mounting each OCFS2 volume.

```
# mounted.ocfs2 -f
Device       Stack  Cluster  F  Nodes
/dev/xvdb1   o2cb               host01, host02
```

5. Restore the VMs to their original state.
   a. Use the `umount` command to unmount `/u01` from the **host01** and **host02** VMs.

```
[host01]# umount /u01
[host02]# umount /u01
```

   b. Use the `service` command to stop the `o2cb` service on all VMs.

```
[host01]# service o2cb stop
...
[host02]# service o2cb stop
...
[host03]# service o2cb stop
...
```

c. Use the `chkconfig` command to disable the `o2cb` service from starting at reboot on all VMs.

```
[host01]# chkconfig o2cb off
[host02]# chkconfig o2cb off
[host03]# chkconfig o2cb off
```

d. Use the `shutdown` command to bring down all the VMs.

```
[host01]# shutdown -h now
...
[host02]# shutdown -h now
...
[host03]# shutdown -h now
...
```

e. From **dom0**, run the `xm list` command to confirm that all VM domains are shut down.

- When all VMs are shut down, only **dom0** appears in the output of the `xm list` command.

```
[dom0]# xm list
Name                    ID    Mem  VCPUs      State      Time(s)
Domain-0                 0   2048      2     r-----        281.1
```

f. From the `/OVS/running_pool/host01` directory on **dom0**, use the `cp` command to restore the **host01** `vm.cfg` file from the backup made in Practice 9-1.

```
[dom0]# cd /OVS/running_pool/host01
[dom0]# cp vm.cfg.backup vm.cfg
cp: overwrite 'vm.cfg'? y
```

g. From the `/OVS/running_pool/host03` directory on **dom0**, use the `cp` command to restore the **host03** `vm.cfg` file from the backup made in Practice 9-1.

```
[dom0]# cd /OVS/running_pool/host03
[dom0]# cp vm.cfg.backup vm.cfg
cp: overwrite 'vm.cfg'? y
```

h. From the `/OVS/running_pool/host01` directory on **dom0**, run the `xm create vm.cfg` command to start the **host01** VM.

```
[dom0]# cd /OVS/running_pool/host01
[dom0]# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host01 (id=#)
```

i. From the `/OVS/running_pool/host02` directory on **dom0**, run the `xm create vm.cfg` command to start the **host02** VM.

```
[dom0]# cd /OVS/running_pool/host02
[dom0]# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host02 (id=#)
```

j. From the `/OVS/running_pool/host03` directory on **dom0**, run the `xm create vm.cfg` command to start the **host03** VM.

```
[dom0]# cd /OVS/running_pool/host03
[dom0]# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host03 (id=#)
```

k. From **dom0**, run the `xm list` command to confirm whether all VM domains are running.

```
# xm list
Name                    ID    Mem  VCPUs       State      Time(s)
Domain-0                 0   2048      2      r-----        304.5
host01                   4   1536      1      -b----         18.7
host02                   2   1536      1      -b----        159.0
host03                   3   1536      1      -b----         13.2
```

- In this example, all VMs are running. The `ID` and `Time(s)` values are examples.

Practices for Lesson 10: iSCSI and Multipathing

Chapter 10

Practices for Lesson 10: iSCSI and Multipathing

Practices Overview

In these practices, you configure the following:

- An iSCSI target
- An iSCSI initiator
- iSCSI multipathing

Practice 10-1: Exploring a Configured iSCSI Server

Overview

In this practice, you:

- Perform all steps from **dom0**
- Explore the iSCSI target configuration
- Start the iSCSI target daemon and show configured targets

Assumptions

- **Dom0** is already configured as an iSCSI server.
- You are the `root` user on **dom0**.

Tasks

1. Explore the iSCSI target configuration on **dom0**.
   a. Use the `rpm` command to query the `scsi-target-utils` package.

```
# rpm –qa scsi-target-utils
scsi-target-utils-1.0.14-1.el5
```

   - In this example, version `1.0.14` is installed on **dom0**.
   - As of this writing, version `1.0.24` is available for Oracle Linux 6.5. Version `1.0.24` includes a man page for `target.conf`.
   - In this course, do not update this package or any packages on **dom0**.

   b. Use the `rpm` command to list the files in the `scsi-target-utils` package.

```
# rpm –ql scsi-target-utils
/etc/rc.d/init.d/tgtd
/etc/sysconfig/tgtd
/etc/tgt/targets.conf
/usr/sbin/tgt-admin
/usr/sbin/tgt-setup-lun
/usr/sbin/tgtadm
/usr/sbin/tgtd
...
```

   - Notice that the configuration file, `/etc/tgt/targets.conf`, is included in the package.
   - Notice the iSCSI target daemon, `tgtd`, and the service startup script, `/etc/rc.d/init.d/tgtd`.
   - Notice the iSCSI target utilities in the `/user/sbin` directory: `tgt-adm`, `tgt-setup-lun`, and `tgtadm`.

   c. Use the `file` command to display the file types of the iSCSI target files.

```
# file /usr/sbin/tgt*
/usr/sbin/tgtadm:         ELF 32-bit LSB executable ...
/usr/sbin/tgt-admin:      perl script text executable
/usr/sbin/tgtd:           ELF 32-bit LSB executable ...
/usr/sbin/tgt-setup-lun:  Bourne-Again shell script ...
```

- Notice that `tgtadm` is a binary executable.
- Notice that `tgt-admin` is an executable Perl script.
- Notice that `tgt-setup-lun` is an executable Bash shell script.

d. Use the `cat` command to view the contents of the iSCSI target configuration file, `/etc/tgt/targets.conf`.

```
# cat /etc/tgt/targets.conf
...
<target iqn.2013-03.com.example.mypc:tgt1>
    backing-store /OVS/sharedDisk/physDisk1.img
    backing-store /OVS/sharedDisk/physDisk2.img
    write-cache off
</target>
<target iqn.2013-03.com.example.mypc:tgt2>
    backing-store /OVS/sharedDisk/physDisk3.img
    backing-store /OVS/sharedDisk/physDisk4.img
    write-cache off
</target>
```

- In this example, two targets are configured, each with two LUNs.
- The LUNs are defined with the `backing-store` directive rather than the `direct-store` directive because the LUNs are regular files and not local SCSI devices.
- The `write-cache` directive is set to `off`; the default is `on`.

2. Start the iSCSI target daemon and show configured targets.

a. Use the `service` command to start the `tgtd` daemon.

```
# service tgtd start
Starting SCSI target daemon: Starting target framework daemon
```

- If the daemon fails to start, run the command again.

b. Use the `tgt-admin` command to show all targets.

```
# tgt-admin -s
Target 1: iqn.2013-03.com.example.mypc:tgt1
    System information:
        Driver: iscsi
        State: ready
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: IET     00010000
            SCSI SN: beaf10
            Size: 0 MB, Block size: 1
            Online: Yes
            Removable media: No
```

```
            Readonly: No
            Backing store type: null
            Backing store path: None
            Backing store flags:
        LUN: 1
            Type: disk
            SCSI ID: IET     00010001
            SCSI SN: beaf11
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk1.img
            Backing store flags:
        LUN: 2
            Type: disk
            SCSI ID: IET     00010002
            SCSI SN: beaf12
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk2.img
            Backing store flags:
    Account information:
    ACL information:
        ALL
Target 2: iqn.2013-03.com.example.mypc:tgt2
    System information:
        Driver: iscsi
        State: ready
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: IET     00020000
            SCSI SN: beaf20
            Size: 0 MB, Block size: 1
            Online: Yes
            Removable media: No
```

```
            Readonly: No
            Backing store type: null
            Backing store path: None
            Backing store flags:
        LUN: 1
            Type: disk
            SCSI ID: IET     00020001
            SCSI SN: beaf21
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk3.img
            Backing store flags:
        LUN: 2
            Type: disk
            SCSI ID: IET     00020002
            SCSI SN: beaf22
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk4.img
            Backing store flags:
    Account information:
    ACL information:
        ALL
```

- Notice that both targets specified in the configuration file are available.
- Each target has a `LUN 0` of type `controller`.
- Each target has a `LUN 1` and `LUN 2`, each 10737 MB in size.
- The LUNs are allowing connections from ALL initiators (`ACL information = ALL`).

Practice 10-2: Modifying the iSCSI Server (Target) Configuration

Overview

In this practice, you:

- Perform all steps from **dom0**
- Add a new iSCSI target and LUN from the command line
- Restrict access to specific initiator IP addresses from the command line
- Modify the configuration file to create a persistent target and LUN

Assumptions

- **Dom0** is configured as an iSCSI server.
- You are the `root` user on **dom0**.

Tasks

1. Add a new iSCSI target and LUN from the command line.
   a. Use the `tgtadm` command to add a new target using the following parameters:
      - Target ID = `3`
      - Target Name = `iqn.2012-11.com.example.mypc:tgt3`

```
# tgtadm –L iscsi -o new -m target --tid 3 --targetname
iqn.2012-11.com.example.mypc:tgt3
```

   b. Use the `tgtadm` command to show all targets.
      - Note that you could also use the "`tgt-admin -s`" command to show all targets.

```
# tgtadm –o show –m target
...
Target 3: iqn.2012-11.com.example.mypc:tgt3
    System information:
        Driver: iscsi
        State: ready
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: IET     00030000
            SCSI SN: beaf30
            Size: 0 MB, Block size: 1
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: null
            Backing store path: None
            Backing store flags:
    Account information:
    ACL information:
```

- Notice that the new target has only `LUN 0` (`controller`).

c. Use the `tgtadm` command to add a LUN to target ID `3` using the following parameters:
   - Target ID = `3`
   - LUN number = `1`
   - Backing-store = `/OVS/sharedDisk/physDisk5.img`

```
# tgtadm -o new -m logicalunit --tid 3 --lun 1 --backing-store
/OVS/sharedDisk/physDisk5.img
```

d. Use the `tgt-admin -s` command to show all targets.

```
# tgt-admin -s
...
Target 3: iqn.2012-11.com.example.mypc:tgt3
...
        LUN: 1
            Type: disk
            SCSI ID: IET     00010001
            SCSI SN: beaf31
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk5.img
            Backing store flags:
    Account information:
    ACL information:
```

- Notice the new `LUN 1` (`disk`) for target `3`.

2. Restrict access to specific initiator IP addresses from the command line.

a. Use the `tgtadm` command to add IP addresses `192.0.2.103` and `192.0.2.104` to the access list for target ID `3`.

```
# tgtadm -o bind -m target --tid 3 --initiator-address
192.0.2.103
# tgtadm -o bind -m target --tid 3 --initiator-address
192.0.2.104
```

b. Use the `tgt-admin -s` command to show all targets.

```
# tgt-admin -s
...
Target 3: iqn.2012-11.com.example.mypc:tgt3
...
        LUN: 1
            Type: disk
            SCSI ID: IET     00010001
            SCSI SN: beaf31
```

```
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk5.img
            Backing store flags:
    Account information:
    ACL information:
        192.0.2.103
        192.0.2.104
```

- Notice that the ACL for target `3` now specifies IP addresses `192.0.2.103` and `192.0.2.104`.
- Only initiators with these IP addresses have access to the target.

3. Modify the iSCSI target configuration file to make available targets persistent.
   - Configuring iSCSI targets with the `tgtadm` command is not persistent.
   a. Use the `service` command to stop and start the `tgtd` daemon.

```
# service tgtd stop
Stopping SCSI target daemon: Stopping target framework daemon
                                                       [  OK  ]
# service tgtd start
Starting SCSI target daemon: Starting target framework daemon
```

   - Stopping and starting seems to work better than restarting.

   b. Use the `tgt-admin -s` command to show all targets and `grep` for `Target`.

```
# tgt-admin –s | grep Target
Target 1: iqn.2013-03.com.example.mypc:tgt1
Target 2: iqn.2013-03.com.example.mypc:tgt2
```

   - Only those targets and LUNs that are configured in `/etc/tgt/targets.conf` exist.

   c. Use the `vi` editor to add the following lines to the end of the `/etc/tgt/targets.conf` file:

```
# vi /etc/tgt/targets.conf
<target iqn.2012-11.com.example.mypc:tgt3>
    backing-store /OVS/sharedDisk/physDisk5.img
    initiator-address 192.0.2.103
    initiator-address 192.0.2.104
    write-cache off
</target>
```

   - These lines add a new target, `3`, define a single LUN by using the `/OVS/sharedDisk/physDisk5.img` file, and restrict access to initiators with the IP addresses of `192.0.2.103` and `192.0.2.104`.

   d. Use the `service` command to stop and start the `tgtd` daemon.

```
# service tgtd stop
Stopping SCSI target daemon: Stopping target framework daemon
                                                     [  OK  ]
# service tgtd start
Starting SCSI target daemon: Starting target framework daemon
```

e. Use the `tgt-admin -s` command to show all targets.

- The order of the targets might be different on your system.

```
# tgt-admin -s
Target 1: iqn.2013-03.com.example.mypc:tgt1
...
Target 2: iqn.2013-03.com.example.mypc:tgt2
...
Target 3: iqn.2012-11.com.example.mypc:tgt3
    System information:
        Driver: iscsi
        State: ready
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: IET     00030000
            SCSI SN: beaf30
            Size: 0 MB, Block size: 1
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: null
            Backing store path: None
            Backing store flags:
        LUN: 1
            Type: disk
            SCSI ID: IET     00010001
            SCSI SN: beaf31
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /OVS/sharedDisk/physDisk5.img
            Backing store flags:
    Account information:
    ACL information:
```

```
        192.0.2.103
        192.0.2.104
```

- In this example the new target, Target `3`, is now persistent with one LUN and with access control information.

Practice 10-3: Configuring an iSCSI Client (Initiator)

Overview

In this practice, you:

- Perform all steps from **host03**
- Install the `iscsi-initiator-utils` package
- Create and delete iSCSI interfaces
- Discover iSCSI targets using the SendTargets discovery method
- Query the Open-iSCSI persistent database
- Observe the settings in the iSCSI initiator configuration file
- Establish a TCP session between the target and the initiator
- Verify the usability of the iSCSI device

Assumptions

- You are the `root` user on **dom0**.
- The **host03** VM is running.

Tasks

1. Use the `ssh` command to log in to **host03**.
   - The `root` password is `oracle`.

```
# ssh host03
root@host03's password: oracle
[root@host03 ~]#
```

2. Install and explore the iSCSI initiator package.
   Run the `yum install` command to install the `iscsi-initiator-utils` package.
   - The package might already be installed on **host03**.

```
# yum install iscsi-initiator-utils
...
===========================================================
Package                  Arch     Version
===========================================================
Installing:
 iscsi-initiator-utils   x86_64   6.2.0.873-10.0.1.el6 ...
...
Is this ok [y/N]: y
...
Complete!
```

3. Create iSCSI interfaces.
   a. Use the `iscsiadm` command to create two iSCSI interfaces: `iface0` and `iface1`.

```
# iscsiadm -m iface -I iface0 -o new
New interface iface0 added
# iscsiadm -m iface -I iface1 -o new
New interface iface1 added
```

b. Use the `ls` command to view the `ifaces` table in the Open-iSCSI persistent database, `/var/lib/iscsi`.

```
# ls /var/lib/iscsi/ifaces
iface0  iface1
```

c. Use the `iscsiadm` command to view the interface properties.

```
# iscsiadm –m iface –I iface0
# BEGIN RECORD 6.2.0-873.10.el6
iface.iscsi_ifacename = iface0
iface.net_ifacename = <empty>
iface.ipaddress = <empty>
iface.hwaddress = <empty>
iface.transport_name = tcp
iface.initiatorname = <empty>
...
# iscsiadm –m iface –I iface1
# BEGIN RECORD 6.2.0-873.10.el6
iface.iscsi_ifacename = iface1
iface.net_ifacename = <empty>
iface.ipaddress = <empty>
iface.hwaddress = <empty>
iface.transport_name = tcp
iface.initiatorname = <empty>
...
```

d. Use the `iscsiadm` command to bind the iSCSI interfaces to the network interfaces. Bind `iface0` to `eth0`. Bind `iface1` to `eth1`.

```
# iscsiadm -m iface -o update -I iface0 –n iface.net_ifacename -
v eth0
iface0 updated.
# iscsiadm -m iface -o update -I iface1 –n iface.net_ifacename -
v eth1
iface1 updated.
```

e. View the interface properties again and notice that the value of `iface.net_ifacename` has been updated.

```
# iscsiadm –m iface –I iface0
# BEGIN RECORD 6.2.0-873.10.el6
iface.iscsi_ifacename = iface0
iface.net_ifacename = eth0
...
# iscsiadm –m iface –I iface1
# BEGIN RECORD 6.2.0-873.10.el6
iface.iscsi_ifacename = iface1
iface.net_ifacename = eth1
```

```
...
# iscsiadm -m iface
default tcp,<empty>,<empty>,<empty>,<empty>
iser iser,<empty>,<empty>,<empty>,<empty>
iface0 tcp,<empty>,<empty>,eth0,<empty>
iface1 tcp,<empty>,<empty>,eth1,<empty>
```

4. Delete the iSCSI interfaces.
   - Delete the iSCSI interfaces because they cause problems in an Oracle VM Server for x86 environment.
   - The purpose of step 3 was to demonstrate how to create iSCSI interfaces.

   a. Use the `iscsiadm` command to delete the `iface0` iSCSI interface.

```
# iscsiadm -m iface -I iface0 -o delete
iface0 unbound and deleted.
# ls -R /var/lib/iscsi
ifaces  isns  nodes  send_targets  slp  static
/var/lib/iscsi/ifaces:
iface1
...
```

   - Notice the `iface0` record is removed.

   b. Use the `iscsiadm` command to delete the `iface1` iSCSI interface.

```
# iscsiadm -m iface -I iface1 -o delete
iface1 unbound and deleted.
```

5. Discover the targets.
   - Sample output is provided. Targets might be discovered in a different order on your system.

   a. Use the `iscsiadm` command to discover iSCSI targets by using the SendTargets discovery method from IP address (portal) `192.0.2.1`.

```
# iscsiadm -m discovery -t st -p 192.0.2.1
Starting iscsid:                                [  OK  ]
192.0.2.1:3260,1 iqn.2013-03.com.example.mypc:tgt1
192.0.2.1:3260,1 iqn.2013-03.com.example.mypc:tgt2
192.0.2.1:3260,1 iqn.2012-11.com.example.mypc:tgt3
```

   - This command causes the `iscsid` daemon to start.
   - Three targets are discovered, as configured in the `/etc/tgt/targets.conf` file on the iSCSI target (`192.0.2.1`).

   b. The iSCSI targets are offered on TCP and UDP port `3260`. Use the `grep` command to search for `3260` in the `/etc/services` file.

```
# grep 3260 /etc/services
iscsi-target    3260/tcp          # iSCSI port
iscsi-target    3260/udp          # iSCSI port
```

6. Query the Open-iSCSI persistent database and the iSCSI initiator configuration file.
   a. View the `nodes` table and the `send_targets` table in the Open-iSCSI persistent database, `/var/lib/iscsi`.

```
# ls /var/lib/iscsi/nodes
iqn.2013-03.com.example.mypc:tgt1
iqn.2013-03.com.example.mypc:tgt2
iqn.2012-11.com.example.mypc:tgt3
# ls /var/lib/iscsi/send_targets
192.0.2.1,3260
```

   b. Use the `iscsiadm` command to query the `send_targets` table in the persistent database.

```
# iscsiadm –m discoverydb –t st –p 192.0.2.1
# BEGIN RECORD 6.2.0-873.10.el6
discovery.startup = manual
discovery.type = sendtargets
discovery.sendtargets.address = 192.0.2.1
discovery.sendtargets.port = 3260
discovery.sendtargets.auth.authmethod = None
discovery.sendtargets.auth.username = <empty>
discovery.sendtargets.auth.password = <empty>
discovery.sendtargets.auth.username_in = <empty>
discovery.sendtargets.auth.password_in = <empty>
discovery.sendtargets.timeo.login_timeout = 15
discovery.sendtargets.use_discoveryd = No
discovery.sendtargets.discoveryd_poll_inval = 30
discovery.sendtargets.repoen_max = 5
discovery.sendtargets.timeo.auth_timeout = 45
discovery.sendtargets.timeo.active_timeout = 30
discovery.sendtargets.iscsi.MaxRecvDataSegmentLength = 32768
# END RECORD
```

   - Much of the information in the database is derived from the settings in the iSCSI initiator configuration file, `/etc/iscsi/iscsid.conf`.

   c. Use the `grep` command to search for `auth.authmethod` in `/etc/iscsi/iscsid.conf`.

```
# grep auth.authmethod /etc/iscsi/iscsid.conf
# To enable CHAP authentication set node.session.auth.authmethod
#node.session.auth.authmethod = CHAP
# set discovery.sendtargets.auth.authmethod to CHAP. The default
is None.
#discovery.sendtargets.auth.authmethod = CHAP
```

   - Challenge-Handshake Authentication Protocol (CHAP) allows iSCSI targets and initiators to prove their identity to each other.
   - CHAP prevents clear-text passwords from appearing on the network.

7. Establish a TCP session between the iSCSI target and the initiator.
   - Log in to establish a session.
   - iSCSI target LUNs are not available until a session is established.

   a. Use the `iscsiadm` command to view active sessions.

```
# iscsiadm -m session
iscsiadm: No active sessions
```

   b. Use the `fdisk` command to display the current devices in `/dev`.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *       1       64      512000   83   Linux
/dev/xvda2          64     1567   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
/dev/xvdb1           1      132    1060258+  83   Linux
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes
Disk /dev/mapper/vg_host01-lv-root: 11.1 GB, 11068768256 bytes
Disk /dev/mapper/vg_host01-lv-swap: 1287 MB, 1287651328 bytes
```

   - This example shows three virtual disks, `/dev/xvda`, `/dev/xvdb`, and `/dev/xvdd`.
   - This example also shows LVM volumes for the `root` and `swap` partitions.

   c. Use the `iscsiadm` command to log in and establish a session.

```
# iscsiadm -m node -l
Login to [iface: default, target: iqn.2013-03.com.example.mypc:tgt1, portal: 192.0.2.1:3260] (multiple)
Login to [iface: default, target: iqn.2012-11.com.example.mypc:tgt3, portal: 192.0.2.1:3260] (multiple)
Login to [iface: default, target: iqn.2013-03.com.example.mypc:tgt2, portal: 192.0.2.1:3260] (multiple)
Login to [iface: default, target: iqn.2013-03.com.example.mypc:tgt1, portal: 192.0.2.1:3260] successful.
Login to [iface: default, target: iqn.2012-11.com.example.mypc:tgt3, portal: 192.0.2.1:3260] successful
Login to [iface: default, target: iqn.2013-03.com.example.mypc:tgt2, portal: 192.0.2.1:3260] successful
```

   - This command logged in to all the targets (`tgt1`, `tgt2`, `tgt3`) offered by the server.

   d. Use the `iscsiadm` command to view active sessions.

```
# iscsiadm -m session
tcp: [1] 192.0.2.1:3260,1 iqn.2013-03.com.example.mypc:tgt1
tcp: [2] 192.0.2.1:3260,1 iqn.2012-11.com.example.mypc:tgt3
tcp: [3] 192.0.2.1:3260,1 iqn.2013-03.com.example.mypc:tgt2
```

   - Three sessions are active, one session per target.

e. Use the `fdisk` command to display available SCSI (`sd`) devices in `/dev`.

```
# fdisk -l | grep /dev/sd
Disk /dev/sda: 10.7 GB, 10737418240 bytes
Disk /dev/sdc: 10.7 GB, 10737418240 bytes
Disk /dev/sde: 10.7 GB, 10737418240 bytes
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
Disk /dev/sdd: 10.7 GB, 10737418240 bytes
/dev/sdd1              1        1305    10482381  83  Linux
```

- All five LUNs (two from target `1`, two from target `2`, and one from target `3`) are available as SCSI block devices.

f. To determine which LUNs are associated with which target, use the `iscsiadm` command to view active sessions with `<printlevel>` `3`.

```
# iscsiadm -m session -P 3
...
Target: iqn.2013-03.com.example.mypc:tgt2
...
          ************************
          Attached SCSI devices:
          ************************
          Host Number: 2 State: running
          scsi12 Channel 00 Id 0 Lun:0
          scsi12 Channel 00 Id 0 Lun:1
               Attached scsi disk sdb         State: running
          scsi12 Channel 00 Id 0 Lun:2
               Attached scsi disk sdc         State: running
...
Target: iqn.2012-11.com.example.mypc:tgt3
...
          ************************
          Attached SCSI devices:
          ************************
          Host Number: 3 State: running
          scsi13 Channel 00 Id 0 Lun:0
          scsi13 Channel 00 Id 0 Lun:1
               Attached scsi disk sda         State: running
...
Target: iqn.2013-03.com.example.mypc:tgt1
...
          ************************
          Attached SCSI devices:
          ************************
          Host Number: 4 State: running
```

```
          scsi14 Channel 00 Id 0 Lun:0
          scsi14 Channel 00 Id 0 Lun:1
               Attached scsi disk sdd        State: running
          scsi14 Channel 00 Id 0 Lun:2
               Attached scsi disk sde        State: running
```

- In this example, `/dev/sdb` and `/dev/sdc` are target `2` LUNs.
- `/dev/sda` is the target `3` LUN.
- `/dev/sdd` and `/dev/sde` are target `1` LUNs.

8. Log out to close a session.
   a. Use the `iscsiadm` command to log out from targets `1` and `2`.

```
# iscsiadm -m node --targetname iqn.2013-03.com.example.mypc:tgt1 -p 192.0.2.1:3260 –u
Logging out of session [sid:3, target: iqn.2013-03.com.example.mypc:tgt1, portal: 192.0.2.1:3260]
Logout of [sid: 3, target: iqn.2013-03.com.example.mypc:tgt1, portal: 192.0.2.1:3260] successful.
# iscsiadm -m node --targetname iqn.2013-03.com.example.mypc:tgt2 –p 192.0.2.1:3260 –u
Logging out of session [sid:1, target: iqn.2013-03.com.example.mypc:tgt2, portal: 192.0.2.1:3260]
Logout of [sid: 1, target: iqn.2013-03.com.example.mypc:tgt2, portal: 192.0.2.1:3260] successful.
```

   b. Use the `iscsiadm` command to view active sessions.

```
# iscsiadm –m session
tcp: [2] 192.0.2.1:3260,1 iqn.2012-11.com.example.mypc:tgt3
```

- Only one session is now active, for target `3`.

   c. Use the `fdisk` command to display available SCSI (`sd`) devices in `/dev`.

```
# fdisk –l | grep /dev/sd
Disk /dev/sda: 10.7 GB, 10737418240 bytes
```

- Only one LUN is available, the LUN from target `3`.
- In this example, the LUN is represented as SCSI block device `/dev/sda`. The device name might be different on your system.

   d. Use the `tail` command to view the `/var/log/messages` file.

```
# tail /var/log/messages
...
<date> host03 kernel: scsi3 : iSCSI Initiator over TCP/IP
...
```

- Notice the iSCSI Initiator over TCP/IP information written to the log file.

9. Verify the usability of the iSCSI device.
   - This example assumes the device name is `/dev/sda`. Use the device name as given in step 8c.

   a. Use the `fdisk` command to create a partition on `/dev/sda` with the following parameters:
      - Primary partition
      - Partition number `1`
      - First cylinder `1`
      - Last cylinder `+2560`

```
# fdisk /dev/sda
...
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1): ENTER
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240): +2560
Command (m for help): w
...
```

   b. Use the `mkfs` command to create an ext4 file system on `/dev/sda1`.

```
# mkfs -t ext4 /dev/sda1
...
```

   c. Use the `mkdir` and `mount` (with the `_netdev` option) commands to mount the file system on `/iscsi_dev`.

```
# mkdir /iscsi_dev
# mount /dev/sda1 -o _netdev /iscsi_dev
```

      - Because it is a network device, be sure to use the `_netdev` option when mounting.

   d. Use the `mount` command to display mounted file systems.

```
# mount
...
/dev/sda1 on /iscsi_dev type ext4 (rw,_netdev)
```

   e. Use the `cp` command to copy `/boot/vmlinuz*` files to `/iscsi_dev`. Use the `ls` command to display the contents of `/iscsi_dev`.

```
# cp /boot/vmlinuz* /iscsi_dev
# ls /iscsi_dev
lost+found
vmlinuz...
```

      - These last steps test the usability of the iSCSI device.

10. Remove the `/dev/sda1` partition in preparation for the next practice.
    a. Use the `umount` command to unmount `/dev/sda1`.

```
# umount /dev/sda1
```

    b. Use the `fdisk` command to remove the `/dev/sda1` partition.

```
# fdisk /dev/sda
...
Command (m for help): d
Selected partition 1
Command (m for help): w
...
```

    c. Use the `cat` command to view the `/proc/partitions` file and ensure that the `sda1` partition no longer exists.

```
# cat /proc/partitions
major minor  # blocks  name
 202      0  12582912  xvda
...
   8      0  10485760  sda
```

- The `sda` device exists but the `sda1` partition is gone.

Practice 10-4: Configuring iSCSI Multipathing

Overview

In this practice, you:

- Perform all steps from **host03**
- Install the `device-mapper-multipath` packages
- Configure DM-Multipath for the iSCSI block device
- Verify failover to redundant path is successful
- Remove DM-Multipath

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Install and explore the DM-Multipath package.
   a. Use the `yum install` command to install the `device-mapper-multipath` packages.
      - The package might already be installed on **host03**.

```
# yum install device-mapper-multipath
...
Transaction Summary
========================================================
Install     2 Package(s)
...
Is this ok [y/N]: y
...
Complete!
```

      - In this example, version `0.4.9` is installed.

   b. Use the `ls` command to view the `/usr/share/doc/device-mapper-multipath-<version>` directory.

```
# ls /usr/share/doc/device-mapper-multipath-0.4.9
AUTHOR                     FAQ       multipath.conf.annotated
multipath.conf.synthetic   COPYING   multipath.conf
multipath.conf.defaults
```

      - The `multipath.conf` file is a basic configuration file that is used as a starting point.
      - The `multipath.conf.defaults` file contains a complete list of the default configuration values for the storage arrays supported by DM-Multipath.
      - The `multipath.conf.annotated` file contains a list of configuration options with descriptions.

c. Use the `less` command to view the `multipath.conf*` files in the `/usr/share/doc/device-mapper-multipath-<version>` directory.

```
# cd /usr/share/doc/device-mapper-multipath-0.4.9
# less multipath.conf
...
# less multipath.conf.defaults
...
# less multipath.conf.annotated
...
```

- You can create the `/etc/multipath.conf` file manually. Otherwise the `/usr/share/.../multipath.conf` file is copied into the `/etc` directory when you enable multipathing.
- You can copy and paste configuration settings from the `multipath.conf.defaults` file into the `/etc/multipath.conf` file for your specific storage arrays.
- The `multipath.conf.annotated` file is helpful in understanding the configuration directives to tune DM-Multipath for your specific environment.

2. Enable DM-Multipath.

a. Use the `mpathconf` command to enable DM-Multipath. Use the `mpathconf` command without any options to display the status.

```
# mpathconf --enable
# mpathconf
multipath is enabled
find_multipaths is disabled
user_friendly_names is enabled
dm_multipath module is loaded
multipathd is chkconfiged on
```

- In this example multipath is enabled, which removes any lines in the configuration file that blacklist all device nodes.
- The `dm_multipath` kernel module is loaded.
- The `multipathd` daemon is configured to start when the system boots to run levels `2-5`.

b. Use the `lsmod` command to view the `dm_multipath` module.

```
# lsmod | grep dm_multipath
dm_multipath  18466  0
dm_mod        84786  9  dm_multipath,dm_mirror,dm_log
```

c. Use the `chkconfig` command to list the run levels in which the `multipathd` is started.

```
# chkconfig multipathd --list
multipathd  0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

3. Obtain a unique identifier for the iSCSI block device.
   - This example assumes the iSCSI block device is `/dev/sda`. Yours might be different.

   Use the `scsi_id` command to obtain a unique identifier for `/dev/sda`.
   - The output of the `scsi_id` command in this example is `1IET_00010001`.

```
# scsi_id --whitelisted --replace-whitespace --device=/dev/sda
1IET_00010001
```

   - The `--whitelisted` option must be included to generate output.
   - The `--replace-whitespace` option replaces all whitespace in the output with underscores.
   - In this example, the unique name is `1IET_00010001`, which is used in the `/etc/multipath.conf` file.
4. Configure the `/etc/multipath.conf` file.

   Use the `vi` editor to edit the `/etc/multipath.conf` file as follows:
   - The default file contains all comments with the following exceptions:

```
defaults {
        user_friendly_names yes
}
blacklist {
}
```

   - Only one `defaults` section is allowed so ensure that you have only one uncommented `defaults` section after making your changes.

```
# vi /etc/multipath.conf
defaults {
    user_friendly_names     yes
    getuid_callout          "/lib/udev/scsi_id --whitelisted --
replace-whitespace --device=/dev/%n"
}
multipaths {
    multipath {
         wwid 1IET_00010001
    }
}
```

   - This configuration performs the following:
     - Enables the `user_friendly_names` option, which creates `mpathN` device names
     - Defines the `scsi_id` command that is used to obtain a unique identifier for the `/dev` device
     - Creates one multipath for `wwid 1IET_00010001`
   - `1IET_00010001` is the expected output of the `scsi_id` command for `/dev/sda`.

5. Start the DM-Multipath daemon and view the multipathed device.
   a. Use the `ls` command to view the `/dev/mapper` directory.

```
# ls -l /dev/mapper
crw-rw----  ...  control
lrwxrwxrwx  ...  vg_host03-lv_root -> ../dm-0
lrwxrwxrwx  ...  vg_host03-lv_swap -> ../dm-1
```

   b. Use the `service` command to start the `multipathd` daemon.

```
# service multipathd start
Starting multipathd daemon:                    [  OK  ]
```

   c. Use the `ls` command to view the `/dev/mapper` directory. (Only a partial output is shown.)

```
# ls -l /dev/mapper
crw-rw----  ...  control
lrwxrwxrwx  ...  mpatha -> ../dm-2
lrwxrwxrwx  ...  vg_host03-lv_root -> ../dm-0
lrwxrwxrwx  ...  vg_host03-lv_swap -> ../dm-1
```

   - Notice the new `mpatha` file in `/dev/mapper`.
   - This file name might be different on your system. Use your “`mpathN`” file name in the remaining steps in this practice.

   d. Use the `fdisk` command to display the devices in `/dev`.

```
# fdisk -l | grep /dev
...
Disk /dev/sda: 10.7 GB, 10737418240 bytes
Disk /dev/mapper/mpatha: 10.7 GB, 10737418240 bytes
```

   - Notice that the `/dev/mapper/mpatha` device now exists.

6. Verify the usability of the multipathed iSCSI device.
   a. Use the `mkfs` command to create an ext4 file system on `/dev/sda1`.

```
# mkfs -t ext4 /dev/mapper/mpatha
...
```

   b. Use the `mount` command (with the `_netdev,errors=continue` option) to mount the file system on `/iscsi_dev`.

```
# mount /dev/mapper/mpatha -o _netdev,errors=continue /iscsi_dev
```

   - Because it is a network device, be sure to use the `_netdev` option when mounting it.
   - Open-iSCSI can fail both paths temporarily while failing over, resulting in the system remounting a file system in read-only mode on error unless you include an `errors=continue` option as a `mount` option.

   c. Use the `mount` command to display the mounted file systems.

```
# mount
...
/dev/mapper/mpatha on /iscsi_dev type ext4
(rw,_netdev,errors=continue)
```

d. Use the `cp` command to copy `/boot/vmlinuz*` files to `/iscsi_dev`. Use the `ls` command to display the contents of `/iscsi_dev`.

```
# cp /boot/vmlinuz* /iscsi_dev
# ls /iscsi_dev
lost+found
vmlinuz...
```

- These last steps test the usability of the iSCSI device.

7. Verify DM-Multipath failover.

a. Use the `exit` command to log out from **host03**.

```
# exit
logout
Connection to host03 closed.
```

b. From **dom0**, use the `ssh` command to log in to **host03** using the `eth1` IP address, `192.0.2.104`. The `root` password is `oracle`.

```
[root@EDDDR9P1 ~]# ssh 192.0.2.104
The authenticity of host '192.0.2.104 (192.0.2.104)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.0.2.104' (RSA) to the list of
known hosts.
root@192.0.2.104's password: oracle
[root@host03 ~]#
```

c. Use the `ifdown` command to bring down the `eth0` interface.

```
# ifdown eth0
Device state: 3 (disconnected)
```

d. Use the `cp` command to copy `/boot/config*` files to `/iscsi_dev`. Use the `ls` command to display the contents of `/iscsi_dev`.

```
# cp /boot/config* /iscsi_dev
# ls /iscsi_dev
lost+found
config...
vmlinuz...
```

e. Use the `rm` command to remove the `vmlinuz*` files on `/iscsi_dev`.

```
# rm /iscsi_dev/vmlinuz*
rm: remove regular file '/iscsi_dev/vmlinuz...'? y
...
```

f. Use the `tail` command to view the `/var/log/messages` file.

- Sample output is shown. Your output might be different.

```
# tail /var/log/messages
...
<date> host03 iscsid: Kernel reported iSCSI connection 3:0 error
<date> host03 iscsid: connection 3:0 is operational after
recovery
```

8. Remove DM-Multipath.
   a. Use the `umount` to unmount `/iscsi_dev`.

```
# umount /iscsi_dev
```

   b. Use the `service` command to stop the `multipathd` daemon.

```
# service multipathd stop
Starting multipathd daemon:                    [  OK  ]
```

   c. Use the `chkconfig` command to disable the `multipathd` service from starting at boot time.

```
# chkconfig multipathd off
```

9. Remove iSCSI initiator and shut down **host03**.
   a. Use the `iscsiadm` command to log out from all sessions.

```
# iscsiadm -m node -u
Logging out of session [sid:2, target: iqn.2012-11.com.example.mypc:tgt3, portal: 192.0.2.1:3260]
Logout of [sid: 2, target: iqn.2012-11.com.example.mypc:tgt3, portal: 192.0.2.1:3260] successful.
```

   b. Use the `service` command to stop the `iscsid` service.

```
# service iscsid stop
Stopping iscsid:                               [ OK ]
```

   c. Use the `chkconfig` command to disable the `iscsid` service from starting at boot time.

```
# chkconfig iscsid off
```

   d. Use the `/bin/rm -r` command to delete the files in the `nodes` directory in the iSCSI database.

```
# /bin/rm -r /var/lib/iscsi/nodes/*
```

   e. Use the `/bin/rm -r` command to delete the files in the `send_targets` directory in the iSCSI database.

```
# /bin/rm -r /var/lib/iscsi/send_targets/*
```

   f. Use the `reboot` command to reboot **host03**.

```
# reboot
...
Connection to host03 closed.
```

Practices for Lesson 11: XFS File System

Chapter 11

Practices for Lesson 11: XFS File System

Practices Overview

In these practices, you will perform the following:

- Create and mount an XFS file system
- Set quotas on an XFS file system
- Back up and restore an XFS file system

Practice 11-1: Creating an XFS File System

Overview

In this practice, you will perform the following tasks:

- Install the `xfsprogs` package.
- Create an XFS file system.
- Mount the file system and copy files to the file system.
- Attempt to resize an XFS file system.
- Change and view the parameters of an XFS file system.

Assumptions

- You are the `root` user on **dom0**.
- You completed Practice 8-1 that created the `/dev/xvdb1` partition on **host03**.
  - If you did not complete this practice, perform step 3b to create the partition.
  - If you did complete this practice, skip step 3b and go to step 3c

Tasks

1. Use the `ssh` command to log in to **host03**.
   - The `root` password is `oracle`.

```
# ssh host03
root@host03's password: oracle
Last login: ...
[root@host03 ~]#
```

2. Use the `yum install` command to install the `xfsprogs` software package.

```
# yum install xfsprogs
...
Transaction Summary
============================================================
Install       1 Package(s)
Total download size: 808 k
Installed size: 3.6 M
Is this ok [y/N]: y
...
Complete!
```

3. Create an XFS file system.
   a. Use the `fdisk` command to display the available devices on your system.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *      1      64     512000   83   Linux
/dev/xvda2         64    1567   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
```

```
/dev/xvdb1          1      132     1060258+  83   Linux
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes
Disk /dev/mapper/vg_host01-lv-root: 9202 MB, 9202302976 bytes
Disk /dev/mapper/vg_host01-lv-swap: 3154 MB, 3154116608 bytes
```

- Notice that there is one partition on the `/dev/xvdb` device.

b. If `/dev/xvdb1` does not exist, use the `fdisk` command as follows to create a 1 GB primary partition on `/dev/xvdb`.

```
# fdisk /dev/xvdb
...
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1): ENTER
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1305, default
1305): +1G
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

c. Use the `mkfs.xfs` command with any arguments or options to display the usage.

```
# mkfs.xfs
No device name given in argument list
Usage: mkfs.xfs
/* blocksize */          [-b log=n|size=num]
/* metadata */           [-m crc=[0|1]
/* data subvol */        [-d agcount=n,agsize=n,file,name=xxx...
/* force overwrite */    [-f]
/* inode size */         [-i log=n|perblock=n|size=num,maxpct...
/* no discard */         [-K]
/* log subvol */         [-l agnum=n,internal,size=num,logdev...
/* label */              [-L label (maximum 12 characters)]
...
```

d. Use the `mkfs.xfs` command to create an XFS file system on `/dev/xvdb1`.

- Use the `–L XFS` option to specify a label of XFS.
- Use the `–b size=512` option to specify a block size of 512 bytes.

```
# mkfs.xfs –L XFS –b size=512 /dev/xvdb1
mkfs.xfs: /dev/xvdb1 appears to contain an existing filesystem
(ext3).
mkfs.xfs: Use the –f option to force overwrite.
```

- Recall that you created an ext3 file system on `/dev/xvdb1` in "Practice 8-5: Using `kpartx`."
- By default, `mkfs.xfs` does not write to the device if it suspects that there is an existing file system or partition table on the device.

e. Repeat the previous step but add the `-f` option to the `mkfs.xfs` command.

- The `-f` option forces overwrite of the existing ext3 file system on `/dev/xvdb1`.

```
# mkfs.xfs –f –L XFS –b size=512 /dev/xvdb1
meta-data=/dev/xvdb1   isize=256   agcount=4, agzise=530129 blks
         =             sectsz=512  attr=2, projid32bit=1
         =             crc=0
data     =             bsize=512   blocks=2120516, imaxpct=25
         =             sunit=0     swidth=0 blks
naming   =version 2    bsize=4096  ascii-ci=0
log      =internal log bsize=512   blocks=20480, version=2
         =             sectsz=512  sunit=0 blks, lazy-count=1
realtime =none         extsz=4096  blocks=0, rtextents=0
```

- Notice that the block size of the `data` section and the `log` section is 512 bytes. The default is 4 KB.
- Notice that the block size of the `naming` (file system directory) section is 4 KB.

f. Use the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
# blkid | grep xfs
/dev/xvdb1: LABEL=”XFS” UUID=... TYPE=”xfs”
```

4. Mount the XFS file system and verify usability.

a. Use the `mkdir` command to make the `/xfs` directory. This directory is used as the mount point.

```
# mkdir /xfs
```

b. Use the `vi` editor to add the following entry in `/etc/fstab` to mount the file system:

```
# vi /etc/fstab
...
/dev/xvdb1    /xfs    xfs    defaults    0  0
```

- You can use LABEL or UUID when mounting as well.

c. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
# mount -a
```

d. Use the `df –h` command to display the mounted file systems.

```
# df –h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/xvdb1    1.1G  4.1M  1022M    1%  /xfs
```

e. Use the `cp` command to copy some files to `/xfs`. Use the `ls` command to list the contents of the `/xfs` directory.

- Copy `/boot/vmlinuz*` to `/xfs`.

```
# cp /boot/vmlinuz* /xfs
# ls /xfs
vmlinuz-2.6.32-431.el6.x86_64
vmlinuz-3.8.13-16.2.1.el6uek.x86_64
vmlinuz-3.8.13-26.1.1.el6uek.x86_64
```

- This confirms that the XFS file system is usable.

5. Attempt to increase the size of an XFS file system.

a. Use the `xfs_growfs` command to expand the data section to use all available space.

```
# xfs_growfs -d /xfs
meta-data=/dev/xvdb1   isize=256   agcount=4, agzise=530129 blks
         =             sectsz=512  attr=2, projid32bit=1
         =             crc=0
data     =             bsize=512   blocks=2120516, imaxpct=25
         =             sunit=0     swidth=0 blks
naming   =version 2    bsize=4096  ascii-ci=0
log      =internal log bsize=512   blocks=20480, version=2
         =             sectsz=512  sunit=0 blks, lazy-count=1
realtime =none         extsz=4096  blocks=0, rtextents=0
data size unchanged, skipping
```

- Notice the "`data size unchanged, skipping`" message.
- The data section was not changed, because there is no space available on the underlying device, `/dev/xvdb1`.
- The `xfs_growfs` command is most often used with logical volumes.

6. Change and view the parameters of an XFS file system.

a. Use the `xfs_admin` command to display the file system label and the UUID.

```
# xfs_admin -l /dev/xvdb1
label = "XFS"
# xfs_admin -u /dev/xvdb1
UUID = ...
```

b. Use the `xfs_admin` command to change the file system label to "`xfs`" (all lowercase).

```
# xfs_admin -L xfs /dev/xvdb1
xfs_admin: /dev/xvdb1 contains a mounted filesystem
fatal error - couldn't initialize XFS library
```

- Notice that the command failed because the file system is mounted.

c. Use the `umount /xfs` command to unmount the file system.

```
# umount /xfs
```

d. Repeat step 5b to change the file system label to "`xfs`".

```
# xfs_admin –L xfs /dev/xvdb1
Writing all SBs
new label = "xfs"
```

- Notice that the command was successful when the file system is not mounted.

e. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
# mount -a
```

f. Use the `blkid` command to display the new label on the XFS file system.

```
# blkid | grep xfs
/dev/xvdb1: LABEL="xfs" UUID=… TYPE="xfs"
```

- Notice that the label is now lowercase.

Practice 11-2: Setting Quotas on an XFS File System

Overview

In this practice, you perform the following:

- Enable user quotas on an XFS file system.
- Use the `xfs_quota` command in interactive mode.
- Set limits for the `oracle` user and view quota information.
- Verify quota settings limit disk space to `oracle` user.
- Remove disk quotas.

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Enable quotas for users on an XFS file system.
   - This step directs you to edit the `/etc/fstab` file to enable quotas.
   - Alternatively, you could use the `mount` command and include XFS quota mount options.

   a. Use the `umount` command to unmount `/xfs`.

   ```
   # umount /xfs
   ```

   b. Use the `vi` editor to change the mount option for the XFS file system as shown.
      - Change "`defaults`" to "`rw,quota`" for the `/dev/xvdb1` entry.

   ```
   # vi /etc/fstab
   ...
   /dev/xvdb1    /xfs    xfs    defaults    0  0  (old value)
   /dev/xvdb1    /xfs    xfs    rw,quota    0  0  (new value)
   ```

   c. Use the `mount` command to mount the file systems in `/etc/fstab`.

   ```
   # mount -a
   ```

   d. Use the `mount` command to view the new mount options for the XFS file system.

   ```
   # mount | grep xfs
   /dev/xvdb1 on /xfs type xfs (rw,quota)
   ```

   - Notice that the new mount options are "`rw,quota`".

   e. Use the `xfs_quota` command to report overall state information.

   ```
   # xfs_quota –x –c state
   User quota state on /xfs (/dev/xvdb1)
     Accounting: ON
     Enforcement: ON
     Inode: #37 (1 blocks, 1 extents)
   Group quota state on /xfs (/dev/xvdb1)
     Accounting: OFF
     Enforcement: OFF
   ```

```
  Inode: N/A
Project quota state on /xfs (/dev/xvdb1)
  Accounting: OFF
  Enforcement: OFF
  Inode: N/A
Blocks grace time: [7 days 00:00:30]
Inodes grace time: [7 days 00:00:30]
Realtime Blocks grace time: [7 days 00:00:30]
```

- Notice that accounting and enforcement are ON for user quota but OFF for group and project quota.

2. Use the `xfs_quota` command in interactive mode.
   a. Enter the `xfs_quota` command without any options or arguments.
      - The `xfs_quota>` prompt is displayed.

```
# xfs_quota
xfs_quota>
```

   b. Enter `help` at the `xfs_quota>` prompt to display online help for the commands.

```
xfs_quota> help
df [-bir] [-hn] [-f file] -- show free and used counts for ...
help [command] -- help for one or all commands
print -- list known mount points and projects
quit -- exit the program
quota [-bir] [-gpu] [-hnNv] [-f file] [id|name]... -- show ...
Use 'help commandname' for extended help.
```

   c. Enter `help quota` at the `xfs_quota>` prompt to display extended help for the `quota` command.

```
xfs_quota> help quota
quota [-bir] [-gpu] [-hnNv] [-f file] [id|name]... -- show ...
 display usage and quota information
 -g -- display group quota information
 -p -- display project quota information
 -u -- display user quota information
 -b -- display number of blocks used
 -b -- display number of inodes used
...
```

   d. Enter `quota -b` at the `xfs_quota>` prompt to display the number of blocks used.

```
xfs_quota> quota -b
Disk quotas for user root (0)
Filesystem     Blocks   Quota   Limit   Warn/Time     Mounted on
/dev/xvdb1      12995       0       0    00 [-------] /xfs
```

e. Enter `quota -i` at the `xfs_quota>` prompt to display the number of inodes used.

```
xfs_quota> quota -i
Disk quotas for user root (0)
Filesystem     Files     Quota    Limit    Warn/Time     Mounted on
/dev/xvdb1         6         0        0    00 [-------] /xfs
```

f. Enter `quit` at the `xfs_quota>` prompt to exit interactive mode.

```
xfs_quota> quit
```

3. Set limits for the `oracle` user and view quota information.

a. Use the `xfs_quota` command to set a soft limit of 5 MB and a hard limit of 6 MB on the XFS file system for user `oracle`.

```
# xfs_quota –x –c 'limit –u bsoft=5m bhard=6m oracle' /xfs
```

b. Use the `xfs_quota` command to print all paths with devices and identifiers.

```
# xfs_quota –x –c print
Filesystem     Pathname
/xfs           /dev/xvdb1 (uquota)
```

c. Use the `xfs_quota` command to report file system usage for blocks.

- The `-h` option provides human-readable output.
- The `-b` option provides block information.

```
# xfs_quota –x –c 'free –hb'
Filesystem  Size   Used    Avail Use%  Pathname
/dev/xvdb1  1.0G  16.7M  1008.1M   2%  /xfs
```

d. Use the `xfs_quota` command to report file system usage for inodes.

- The `-i` option provides inode information.

```
# xfs_quota –x –c 'free –hi'
Filesystem  Inodes  Used   Free  Use%  Pathname
/dev/xvdb1    1.1m     7   1.1m    0%  /xfs
```

e. Use the `xfs_quota` command to report file system quota information in human-readable form.

```
# xfs_quota –x –c 'report –h'
User quota on /xfs (/dev/xvdb1)
                       Blocks
User ID      Used     Soft     Hard    Warn/Grace
----------  ----------------------------------------
root        12.7M        0        0     00 [--------]
oracle          0       5M       6M     00 [--------]
```

4. Test quota settings.
   a. Use the `chmod` command to give read, write, and execute permissions on the `/xfs` directory. Use the `ls –d /xfs` command to view the permissions on the `/xfs` directory.

```
# chmod 777 /xfs
# ls -ld /xfs
drwxrwxrwx ... /xfs
```

   b. Use the `su – oracle` command to become the `oracle` user.

```
# su - oracle
[oracle@host03 ~]$
```

   c. As the `oracle` user, use the `cd` command to change to the `/xfs` directory.

```
[oracle@host03 ~]$ cd /xfs
[oracle@host03 xfs]$
```

   d. As the `oracle` user, use the `dd if=/dev/zero of=bigfile bs=1K count=7168` command to attempt to create a 7 MB file.
      - The command fails because the `oracle` user has a hard limit of 6 MB on the XFS file system.

```
[oracle@host03 xfs]$ dd if=/dev/zero of=bigfile bs=1K count=7168
dd: writing ‘bigfile’: Disk quota exceeded
6145+0 records in
6144+0 records out
6291456 bytes (6.3 MB) copied, ...
```

      - Notice the “Disk quota exceeded” error message.

5. Disable XFS quotas and re-test.
   a. Use the `exit` command to log off as the `oracle` user. Use the `whoami` command to confirm that you are the root user.

```
[oracle@host03 xfs]$ exit
logout
# whoami
root
```

   b. Use the `umount` command to unmount `/xfs`.

```
# umount /xfs
```

   c. Use the `vi` editor to change the mount option for the XFS file system as shown.
      - Change “`rw,quota`” to “`defaults`” for the `/dev/xvdb1` entry.

```
# vi /etc/fstab
...
/dev/xvdb1    /xfs    xfs    rw,quota    0  0  (old value)
/dev/xvdb1    /xfs    xfs    defaults    0  0  (new value)
```

   d. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
# mount -a
```

e. Use the `mount` command to view the new mount options for the XFS file system.

```
# mount | grep xfs
/dev/xvdb1 on /xfs type xfs (rw)
```

- Notice that the mount option does not show that `quota` is enabled.

f. Use the `su - oracle` command to become the `oracle` user.

```
# su - oracle
[oracle@host03 ~]$
```

g. As the `oracle` user, use the `cd` command to change to the `/xfs` directory.

```
[oracle@host03 ~]$ cd /xfs
[oracle@host03 xfs]$
```

h. As the `oracle` user, use the `dd if=/dev/zero of=bigfile bs=1K count=7168` command to attempt to create a 7 MB file.

```
[oracle@host03 xfs]$ dd if=/dev/zero of=bigfile bs=1K count=7168
7168+0 records in
7168+0 records out
7340032 bytes (7.3 MB) copied, ...
```

- The file is successfully created with no disk quotas.

i. Use the `exit` command to log off as the `oracle` user. Use the `whoami` command to confirm that you are the `root` user.

```
[oracle@host03 xfs]$ exit
logout
# whoami
root
```

Practice 11-3: Backing Up and Restoring XFS File Systems

Overview

In this practice, you perform the following:

- Install the `xfsdump` package.
- Use `xfsdump` to back up an XFS file system.
- Use `xfsdump` to back up a specific file in an XFS file system.
- View the `xfsdump` inventory.
- Use `xfsrestore` to restore from a backup.

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Use the `yum install` command to install the `xfsdump` software package.

```
# yum install xfsdump
...
Transaction Summary
=============================================================
Install       1 Package(s)
Total download size: 293 k
Installed size: 938 k
Is this ok [y/N]: y
...
Complete!
```

2. Use the `xfsdump` utility.
   a. Run the `xfsdump -h` command to display the usage.

```
# xfsdump -h
xfsdump: version 3.1.3 (dump format 3.0)
xfsdump: usage: xfsdump [ -a (dump DMF dualstate files as ...
                        [ -b <blocksize> ]
                        [ -c <media change alert program> ]
                        [ -d <dump media file size> ]
                        [ -e (allow files to be excluded) ]
                        [ -f <destination> ... ]
                        [ -h (help) ]
                        [ -l <level> ]
                        [ -m (force usage of minimal rmt) ]
                        [ -o (overwrite tape) ]
                        [ -p <seconds between progress ...
                        [ -q <use QIC tape settings> ]
...
```

b. Use the `xfsdump` command to back up the entire XFS file system on `/xfs` to a local file, `/usr/tmp/backup`.

- Omitting the `-l <level>` option performs a full backup (level 0).
- Enter a session label of your choice when prompted.
- Enter a media label of your choice when prompted.
- These labels could be provided as options to the `xfsdump` command:
  - (`-L <session_label>`)
  - (`-M <media_label>`)

```
# xfsdump -f /usr/tmp/backup /xfs
xfsdump: using file dump (drive_simple strategy
xfsdump: version 3.1.3 (dump format 3.0) - type ^C for status...

================ dump label dialog ===================
please enter label for this dump session (timeout in 300 sec)
 -> Full backup of /xfs on 2/3/2014
session label entered: "Full backup of /xfs on 2/3/2014"
-------------------- end dialog ----------------------

xfsdump: level 0 dump of host03.example.comj:/xfs
xfsdump: dump date: ...
...
xfsdump: /var/lib/xfsdump/inventory created

================ media label dialog ===================
please enter label for media in drive 0 (timeout in 300 sec)
 -> Full backup to /usr/tmp/backup on 2/3/14
media label entered: "Full backup to /usr/tmp/backup on 2/3/14"
-------------------- end dialog ----------------------

xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
...
xfsdump: Dump Status: SUCCESS
```

c. Use the `xfsdump` command to back up only the `/xfs/bigfile` to `/usr/tmp/bigfile_backup`.

- Provide a session label of your choice as a command-line argument.
- Provide a media label of your choice as a command-line argument.

```
# xfsdump -f /usr/tmp/bigfile_backup -L "session label" -M
"media label" -s bigfile /xfs
xfsdump: using file dump (drive_simple strategy
xfsdump: version 3.1.3 (dump format 3.0) - type ^C for status...
```

```
xfsdump: level 0 dump of host03.example.comj:/xfs
xfsdump: dump date: ...
...
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
...
xfsdump: Dump Status: SUCCESS
```

- Notice that you were not prompted for session label or media label.

d. Use the `xfsdump -I` command to display the inventory.

- Only selected lines are shown from the output.

```
# xfsdump -I
file system 0:
        fs id:    ...
        session 0:
                ...
                session label: "Full back of /xfs on 2/3/2014"
                ...
                level:        0
                ...
                stream 0:
                        pathname: /usr/tmp/backup
                        ...
                        media file 0:
                                 ...
                                 media label: "Full backup ...
        session 1:
                ...
                session label: "session label"
                ...
                level:        0
                ...
                stream 0:
                        pathname: /usr/tmp/bigfile_backup
                        ...
                        media file 0:
                                 ...
                                 media label: "media label"
                                 ...
xfsdump: Dump Status: SUCCESS
```

- Notice that there are two sessions:
  - One session for the full backup of the `/xfs` directory
  - One session for the backup of the single file, `bigfile`, in the `/xfs` directory.

e. Make a change to the `bigfile` in the `/xfs` directory. Use the `ls -l` command before and after making the change to note the difference in file size.

- The change to `bigfile` is made by copying `/etc/fstab` and overwriting the original `bigfile`.

```
# cd /xfs
# ls -l bigfile
-rw-rw-r-- ... 7340032 ... bigfile
# cp /etc/fstab bigfile
cp: overwrite ‘bigfile’? y
# ls -l bigfile
-rw-rw-r-- ... 817 ... bigfile
```

- In this example, the original size of `bigfile` was 7340032 bytes. The new size is 817 bytes.

f. Use the `xfsdump` command to perform a level-1 back up of the entire XFS file system on `/xfs` to a local file, `/usr/tmp/level1_backup`.

- Provide a session label of your choice as a command-line argument.
- Provide a media label of your choice as a command-line argument.

```
# xfsdump –l 1 –f /usr/tmp/level1_backup –L “level 1 session” –M
“level 1 media” /xfs
xfsdump: using file dump (drive_simple strategy
xfsdump: version 3.1.3 (dump format 3.0) – type ^C for status...
xfsdump: level 1 incremental dump of host03.example.comj:/xfs
based on level 0 dump begun ...
xfsdump: dump date: ...
...
xfsdump: Dump Status: SUCCESS
```

g. Use the `xfsdump –I` command to display the inventory.

- Only selected lines are shown from the output.

```
# xfsdump –I
...
        session 2:
                ...
                session label: “level 1 session”
                ...
                level:         1
                ...
                stream 0:
                        pathname: /usr/tmp/level1_backup
                        ...
                        media file 0:
                                 ...
                                 media label: “level 1 media”
                                 ...
xfsdump: Dump Status: SUCCESS
```

- Notice that the last backup is a level-1 backup.

3. Use the `xfsrestore` utility.
   a. Run the `xfsrestore -h` command to display the usage.

```
# xfsrestore -h
xfsrestore: version 3.1.3 (dump format 3.0)
xfsrestore: usage: xfsrestore [ -a <alt. workspace dir> ... ]
                              [ -b <blocksize> ]
                              [ -c <media change alert ... ]
                              [ -e (don't overwrite existing ...
                              [ -f <source> ... ]
                              [ -h (help) ]
                              [ -i (interactive) ]
                              [ -m (force usage of minimal ...
                              [ -n <file> (restore only if ...
                              [ -o (restore owner/group even ...
                              [ -p <seconds between progress ...
                              [ -q <use QIC tape settings> ]
...
```

   b. Use the `ls -l /xfs` command to view the contents and file sizes of the XFS file system before initiating an `xfsrestore`.

```
# ls -l /xfs
-rw-rw-r-- ...     817 ... bigfile
-rwxr-xr-x ... 4128944 ... vmlinuz-2.6.32-431.el6.x86_64
-rwxr-xr-x ... 4588304 ... vmlinuz-3.8.13-16.2.1.el6uek.x86_64
-rwxr-xr-x ... 4589488 ... vmlinuz-3.8.13-26.1.1.el6uek.x86_64
```

   - Notice that the size of `bigfile` is 817 bytes.

   c. Use the `xfsrestore` command to restore from the `/usr/tmp/backup` to the `/xfs` directory.
   - The `/usr/tmp/backup` is a level-0 backup.

```
# xfsrestore -f /usr/tmp/backup /xfs
xfsrestore: using file dump (drive_simple strategy
xfsrestore: version 3.1.3 (dump format 3.0) - type ^C for...
...
xfsrestore: level: 0
xfsrestore: session label: "Full backup of /xfs on 2/3/2014"
xfsrestore: media label: "Full backup to /usr/tmp/backup on ...
...
xfsrestore: Restore Status: SUCCESS
```

   d. Use the `ls -l /xfs` command to view the contents and file sizes of the XFS file system.

```
# ls -l /xfs
-rw-rw-r-- ... 7340032 ... bigfile
-rwxr-xr-x ... 4128944 ... vmlinuz-2.6.32-431.el6.x86_64
```

```
-rwxr-xr-x ... 4588304 ... vmlinuz-3.8.13-16.2.1.el6uek.x86_64
-rwxr-xr-x ... 4589488 ... vmlinuz-3.8.13-26.1.1.el6uek.x86_64
```

- Notice that the size of `bigfile` is now 7340032 bytes.
- Recall that the level-0 backup was done before changing the size of `bigfile` in step 2e.

e. Use the `rm` command to remove the `vmlinuz-2.6.32-431.el6.x86_64` file in the `/xfs` directory.

```
# rm /xfs/vmlinuz-2.6.32-431.el6.x86_64
rm: remove regular file '/xfs/vmlinuz-2.6.32-431.el6.x86_64'? y
```

f. Use the `xfsrestore` command to restore from the `/usr/tmp/level1_backup` to the `/xfs` directory.

- The `/usr/tmp/level1_backup` is a level-1 backup.

```
# xfsrestore -f /usr/tmp/level1_backup /xfs
xfsrestore: using file dump (drive_simple strategy
xfsrestore: version 3.1.3 (dump format 3.0) - type ^C for...
...
xfsrestore: level: 1
xfsrestore: session label: "level 1 session"
xfsrestore: media label: "level 1 media"
...
xfsrestore: Restore Status: SUCCESS
```

g. Use the `ls –l /xfs` command to view the contents and file sizes of the XFS file system.

```
# ls –l /xfs
-rw-rw-r-- ...     817 ... bigfile
-rwxr-xr-x ... 4588304 ... vmlinuz-3.8.13-16.2.1.el6uek.x86_64
-rwxr-xr-x ... 4589488 ... vmlinuz-3.8.13-26.1.1.el6uek.x86_64
```

- Recall that the level-1 `xfsdump` to `/usr/tmp/level1_backup` (performed in step 2f) only backed up the file that changed (`bigfile`) since the previous backup.
- Therefore, only the `bigfile` (817 bytes) was restored.
- The `vmlinuz-2.6.32-431.el6.x86_64` file was not restored, because it was not backed up in the level-1 `xfsdump`.

h. Use the `xfsrestore` utility to restore from `/usr/tmp/backup`, and then restore from the `/usr/tmp/level1_backup`.

- Use the `ls –l` command after each restore to display the contents of `/xfs`.

```
# xfsrestore -f /usr/tmp/backup /xfs
...
xfsrestore: Restore Status: SUCCESS
# ls –l /xfs
-rw-rw-r-- ... 7340032 ... bigfile
-rwxr-xr-x ... 4128944 ... vmlinuz-2.6.32-431.el6.x86_64
```

```
-rwxr-xr-x ... 4588304 ... vmlinuz-3.8.13-16.2.1.el6uek.x86_64
-rwxr-xr-x ... 4589488 ... vmlinuz-3.8.13-26.1.1.el6uek.x86_64
# xfsrestore -f /usr/tmp/level1_backup /xfs
...
xfsrestore: Restore Status: SUCCESS
# ls -l /xfs
-rw-rw-r-- ...     817 ... bigfile
-rwxr-xr-x ... 4128944 ... vmlinuz-2.6.32-431.el6.x86_64
-rwxr-xr-x ... 4588304 ... vmlinuz-3.8.13-16.2.1.el6uek.x86_64
-rwxr-xr-x ... 4589488 ... vmlinuz-3.8.13-26.1.1.el6uek.x86_64
```

- All files are restored, including the latest copy of `bigfile` (817 bytes).

4. Clean up before proceeding to Lesson 12.
   a. Use the `umount` command to unmount `/xfs`.
      - Use the `cd` command to ensure that you are not in the `/xfs` directory.

```
# cd
# umount /xfs
```

   b. Use the `vi` editor to delete the XFS entry.

```
# vi /etc/fstab
/dev/xvdb1    /xfs    xfs    defaults    0  0   (delete this line)
```

   c. Use the `rm` command to remove the `xfsdump` files in `/usr/tmp`.

```
# rm /usr/tmp/*
rm: remove regular file '/usr/tmp/backup'? y
rm: remove regular file '/usr/tmp/bigfile_backup'? y
rm: remove regular file '/usr/tmp/level1_backup'? y
```

Practices for Lesson 12: Btrfs File System

Chapter 12

Practices for Lesson 12: Btrfs File System

Practices Overview

In these practices, you will perform the following:

- Verify that the `btrfs-progs` package is installed.
- Create Btrfs file systems.
- Resize a Btrfs file system.
- Create and mount Btrfs subvolumes and snapshots.
- Take a snapshot of a file in a Btrfs subvolume.
- Recover a corrupted Btrfs file system.
- Install Oracle Linux 6.5 on the **host06** VM by using the UEK Boot ISO.
- Explore the Btrfs root file system on the **host06** VM.

Practice 12-1: Creating a Btrfs File System

Overview

In this practice, you will perform the following tasks:

- Install the `btrfs-progs` package.
- Load the `btrfs` kernel module.
- Create a single-disk Btrfs file system with different specifications.
- Mount the file system, copy a file to it, and display the file system information.
- Resize a Btrfs file system.
- Create a two-disk Btrfs file system with different specifications.
- Mount the file system, copy a file to it, and display the file system information.

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Install the `btrfs-progs` package and load the kernel module.
   a. Use the `yum install` command to install the `btrfs-progs` package.

```
# yum install btrfs-progs
...
Transaction Summary
==============================================================
Install       1 Package(s)
Total download size: 393 k
Installed size: 2.7 M
Is this ok [y/N]: y
...
Complete!
```

   b. Use the `modprobe` command to load the `btrfs` kernel module. Use the `lsmod` command to ensure that the `btrfs` kernel module is loaded.

```
# modprobe btrfs
# lsmod | grep btrfs
btrfs          810770   0
zlib_deflate    21991   1   btrfs
libcrc32c        1252   2   btrfs,xfs
```

2. Complete the single-disk file system exercise.

   a. Use the `fdisk` command to display the available devices on your system.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *      1      64     512000   83   Linux
/dev/xvda2         64    1567   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
/dev/xvdb1          1     132   1060258+  83   Linux
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes
Disk /dev/mapper/vg_host03-lv-root: 9202 MB, 9202302976 bytes
Disk /dev/mapper/vg_host03-lv-swap: 3154 MB, 3154116608 bytes
```

   - In this example, three disks are available:
     - `/dev/xvda` – 12 GB
     - `/dev/xvdb` – 10 GB
     - `/dev/xvdd` – 10 GB
   - The `/dev/xvdb` device has one partition, `/dev/xvdb1`.

   b. Use the `fdisk` command to delete the `/dev/xvdb1` partition.

   - This step assumes that you created the `/dev/xvdb1` partition in either Practice 8-1 or Practice 11-1.
   - If the `/dev/xvdb1` partition does not exist, skip this step and go to step 2c.

```
# fdisk /dev/xvdb
...
Command (m for help): d
Selected partition 1
Command (m for help): w
```

   c. Use the `mkfs.btrfs` command to make a Btrfs file system on `/dev/xvdb`.

```
# mkfs.btrfs -L Btrfs /dev/xvdb
WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
/dev/xvdb appears to contain a partition table (dos).
Use the -f option to force overwrite.
```

d. Repeat the previous command but include the `-f` option.

```
# mkfs.btrfs -f -L Btrfs /dev/xvdb
WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
Detected a SSD, turning off metadata duplication. Mkfs with -m
dup if you want to force metadata duplication.
...
```

- In this example, metadata duplication is turned off.
- "Detected a SSD" is technically not a bug. The SSD detection code in Btrfs is looking for rotational disks, which a Xen-backed image is not. You can check this by looking at `/sys/class/block/$device/queue/rotational`. It will be set to `0`. If you want Btrfs to think the disk is rotational, set this to `1`. This happens on all HVM with PV and PVM virtual machines using the `xvd` block devices.

e. Repeat the previous command but include the `-m dup` option.

```
# mkfs.btrfs -f -m dup -L Btrfs /dev/xvdb
WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
fs created label Btrfs on /dev/xvdb
    nodesize 4096 leafsize 4096 sectorsize 4096 size 10.00GB
Btrfs Btrfs v0.20-rc1
```

f. Run the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
# blkid | grep btrfs
/dev/xvdb:  LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
```

g. Use the `mkdir` command to make the `/btrfs` directory. This directory is used as the mount point.

```
# mkdir /btrfs
```

h. Use the `vi` editor to add the following entry in `/etc/fstab` to mount the file system:

```
# vi /etc/fstab
...
/dev/xvdb  /btrfs  btrfs  defaults  0  0
```

- You can use LABEL or UUID when mounting as well.

i. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
# mount -a
```

j. Use the `cp` command to copy `/boot/vmlinuz-2.6.32*` to `/btrfs`.

```
# cp /boot/vmlinuz-2.6.32* /btrfs
```

k. Run the `df -h` command, then run the `sync` command, and then run `df -h` again to display the space information for `/btrfs`.

```
# df -h /btrfs
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvdb       10G   56K   8.0G    1%  /btrfs
# sync
```

```
# df -h /btrfs
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvdb       10G  4.3M   8.0G    1%  /btrfs
```

- Notice that the `sync` command resulted in a difference in the output of `df -h`.

l. Run the `btrfs file` command to display the available `btrfs filesystem` commands.

```
# btrfs file
usage: btrfs filesystem [<group>] <command> [<args>]
    btrfs filesystem df <path>
         Show space usage information for a mount point
    btrfs filesystem show [--all-devices] [<uuid>|<label>
         Show the structure of a filesystem
    ...
```

m. Use the `btrfs filesystem show` command to view the structure of the file system.

```
# btrfs file show
Label: 'Btrfs' uuid: ...
        Total devices 1 FS bytes used 4.23MB
        devid    1 size 10.00GB used 2.04GB path /dev/xvdb
Btrfs Btrfs v0.20-rc1
```

n. Use the `btrfs filesystem df` command to show the space used on the mount point.

```
# btrfs file df /btrfs
Data, total=8.00MB, used=4.19MB
System, DUP: total=8.00MB, used=4.00KB
System, total=4.00MB, used=0.00
Metadata, DUP: total=1.00GB, used=32.00KB
Metadata, total=8.00MB, used=0.00
```

- Notice that, by default, the metadata is duplicated even on a single-disk file system.

3. Complete the single-disk file system without duplicating the metadata exercise.

a. Use the `umount` command to unmount `/btrfs`.

```
# umount /btrfs
```

b. Use the `mkfs.btrfs -m single` command to make a Btrfs file system on `/dev/xvdb`.

```
# mkfs.btrfs -f -m single -L Btrfs /dev/xvdb
WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
fs created label Btrfs on /dev/xvdb
    nodesize 4096 leafsize 4096 sectorsize 4096 size 10.00GB
Btrfs Btrfs v0.20-rc1
```

c. Use the `mount` command to mount the file system.

```
# mount -a
```

d. Use the `cp` command to copy `/boot/vmlinuz-2.6.32*` to `/btrfs`.

```
# cp /boot/vmlinuz-2.6.32* /btrfs
```

e. Run the `sync` command, and then use the `df -h` command to display the space information for `/btrfs`.

```
# sync
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      10G  4.0M    10G    1%  /btrfs
```

f. Use the `btrfs filesystem df` command to show the space used on the mount point.

```
# btrfs file df /btrfs
Data, total=8.00MB, used=3.94MB
System, total=4.00MB, used=4.00KB
Metadata, total=8.00MB, used=32.00KB
```

- Notice that the metadata is not duplicated.

4. Increase and decrease the size of a Btrfs file system.

a. Use the `btrfs file resize` command to increase the size of the file system by 2 GB.

```
# btrfs file resize +2G /btrfs
Resize '/btrfs/' of '+2G'
ERROR: unable to resize '/btrfs/' - File too large
```

- The error indicates that you are unable to increase the size of the file system.
- The resize command does not change the size of the underlying partition.
- If you wish to enlarge a file system, you must expand the partition first.

b. Use the `btrfs file resize` command to reduce the size of the file system by 2 GB.

```
# btrfs filesystem resize -2G /btrfs
Resize '/btrfs/' of '-2G'
```

- Notice that you are able to reduce the size of the file system.

c. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb     8.0G  4.0M   8.0G    1%  /btrfs
```

- Notice that the size is reduced by 2 GB, from 10 GB down to 8 GB.

d. Use the `btrfs file resize` command to increase the size of the file system by 1 GB.

```
# btrfs file resize +1G /btrfs
Resize '/btrfs/' of '+1G'
```

- Notice that the resize was successful this time because space is available on the underlying partition.

e. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb     9.0G  4.0M   9.0G    1%  /btrfs
```

- Notice that the size is increased by 1 GB, from 8 GB up to 9 GB.

5. Add a disk to and remove a disk from a Btrfs file system.

a. Use the `btrfs device add` command to add a 10 GB disk, `/dev/xvdd`, to the existing `/btrfs` file system.

```
# btrfs device add /dev/xvdd /btrfs
```

b. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      19G  4.0M    19G    1%  /btrfs
```

- Notice that the size is increased by 10 GB, from 9 GB up to 19 GB.

c. Use the `btrfs file show` command to verify that the file system now contains two disks.

```
# btrfs file show
Label: 'Btrfs'  uuid: ...
        Total devices 2 FS bytes used 3.98MB
        devid    2 size 10.00GB used 0.00MB path /dev/xvdd
        devid    1 size 9.00GB used 20.00MB path /dev/xvdb
Btrfs Btrfs v0.20-rc1
```

- Notice that all of the data is on `/dev/xvdb` (20.00MB) and that `/dev/xvdd` has 0.00MB).

d. Run the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
# blkid | grep btrfs
/dev/xvdb: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
/dev/xvdd: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
```

- Notice that `/dev/xvdd` has the same label and UUID as `/dev/xvdb`.
- The two devices do have different UUID_SUB values.

e. Run the `btrfs filesystem balance` command on the file system.

```
# btrfs file balance /btrfs
Done, had to relocate 2 out of 2 chunks
```

- After adding a device, it is recommended that you run this command on the file system to redistribute the chunks of the file system across all the devices.

f. Use the `btrfs file show` command to show the effect of the balance command.

```
# btrfs file show
Label: Btrfs'  uuid: ...
        Total devices 2 FS bytes used 4.30MB
        devid    2 size 10.00GB used 1.25GB path /dev/xvdd
        devid    1 size 9.00GB used 4.00MB path /dev/xvdb
```

```
Btrfs Btrfs v0.20-rc1
```

- Notice that both devices now have some data on them.

g. Use the `btrfs device delete` command to remove `/dev/xvdd` from the `/btrfs` file system.

```
# btrfs device delete /dev/xvdd /btrfs
```

h. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# df -h /btrfs
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvdb      9.0G  4.5M   8.8G    1%  /btrfs
```

- Notice that the size is decreased by 10 GB, from 19 GB down to 9 GB.

6. Complete the two-disk file system with the default RAID exercise.

a. Use the `umount` command to unmount `/btrfs`.

```
# umount /btrfs
```

b. Use the `mkfs.btrfs` command to make a Btrfs file system with two disks, `/dev/xvdb` and `/dev/xvdd`.

```
# mkfs.btrfs -f -L Btrfs /dev/xvd[bd]
WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
adding device /dev/xvdd id 2
fs created label Btrfs on /dev/xvdb
    nodesize 4096 leafsize 4096 sectorsize 4096 size 20.00GB
Btrfs Btrfs v0.20-rc1
```

- Notice that `/dev/xvdd` was added and that the file system size is now 20 GB.

c. Run the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
# blkid | grep btrfs
/dev/xvdb:  LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
/dev/xvdd:  LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
```

- Notice that both devices are listed, each having the same `UUID` but different `UUID_SUB` values.

d. Use the `mount` command to mount the file system.

```
# mount -a
```

- Notice that the original `/etc/fstab` entry, which references only one device, `/dev/xvdb`, still works.
- You need to reference only one device even when your file system contains multiple devices.
- You can reference either device, `/dev/xvdb` or `/dev/xvdd`, when mounting the file system.

e. Use the `cp` command to copy `/boot/vmlinuz-2.6.32*` to `/btrfs`.

```
# cp /boot/vmlinuz-2.6.32* /btrfs
```

f. Run the `sync` command, and then use the `df -h` command to display the space information for `/btrfs`.

```
# sync
# df -h /btrfs
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvdb       20G  4.8M    18G    1%  /btrfs
```

g. Use the `btrfs filesystem show` command to view the structure of the file system.

```
# btrfs file show
Label: 'Btrfs' uuid: ...
        Total devices 2 FS bytes used 4.73MB
        devid    2 size 10.00GB used 2.01GB path /dev/xvdd
        devid    1 size 10.00GB used 2.03GB path /dev/xvdb
Btrfs Btrfs v0.20-rc1
```

- Notice that the data is distributed evenly across the two devices.

h. Use the `btrfs filesystem df` command to show the space used on the mount point.

```
# btrfs file df /btrfs
Data, RAID0: total=2.00GB, used=4.69MB
Data, total=8.00MB, used=0.00
System, RAID1: total=8.00MB, used=4.00KB
System, total=4.00MB, used=0.00
Metadata, RAID1: total=1.00GB, used=32.00KB
Metadata, total=8.00MB, used=0.00
```

- Notice that, by default, the data is `RAID0` and the metadata is `RAID1`.

7. Complete the two-disk file system with RAID-1 for both the data and metadata exercise.

a. Use the `umount` command to unmount `/btrfs`.

```
# umount /btrfs
```

b. Use the `mkfs.btrfs` command to make a Btrfs file system with two disks, `/dev/xvdb` and `/dev/xvdd`, and specify RAID-1 for data.

- Metadata is RAID1 by default.

```
# mkfs.btrfs -f -L Btrfs -d raid1 /dev/xvd[bd]
WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
adding device /dev/xvdd id 2
fs created label Btrfs on /dev/xvdb
    nodesize 4096 leafsize 4096 sectorsize 4096 size 20.00GB
Btrfs Btrfs v0.20-rc1
```

c. Use the `mount` command to mount the file system.

```
# mount -a
```

d. Use the `cp` command to copy `/boot/vmlinuz-2.6.32*` to `/btrfs`.

```
# cp /boot/vmlinuz-2.6.32* /btrfs
```

e. Run the `sync` command, and then use the `df -h` command to display the space information for `/btrfs`.

```
# sync
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      20G  9.0M    18G    1%  /btrfs
```

- Notice that the output shows that the file system has a size of 20 GB.
- This is inaccurate because this is a RAID-1 array.

f. Use the `btrfs filesystem df` command to show the space used on the mount point.

```
# btrfs file df /btrfs
Data, RAID1: total=1.00GB, used=4.44MB
Data, total=8.00MB, used=0.00
System, RAID1: total=8.00MB, used=4.00KB
System, total=4.00MB, used=0.00
Metadata, RAID1: total=1.00GB, used=32.00KB
Metadata, total=8.00MB, used=0.00
```

- Notice that both the data and metadata are `RAID1`.
- You can also see that data is allocated in 1 GB chunks.

Practice 12-2: Working with Subvolumes and Snapshots

Overview

In this practice, you perform the following:

- Create a Btrfs subvolume.
- Create a snapshot of the subvolume.
- List the subvolume and snapshot.
- Mount the subvolume and snapshot.
- Delete the snapshot.
- Take a snapshot of a file by using the `cp --reflink` command.

Assumptions

- You are the `root` user on the **host03** VM.

Tasks

1. Create a Btrfs subvolume.
   a. Run the `btrfs sub` command to display the available `btrfs subvolume` commands.

```
# btrfs sub
usage: btrfs subvolume <command> <args>
    btrfs subvolume create [<dest>/]<name>
        Create a subvolume
    btrfs subvolume delete <subvolume> [<subvolume>...]
        Delete a subvolume
    btrfs subvolume list [-agopurts] [-G [+|-]value] ...
        List subvolumes (and snapshots)
    btrfs subvolume snapshot [-r] <source> [<dest>/]<name>
        Create a snapshot of the subvolume
    btrfs subvolume get-default <path>
        Get the default subvolume of a filesystem
    btrfs subvolume set-default <subvolid> <path>
        Set the default subvolume of a filesystem
    btrfs subvolume find-new <path> <lastgen>
        List the recently modified files in a filesystem
    btrfs subvolume show <subvol-path>
        Show more information of the subvolume
```

b. Use the `btrfs sub create` command to create a subvolume named `SV1`.

```
# btrfs sub create /btrfs/SV1
Create subvolume '/btrfs/SV1'
```

c. Use the `ls –l` command to display the contents of `/btrfs`.

```
# ls –l /btrfs
drwxr-xr-x ... SV1
-rwxr-xr-x ... vmlinuz-2.6.32...
```

- Notice that the `SV1` subvolume is created and appears as a regular directory.

d. Use the `mv` command to move the `vmlinuz*` file to the `/btrfs/SV1` subvolume.

```
# mv /btrfs/vmlinuz* /btrfs/SV1
```

e. Use the `ls –lR` command to display the contents of `/btrfs`.

```
# ls –lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-2.6.32...
```

- Notice that the `vmlinuz*` file now resides in the `SV1` subvolume.

f. Run the `sync` command, and then use the `df –h` command to display the disk space usage on the `/btrfs` file system.

```
# sync
# df –h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      20G  9.0M    18G    1%  /btrfs
```

2. Create a Btrfs snapshot.

   a. Use the `btrfs sub snapshot` command to create a snapshot of the `SV1` subvolume. Name the snapshot `SV1-snap`.

```
# btrfs sub snapshot /btrfs/SV1 /btrfs/SV1-snap
Create a snapshot of '/btrfs/SV1' in '/btrfs/SV1-snap'
```

   b. Use the `ls –lR` command to display the contents of `/btrfs`.

```
# ls –lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1
drwxr-xr-x ... SV1-snap

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-2.6.32...

/btrfs/SV1-snap:
-rwxr-xr-x ... vmlinuz-2.6.32...
```

   - Notice that the new `SV1-snap` directory in `/btrfs` is created.
   - The `SV1-snap` snapshot is a point-in-time copy of the `SV1` subvolume.

   c. Run the `sync` command, and then use the `df –h` command to display disk space usage on the `/btrfs` file system.

```
# sync
# df –h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      20G  9.0M    18G    1%  /btrfs
```

   - Notice that the creation of the snapshot did not reduce the available disk space.

   d. Use the `cp` command to copy `/boot/vmlinuz-3.8.13-16*` to the `/btrfs/SV1` subvolume.

```
# cp /boot/vmlinuz-3.8.13-16* /btrfs/SV1
```

   e. Use the `ls –lR` command to display the contents of `/btrfs`.

```
# ls –lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1
drwxr-xr-x ... SV1-snap

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-2.6.32...
-rwxr-xr-x ... vmlinuz-3.8.13...

/btrfs/SV1-snap:
-rwxr-xr-x ... vmlinuz-2.6.32...
```

   - Notice that the `SV1` subvolume now contains two `vmlinuz*` files.

3. Mount the subvolume and the snapshot.
   a. Use the `btrfs sub list` command to list the subvolumes and snapshots.
      - Output is sample only. The `ID` and `gen` numbers may be different.

```
# btrfs sub list /btrfs
ID 258 gen 12 top level 5 path SV1
ID 259 gen 11 top level 5 path SV1-snap
```

      - In this example, the ID of the `SV1` subvolume is `258`.
      - The ID of the `SV1-snap` subvolume (snapshot) s `259`.
      - The IDs of your subvolumes might be different. Substitute your IDs as necessary in the following steps.
   b. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
# btrfs sub get-default /btrfs
ID 5 (FS_TREE)
```

      - Notice that the ID of `5` indicates that the root subvolume is the default.
   c. Use the `btrfs sub set-default` command to set the subvolume ID to `258` (ID of the `SV1` subvolume).

```
# btrfs sub set-default 258 /btrfs
```

   d. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
# btrfs sub get-default /btrfs
ID 258 gen 13 top level 5 path SV1
```

      - Notice that the default subvolume ID is `258` for the `SV1` subvolume.
   e. Use the `umount` command to unmount the file system mounted on `/btrfs`.

```
# umount /btrfs
```

   f. Use the `mount` command to mount the file system.

```
# mount -a
```

   g. Use the `ls –lR` command to display the contents of `/btrfs`.

```
# ls –lR /btrfs
/btrfs/:
-rwxr-xr-x ... vmlinuz-2.6.32...
-rwxr-xr-x ... vmlinuz-3.8.13...
```

      - Notice that the contents of the `SV1` subvolume are now mounted on `/btrfs`.
   h. Use the `btrfs sub set-default` command to set the subvolume ID to `259` (ID of the `SV1-snap` snapshot).

```
# btrfs sub set-default 259 /btrfs
```

i. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
# btrfs sub get-default /btrfs
ID 259 gen 11 top level 5 path SV1-snap
```

- Notice that the default subvolume ID is `259` for the `SV1-snap` subvolume.

j. Use the `umount` command to unmount the file system mounted on `/btrfs`.

```
# umount /btrfs
```

k. Use the `mount` command to mount the file system.

```
# mount -a
```

l. Use the `ls –lR` command to display the contents of `/btrfs`.

```
# ls –lR /btrfs
/btrfs/:
-rwxr-xr-x ... vmlinuz-2.6.32...
```

- Notice that the contents of the `SV1-snap` snapshot are now mounted on `/btrfs`.

4. Mount the root subvolume.

a. Use the `btrfs sub set-default` command to set the subvolume ID to `5` (ID of the root subvolume).

```
# btrfs sub set-default 5 /btrfs
```

b. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
# btrfs sub get-default /btrfs
ID 5 (FS_TREE)
```

- Notice that the default subvolume ID is `5` for the root subvolume.

c. Use the `umount` command to unmount the file system mounted on `/btrfs`.

```
# umount /btrfs
```

d. Use the `mount` command to mount the file system.

```
# mount -a
```

e. Use the `ls –lR` command to display the contents of `/btrfs`.

```
# ls –lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1
drwxr-xr-x ... SV1-snap

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-2.6.32...
-rwxr-xr-x ... vmlinuz-3.8.13...

/btrfs/SV1-snap:
-rwxr-xr-x ... vmlinuz-2.6.32...
```

- Notice that the root subvolume is now mounted on `/btrfs`.

- You could also mount the root file system as follows, but this does not change the default subvolume ID (do not perform this command, this is information only):
  `# mount -o subvolid=5 /dev/xvdb /btrfs`
- ID of `0` can also be used to mount the root subvolume.

5. Delete the snapshot.

   a. Use the `btrfs sub delete` command to delete the snapshot.

```
# btrfs sub delete /btrfs/SV1-snap
Delete subvolume '/btrfs/SV1-snap'
```

   b. Use the `btrfs sub list` command to list the subvolumes and snapshots.

   - Rerun this command if an error is encountered.

```
# btrfs sub list /btrfs
ID 258 gen 13 top level 5 path SV1
```

   - Notice that the `SV1-snap` snapshot is removed.

   c. Use the `ls -lR` command to display the contents of `/btrfs`.

```
# ls -lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-2.6.32...
-rwxr-xr-x ... vmlinuz-3.8.13...
```

   - Notice that the `SV1-snap` directory is removed.

6. Take a snapshot of a file.

   a. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# sync
# df -h /btrfs
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvdb       20G   18M    18G    1%  /btrfs
```

   b. Use the `cd` command to change to the `/btrfs/SV1` directory.

```
# cd /btrfs/SV1
```

   c. Use the `cp` command to copy the `vmlinuz-2.6.32*` file to `copy_of_32`.

```
# cp vmlinuz-2.6.32* copy_of_32
```

d. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# sync
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      20G   26M    18G    1%  /btrfs
```

- Notice that the "Used" amount increased from 18M to 26M.

e. Use the `rm` command to remove the `copy_of_32` file.

```
# rm copy_of_32
rm: remove regular file 'copy_of_32'? y
```

f. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# sync
# df -h /btrfs
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvdb      20G   18M    18G    1%  /btrfs
```

- Notice that the "Used" amount decreased from 26M to 18M.

g. Use the `cp --reflink` command to copy the `vmlinuz-2.6.32*` file to `copy_of_32`.

```
# cp --reflink vmlinuz-2.6.32* copy_of_32
```

h. Use the `ls -l` command to display the sizes of the original file and the copy.

```
# ls -l
-rwxr-xr-x ... 4128944  ...  copy_of_32
-rwxr-xr-x ... 4128944  ...  vmlinuz-2.6.32...
-rwxr-xr-x ... 4588304  ...  vmlinuz-3.8.13...
```

- Notice that the size of the `vmlinuz-2.6.32*` file and the size of the `copy_of_32` file is the same.

i. Use the `diff` command to compare the original file and the copy.

```
# diff vmlinuz-2.6.32* copy_of_32
```

- No output indicates that there are no differences in the two files.

j. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
# sync
# df -h /btrfs
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvdb       20G   18M    18G    1%  /btrfs
```

- Notice that the "Used" amount remained at 18M even after making a copy.
- The data blocks are not duplicated when the `--reflink` option is used.
- This allows the copy to be almost instantaneous and also saves disk space.
- One restriction is that this operation works only within the boundaries of the same file system and within the same subvolume.

Practice 12-3: Recovering from Data Corruption

Overview

In this practice, you perform the following:

- Prepare the environment by recreating a RAID-1 file system.
- Use the `btrfs-corrupt-block` utility to induce data corruption.
- Use the `btrfs scrub` command to recover from data corruption.

Assumptions

- You are the `root` user on the **dom0**.
- You are the `root` user on the **host03** VM.

Tasks

1. Prepare the environment.
   a. From **host03**, use the `cd` command to change to the `root` user home directory, and then use the `umount` command to unmount `/btrfs`.

   ```
   # cd
   # umount /btrfs
   ```

   b. Use the `mkfs.btrfs` command to make a Btrfs file system with two disks; `/dev/xvdb` and `/dev/xvdd`, and specify RAID-1 for data.

   ```
   # mkfs.btrfs -f -L Btrfs -d raid1 /dev/xvd[bd]
   WARNING! - Btrfs Btrfs v0.20-rc1 IS EXPERIMENTAL
   WARNING! - see http://btrfs.wiki.kernel.org before using
   adding device /dev/xvdd id 2
   fs created label Btrfs on /dev/xvdb
       nodesize 4096 leafsize 4096 sectorsize 4096 size 20.00GB
   Btrfs Btrfs v0.20-rc1
   ```

   c. Use the `mount` command to mount the file system.

   ```
   # mount -a
   ```

   d. Use the `cp` command to copy `/boot/vmlinuz-2.6.32*` to `/btrfs`.

   ```
   # cp /boot/vmlinuz-2.6.32* /btrfs
   ```

   e. Run the `sync` command, and then use the `df -h` command to display space information for `/btrfs`.

   ```
   # sync
   # df -h /btrfs
   Filesystem    Size  Used  Avail  Use%  Mounted on
   /dev/xvdb      20G  9.0M   18G     1%  /btrfs
   ```

f. Use the `btrfs filesystem show` command to view the structure of the file system.

```
# btrfs file show
Label: 'Btrfs' uuid: ...
        Total devices 2 FS bytes used 4.48MB
        devid    2 size 10.00GB used 2.01GB path /dev/xvdd
        devid    1 size 10.00GB used 2.03GB path /dev/xvdb
Btrfs Btrfs v0.20-rc1
```

2. Induce data corruption on the Btrfs file system.

Perform these steps from **dom0**.

a. From **dom0**, open a new terminal window and use the `su –` command to become the `root` user (password is `oracle`) on **dom0**.

```
[dom0]$ su –
Password: oracle
[dom0]#
```

b. As the `root` user on **dom0**, use the `cd` command to change to the `/OVS/seed_pool/btrfs` directory. Use the `ls` command to list the directory contents.

```
[dom0]# cd /OVS/seed_pool/btrfs
[dom0]# ls
btrfs-corrupt-block
```

c. From **dom0**, use the `sftp` command to connect to **host03** as `root`. Password is `oracle`.

```
# sftp host03
Connecting to host03...
root@host03's password: oracle
sftp>
```

d. From the `sftp>` prompt on **dom0**, use the `put` command to copy `btrfs*` to **host03**. Use the `quit` command to close the connection after copying the file.

```
sftp> put *
Uploading btrfs-corrupt-block to /root/btrfs-corrupt-block
btrfs-corrupt-block ...
sftp> quit
```

Perform remaining steps from **host03**.

e. From **host03**, use the `cd` command to change to the `/btrfs` directory, and then run the `filefrag` command to report on file fragmentation on the `vmlinuz-2.6.32*` file.

```
# cd /btrfs
# filefrag –v vmlinuz-2.6.32*
Filesystem type is 9123683e
File size of vmlinuz-2.6.32... is 4128944 (1009 blocks of 4096
bytes)
 ext: logical_offset:  physical_offset: length: expected: flags:
   0:    0..    1008: 269312..  270320:   1009:          eof
vmlinuz-2.6.32...: 1 extent found
```

- In this example, the file is on physical block `269312` and has a 4K (`4096`) block size.

f. Use a calculator to multiply the physical block by the block size.

1) Multiply `4096 * 269312 = 1103101952`.
2) In this example, `1103101952` is the amount that you want to corrupt.

g. Use the `cd` command to change to your home directory, and then use the `umount` command to unmount `/btrfs`.

```
# cd
# umount /btrfs
```

h. Use the `btrfs-corrupt-block` tool to corrupt the first copy of that block.

- This is a very dangerous tool, and is not part of the `btrfs-progs` RPM.
- By default, this tool corrupts all the copies of the block.
- You want to corrupt only the first copy of the block.

```
# ./btrfs-corrupt-block –c 1 -l 1103101952 /dev/xvdb
mirror 1 logical 1103101952 physical 1083179008 device /dev/xvdd
corrupting 1103101952 copy 1
mirror 2 logical 1103101952 physical 1103101952 device /dev/xvdb
```

3. Mount the file system and scrub.

a. Use the `mount` command to mount the file system.

```
# mount -a
```

b. Use the `btrfs scrub` command to start a file system scrub.

```
# btrfs scrub start /btrfs
scrub started on /btrfs/, fsid ... (pid=...)
```

c. Use the `btrfs scrub` command to get the status of the file system scrub.

```
# btrfs scrub status /btrfs
Scrub status for ...
        scrub started at ... and finished after 0 seconds
        total bytes scrubbed: 8.95MB with 1 errors
        error details: csum=1
        corrected errors: 1, uncorrectable errors: 0,
        unverified errors: 0
```

- Notice that one error was found, a checksum error, and one error was corrected.

d. Use the `dmesg` command to determine what happened.

```
# dmesg
...
btrfs: checksum error at logical 1103101952 on dev /dev/xvdd,
sector 2115584, root 5, inode 257, offset 0, length 4096, links
1 (path: vmlinuz-2.6.32-431.el6.x86_64)
btrfs: bdev /dev/xvdd errs: wr 0, rd 0, flush 0, corrupt 1, gen
0
btrfs: fixed up error at logical 1103101952 on dev /dev/xvdd
```

- Notice that the error induced by the `btrfs-corrupt-block` command at logical `1103101952` was corrected by the scrub.

e. Use the `btrfs scrub` command to start a file system scrub.

```
# btrfs scrub start /btrfs
scrub started on /btrfs/, fsid ... (pid=...)
```

f. Use the `btrfs scrub` command to get the status of the file system scrub.

```
# btrfs scrub status /btrfs
Scrub status for ...
        scrub started at ... and finished after 0 seconds
        total bytes scrubbed: 8.95MB with 0 errors
```

- Notice that 0 errors were found.

4. Prepare for future practices.

a. Use the `umount` command to unmount `/btrfs`.

```
# umount /btrfs
```

b. Use the `vi` editor to remove the following entry in `/etc/fstab`:

```
# vi /etc/fstab
...
/dev/xvdb  /btrfs  btrfs  defaults  0  0     (delete this entry)
```

c. Use the `exit` command to log off of **host03**.

```
# exit
logout
Connection to host03 closed.
```

Practice 12-4: Creating a Btrfs `root` File System by Installing Oracle Linux from the UEK Boot ISO

Overview

In this practice, you perform the following:

- Configure **dom0** as an NFS network installation server.
- Install Oracle Linux 6.5 on **host06** by using the UEK Boot ISO.
- Log in to **host06** and explore the Btrfs root file system.
- Shut down the **host06** VM.

Assumptions

- The Oracle Linux 6 update 5 DVD image file exists in `/OVS/seed_pool` on **dom0**.
- The UEK Boot ISO image file exists in `/OVS/seed_pool` on **dom0**.
- You are the `root` user on **dom0**.

Tasks

1. Configure **dom0** as an NFS server.
   a. On **dom0**, use the `mkdir -p` command to make the `/OVS/OSimage/OL6.5` directory

   ```
   # mkdir -p /OVS/OSimage/OL6.5
   ```

   b. Use the `vi` editor to add the following entry to the `/etc/exports` file.

   ```
   # vi /etc/exports
   /OVS/OSimage/OL6.5 192.0.2.0/24(ro)
   ```

   c. Use the `service` command to start the `nfs` service and the `nfslock` service.

   ```
   # service nfs start
   ...
   # service nfslock start
   ...
   ```

   d. Use the `showmount` command to display the exported NFS file system.
   - `/OVS` is a symbolic link to `/var/ovs/mount/...`

   ```
   # showmount -e
   Export list for ...:
   /var/ovs/mount/.../OSimage/OL6.5  192.0.2.0/24
   ```

2. Copy the contents of the full Oracle Linux 6 update 5 DVD image to the NFS file system.
   a. On **dom0**, use the `cd` command to change to the `/OVS/seed_pool` directory. Use the `ls` command to display the contents of the directory.

```
# cd /OVS/seed_pool
# ls
...
OracleLinux-R6-U5-Server-x86_64-dvd.iso
V41364-01.iso
...
```

   - The `OracleLinux-R6-U5-Server-x86_64-dvd.iso` file is the full Oracle Linux 6 update 5 DVD image.
   - The `V41364-01.iso` file is the UEK Boot ISO image.

   b. Use the `mkdir` command to make the `/mnt/iso` directory.
   - Using a temporary mount point other than `/mnt` is a requirement imposed by Oracle University (OU). On OU systems, a FAT file system created is mounted in `/mnt/cdrive`. This file system holds binaries that monitor the machine status and take care of initiating the build for the next class after the current class is finished. If you are mounting an ISO on `/mnt`, it mounts on top of `/mnt/cdrive`. This causes the binaries to fail to report to the OU Dashboard. Outside of the OU environment, you can use `/mnt` for this procedure.

```
# mkdir /mnt/iso
```

   c. Use the `mount` command to mount the full Oracle Linux 6.5 DVD image on `/mnt`.

```
# mount -t iso9660 -o loop OracleLinux-R6-U5-Server-x86_64-
dvd.iso /mnt/iso
```

   d. Use the `cp` command to copy all files and directories from `/mnt/iso` to `/OVS/OSimage/OL6.5`.
   - This command takes a few minutes to complete.

```
# cp –r /mnt/iso/* /OVS/OSimage/OL6.5
```

   e. Use the `umount` command to unmount `/mnt/iso`.

```
# umount /mnt/iso
```

3. Replace the contents of the `/OVS/OSimage/OL6.5/images` directory with the contents of the `images` directory from the UEK Boot ISO image.
   a. Use the `mount` command to mount the UEK Boot ISO image on `/mnt/iso`.

```
# mount -t iso9660 -o loop V41364-01.iso /mnt/iso
```

   b. Use the `rm` command to remove the contents of the `images` directory in `/OVS/OSimage/OL6.5`.

```
# rm –rf /OVS/OSimage/OL6.5/images/*
```

   - Default Oracle Linux 6.5 images do not support Btrfs file systems.

c. Use the `cp` command to copy the `images` directory from the UEK Boot ISO, which is mounted on `/mnt/iso`.

- This command takes a few seconds to complete.

```
# cp -r /mnt/iso/images /OVS/OSimage/OL6.5
```

d. Use the `umount` command to unmount `/mnt/iso`. Use the `rmdir` command to remove the `/mnt/iso` directory.

```
# umount /mnt/iso
# rmdir /mnt/iso
```

4. Boot **host06** using the UEK Boot ISO.

a. From **dom0**, use the `cd` command to change to the `/OVS/running_pool/host06` directory. Use the `ls` command to display the contents of the directory.

```
# cd /OVS/running_pool/host06
# ls -l
-rw-r--r--  12884901888  system.img
-rw-r--r--          502  vm.cfg
```

- The `system.img` file represents a 12 GB virtual disk on which you install Oracle Linux.
- The `vm.cfg` file is the configuration file for the virtual machine. This file is read when the virtual machine is created.

b. Use the `cat` command to view the `vm.cfg` file.

```
# cat vm.cfg
# Automatically generated xen config file
name = "host06"
builder = "hvm"
memory = "1536"
boot = 'cd'
disk = [ 'file:/OVS/running_pool/host06/system.img,hda,w',
         'file:/OVS/seed_pool/V41364-01.iso,hdc:cdrom,r']
vif = [ 'mac=00:16:3e:00:01:06, bridge=virbr0']
device_model = "/usr/lib/xen/bin/qemu-dm"
kernel = "/usr/lib/xen/boot/hvmloader"
vnc=1
vncunused=1
vcpus=1
timer_mode=0
apic=1
acpi=1
pae=1
serial = "pty" # enable serial console
on_reboot   = 'restart'
on_crash    = 'restart'
usb = 1
usbdevice = 'tablet'
```

- Notice the one virtual disk represented by the `system.img` file.
- Notice that the UEK Boot ISO is mounted on a virtual cdrom device.
- Notice that there is one virtual network interface.

c. Use the `xm shutdown -w host03` command to shut down the **host03** VM.

- There is not enough memory to create a fourth VM; therefore, one VM must be shut down before creating another.

```
# xm shutdown -w host03
Domain host03 terminated
All domains terminated
```

d. Run the `xm create vm.cfg` command to start the **host06** VM.

```
# xm create vm.cfg
...
```

e. Determine the VNC port number for **host06** by running the `xm list -l host06 | grep location` command.

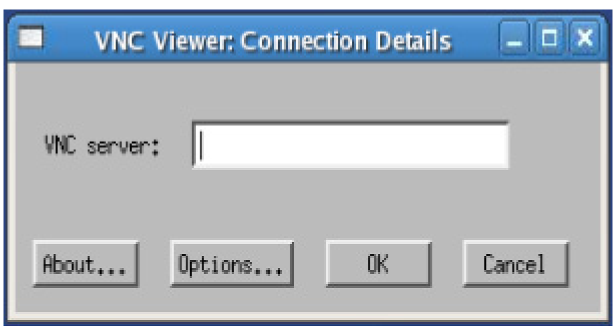
```
# xm list -l host06 | grep location
                (location 0.0.0.0:5903)
                (location 3)
```

- The sample shown indicates that the port number is `5903`. This might not be true in your case.

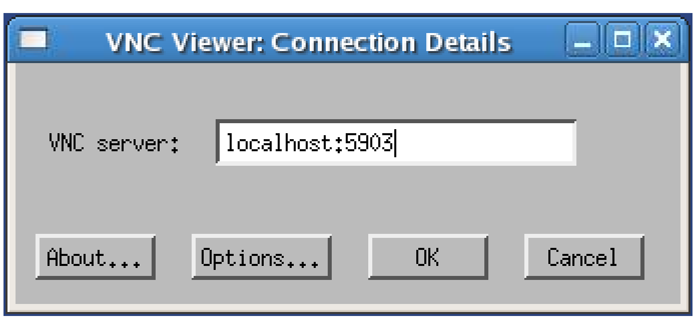
f. Run the `vncviewer&` command.

```
# vncviewer&
```

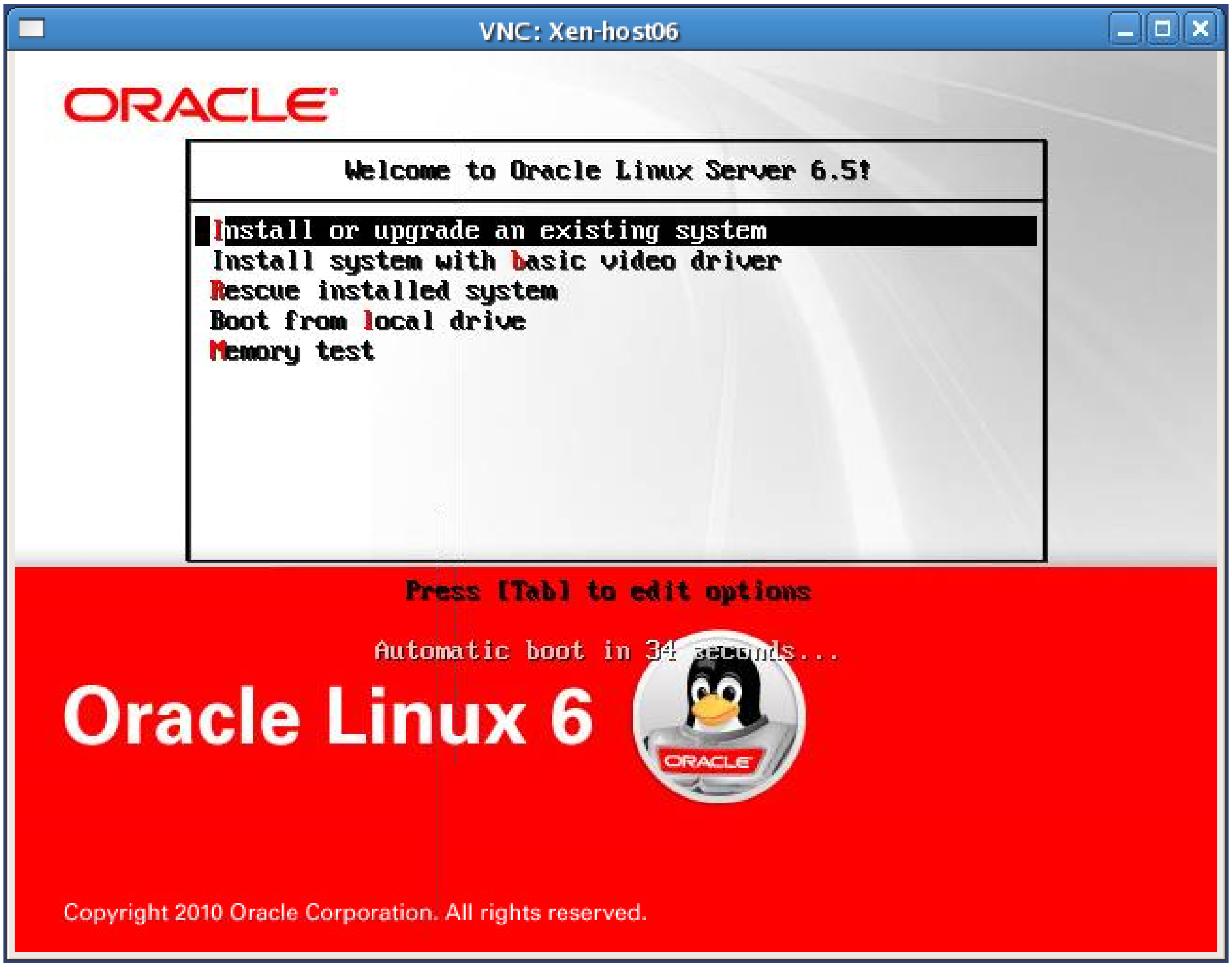
- The **VNC Viewer: Connection Details** dialog box is displayed.



g. Enter `localhost:`*`<port_number>`*, substituting the port number displayed from the previous `xm list -l host06 | grep location` command. For example, if the port number is `5903`, enter `localhost:5903` and click **OK**.



- The Oracle Linux boot menu appears:



- The Oracle Linux boot menu screen appears for only 60 seconds after which the "Install or upgrade an existing system" menu option is selected by default.
- If you do not see this screen, meaning the 60 second timeout has expired, click the `X` in the top-right corner of the screen to close it, enter the following command from **dom0**, and begin step 4 again starting with 4d.

```
# xm destroy host06
```

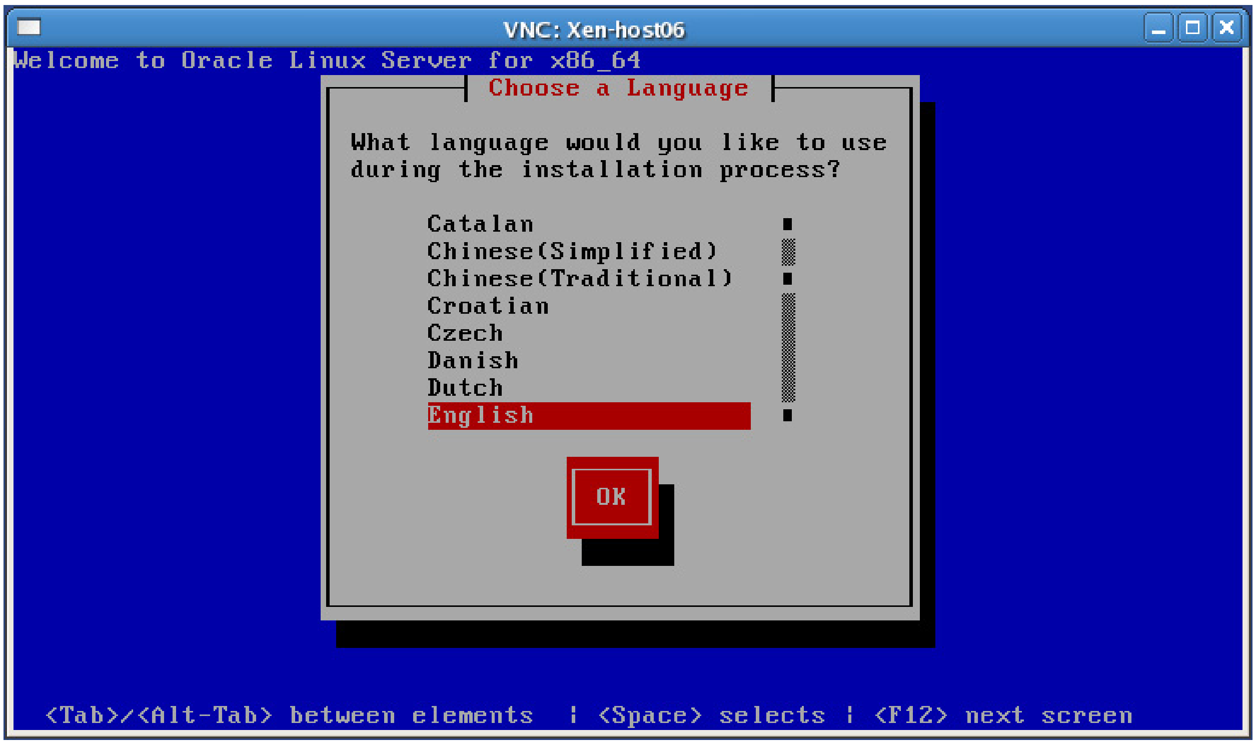
5. Supply the `askmethod` parameter on the command line at the Oracle Linux boot menu.
   a. With the "Install or upgrade an existing system" menu option selected, press the **Tab** key. The following boot command line appears:

```
> vmlinuz initrd=initrd.img
```

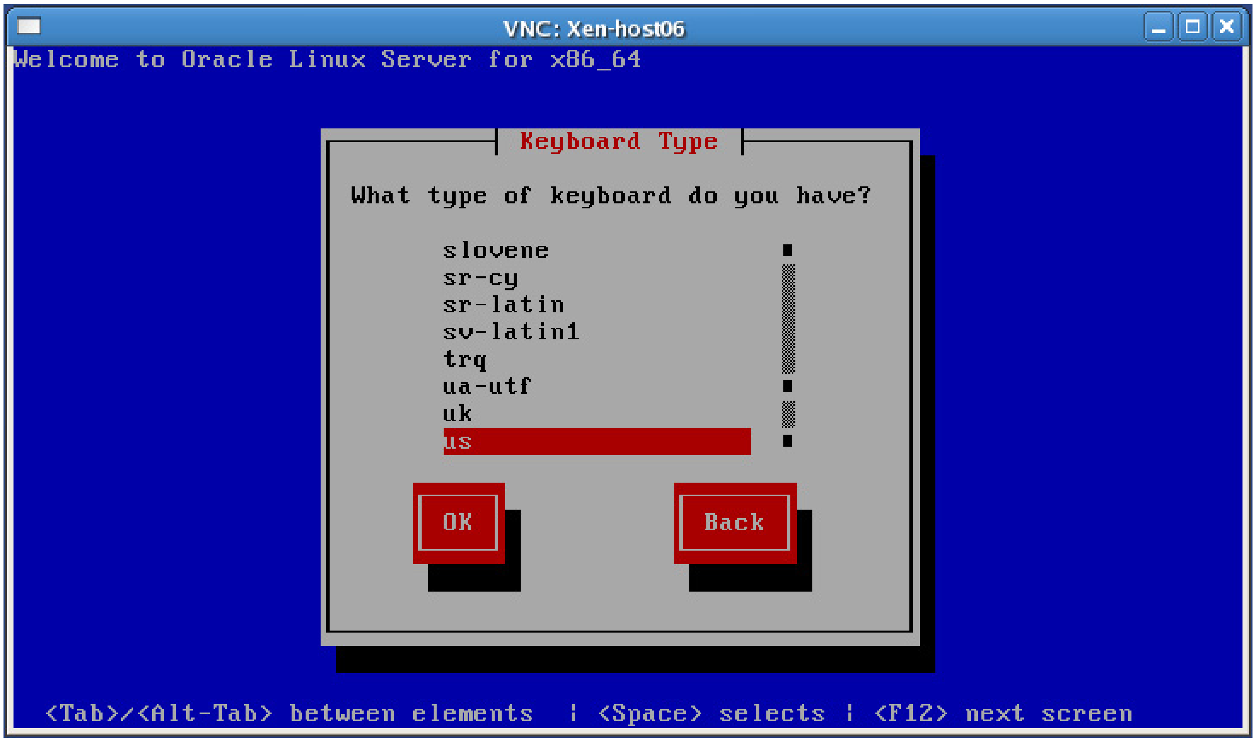
   b. Enter `askmethod` as an additional parameter on the boot command line and press Return.

```
> vmlinuz initrd=initrd.img askmethod
```

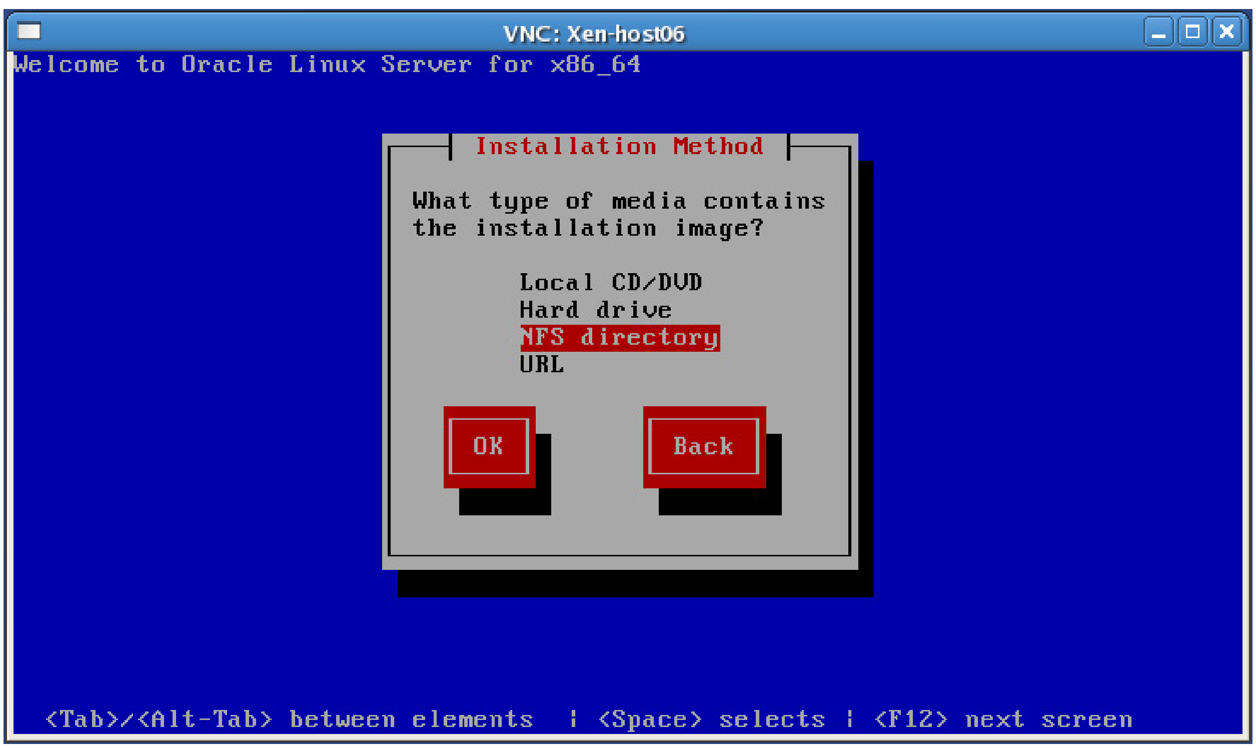
6. Complete the installation.
   a. Select a language on the following screen:



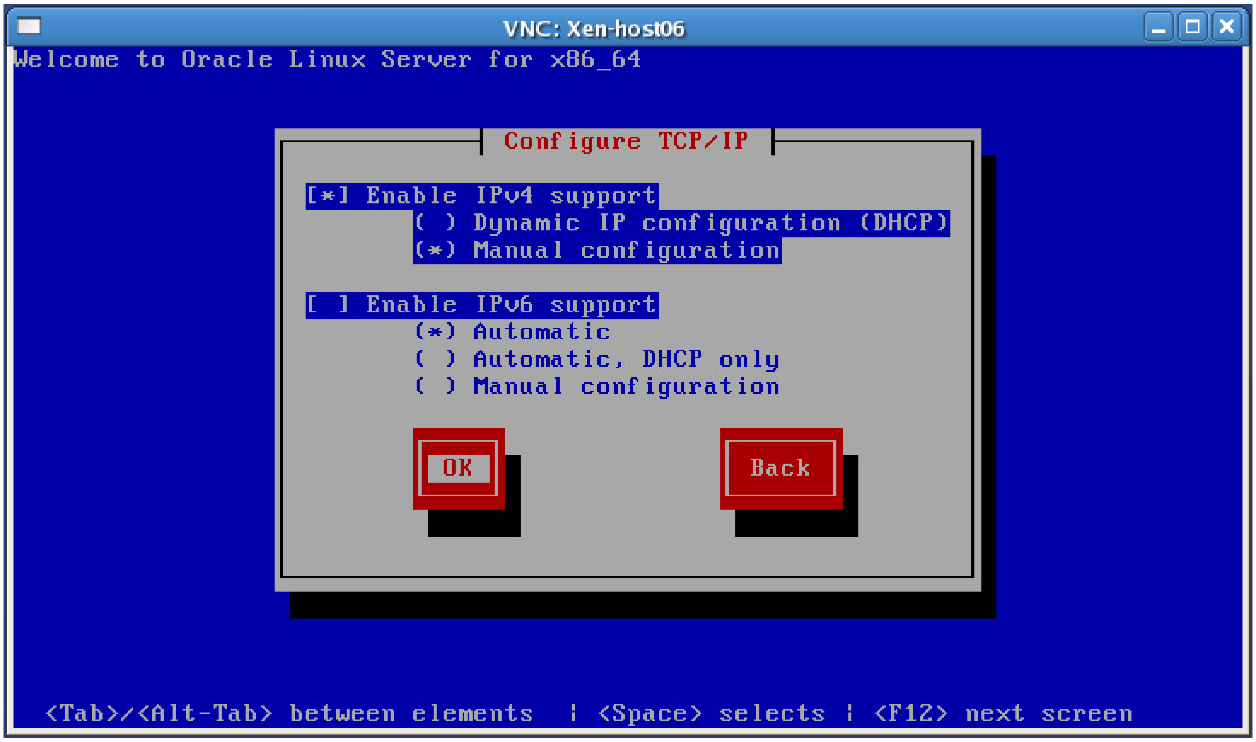
b. Select a keyboard type on the following screen:



c. Select **NFS directory** as the Installation Method on the following screen:

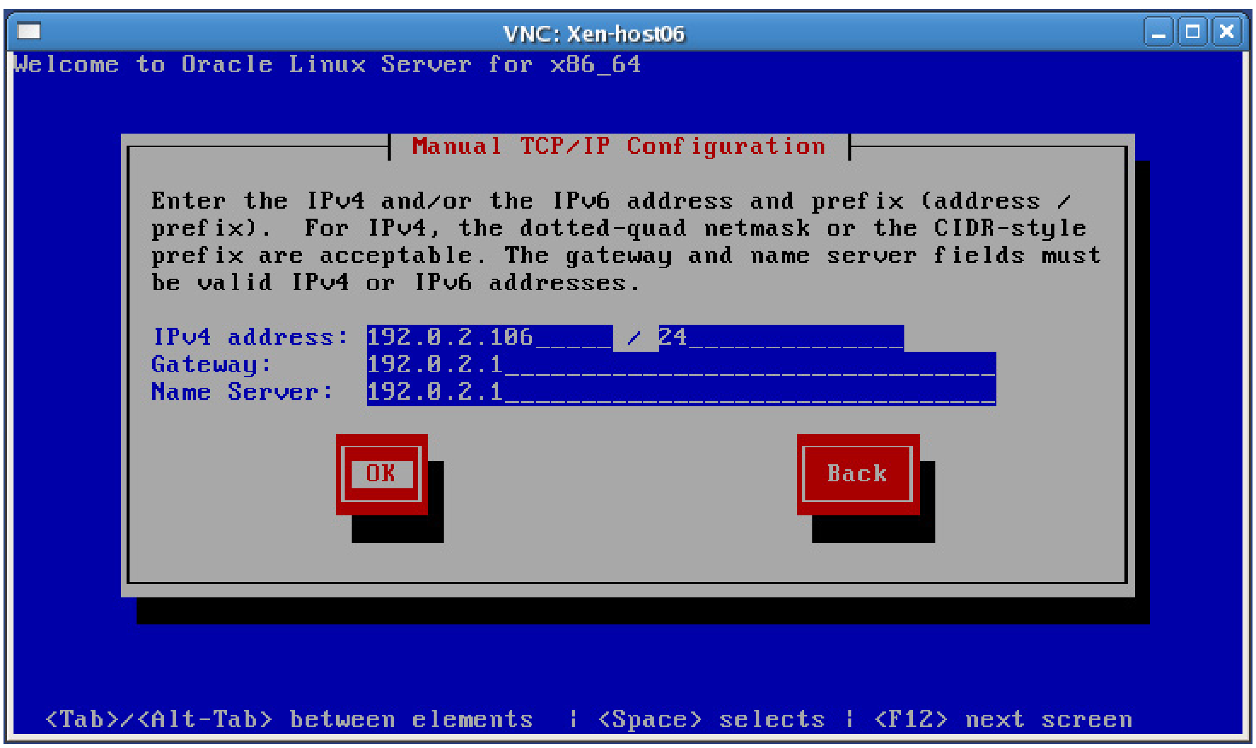


d. Manually configure IPv4 and disable IPv6 as shown on the following screen:



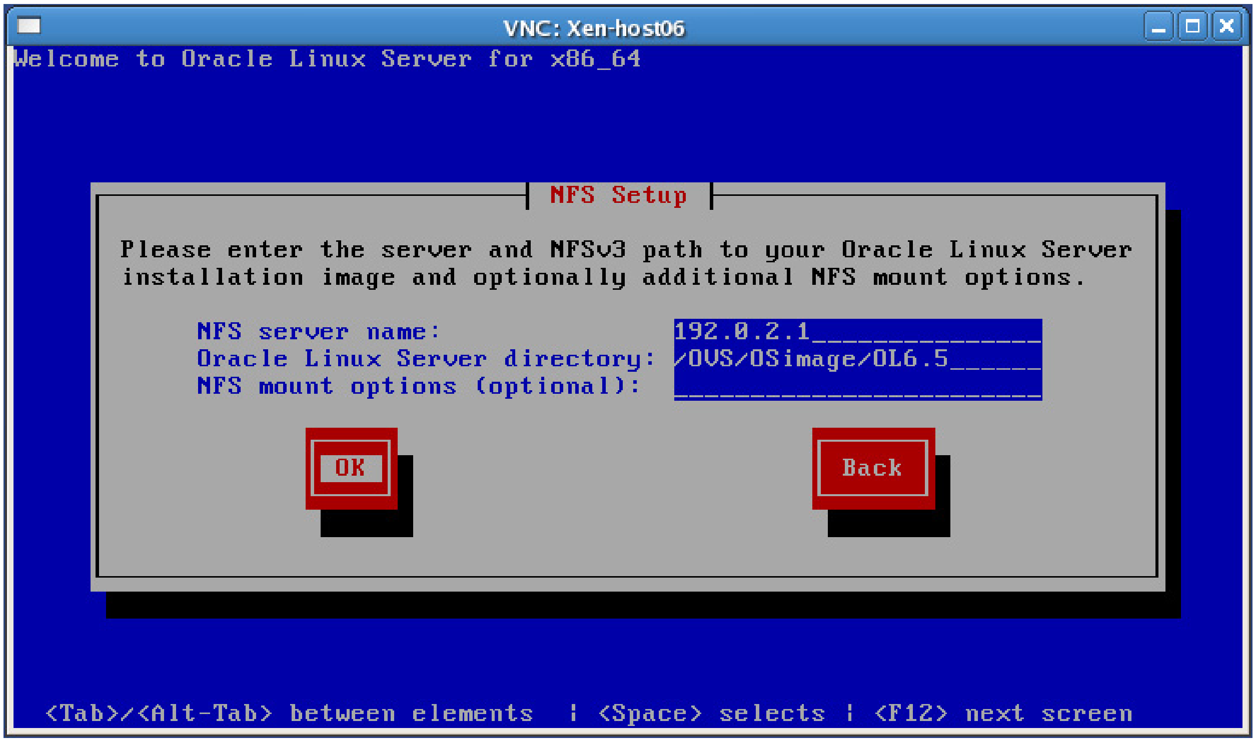
e. Configure the network as shown:

- IPv4 address: `192.0.2.106/24`
- Gateway: `192.0.2.1`
- Name Server: `192.0.2.1`

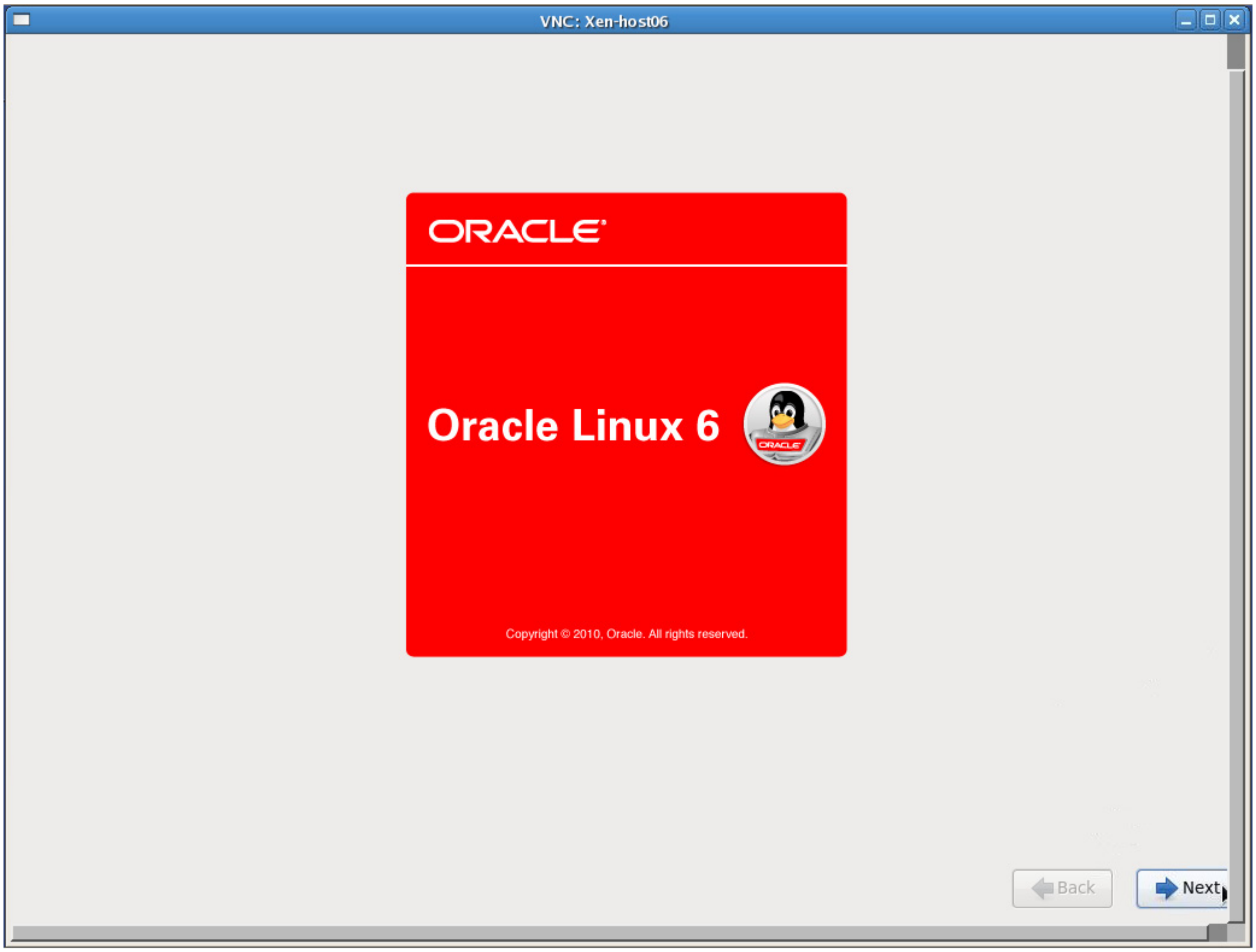


f. Configure the settings for the NFS installation server as shown:

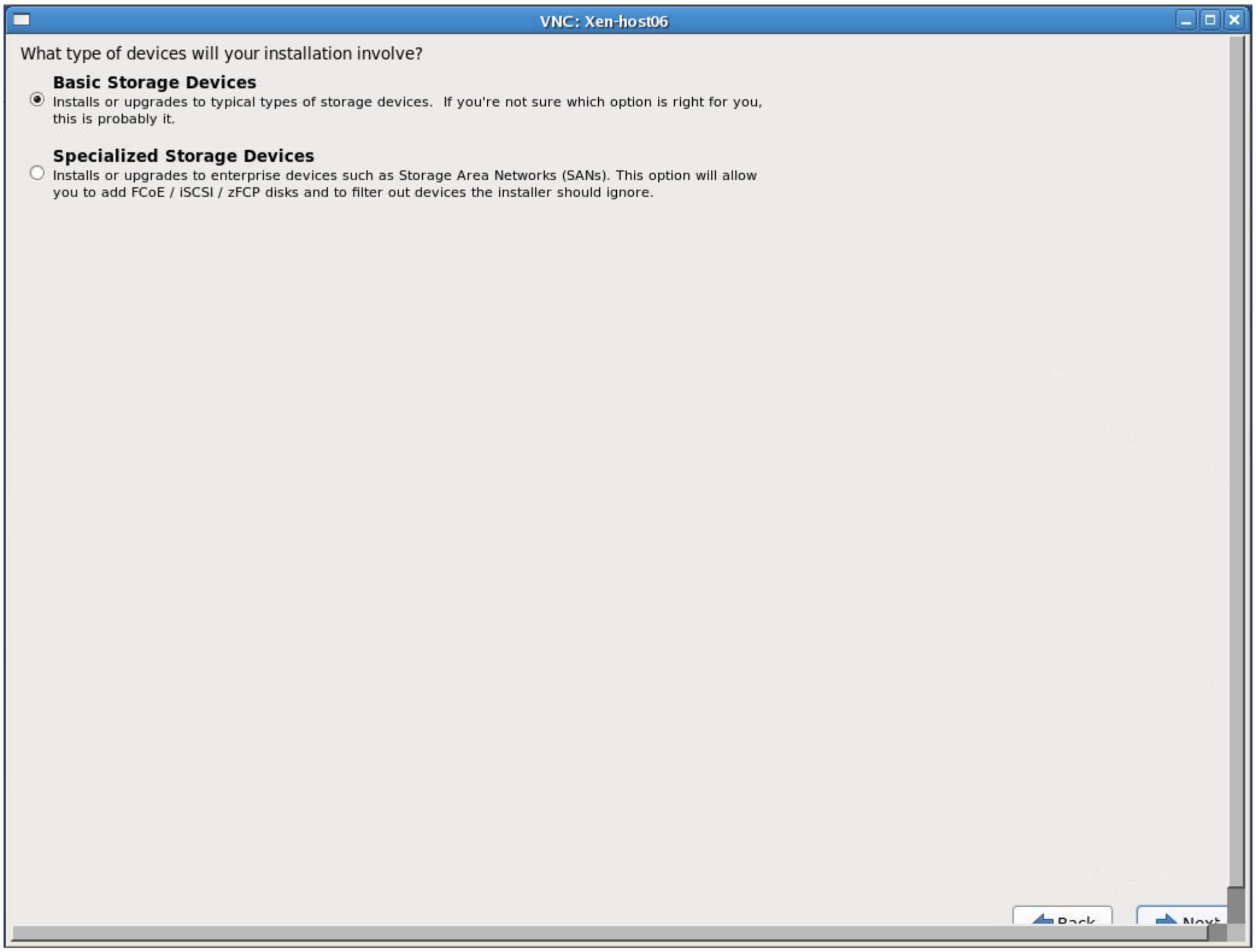
- NFS server name (IP address of **dom0**): `192.0.2.1`
- Oracle Linux Server directory: `/OVS/OSimage/OL6.5`



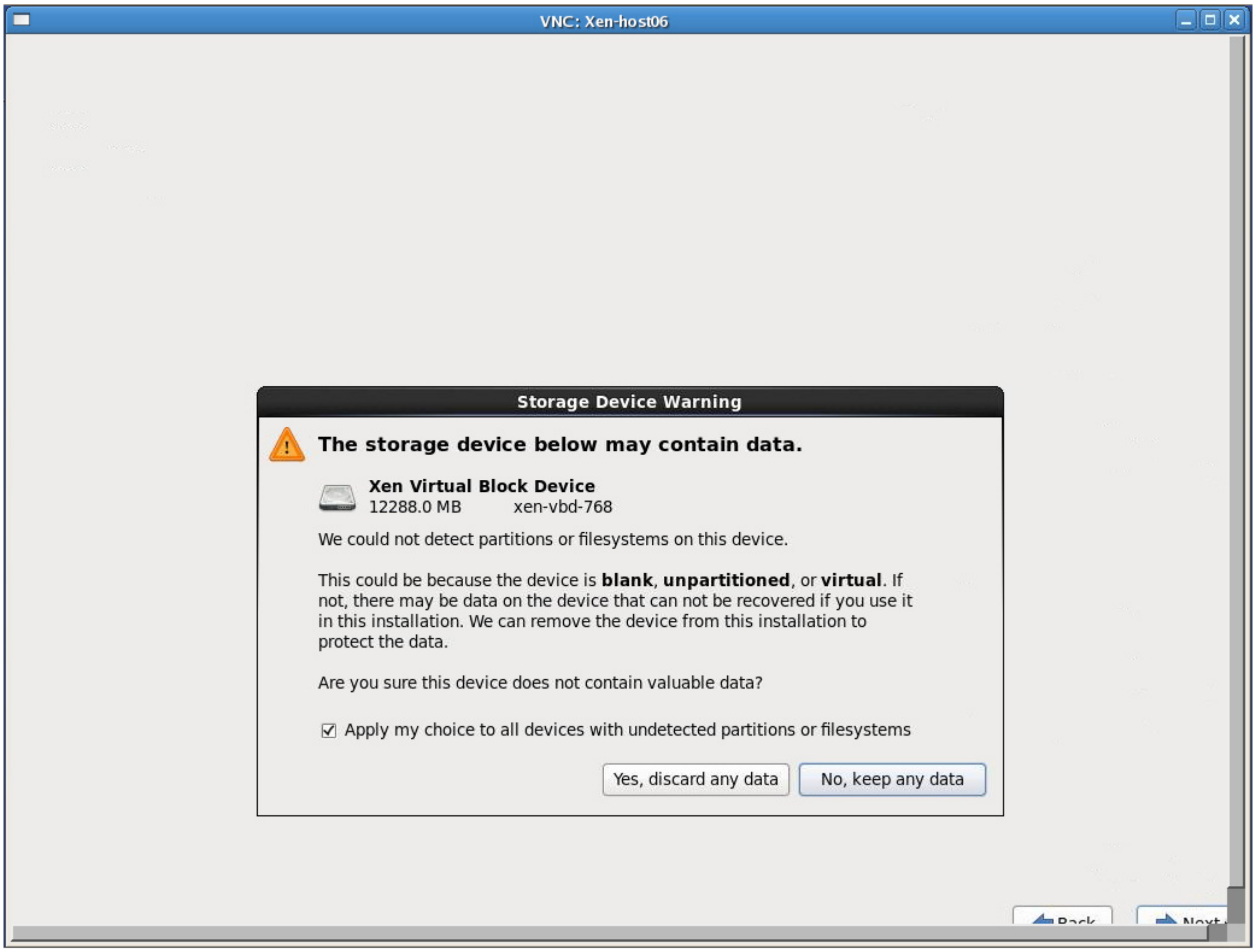
g. The Logo screen appears. Scroll down if necessary and click **Next**.



h. Select **Basic Storage Devices** on the following screen. Scroll down if necessary and click **Next**.

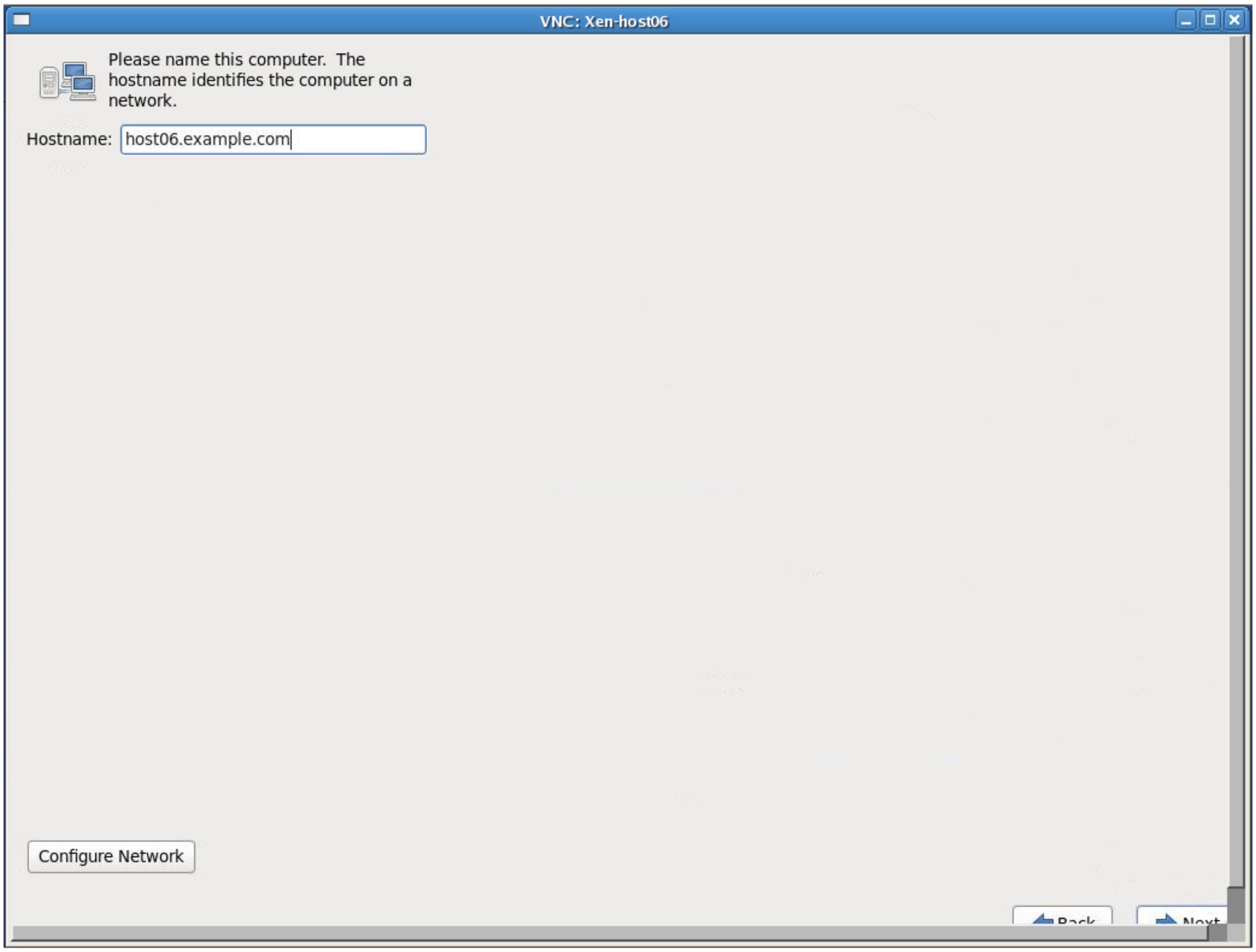


i. The Storage Device Warning screen appears. Click "**Yes, discard any data**."



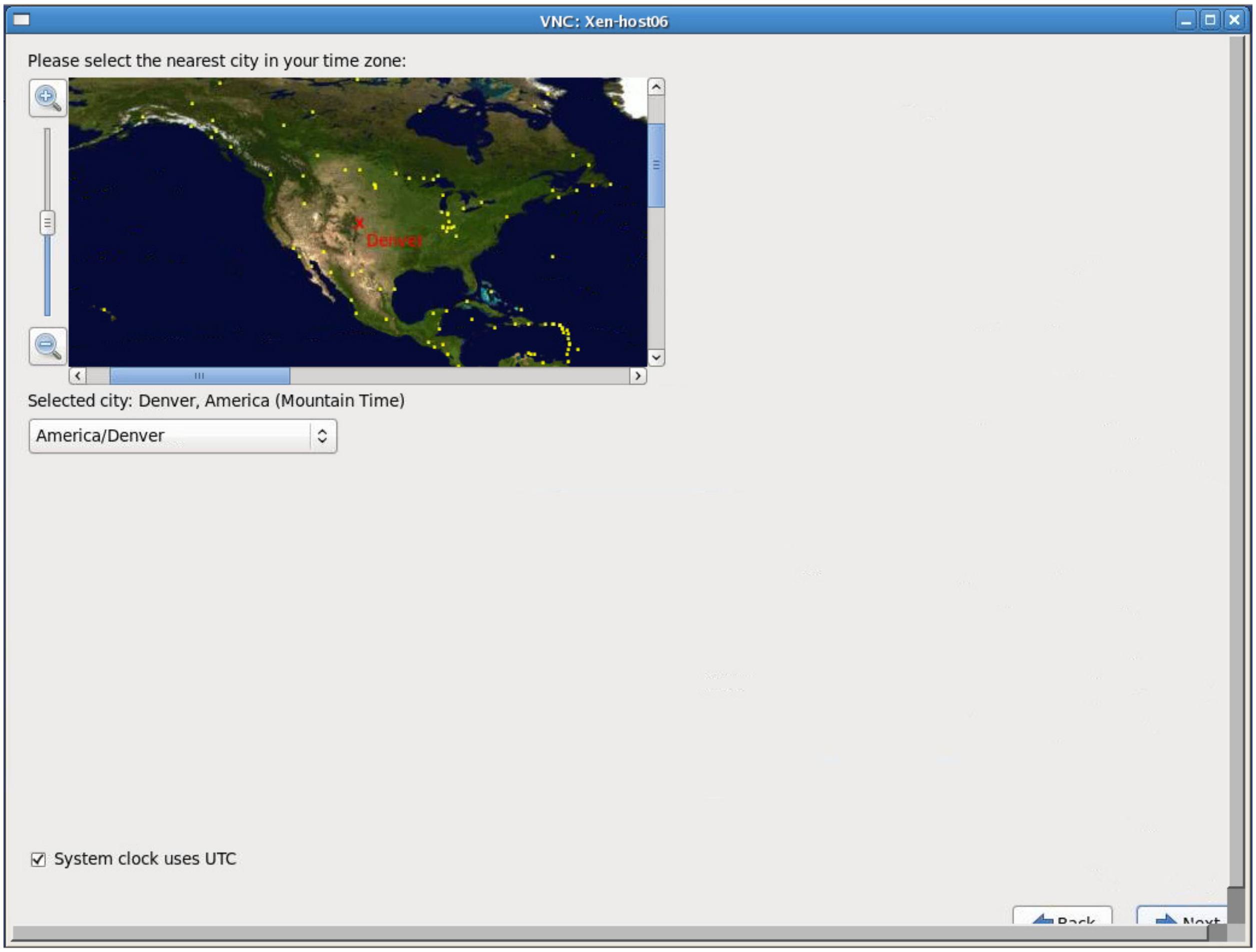
j. Enter `host06.example.com` as the Hostname. Scroll down if necessary and click **Next**.

- Do not click "Configure Network."



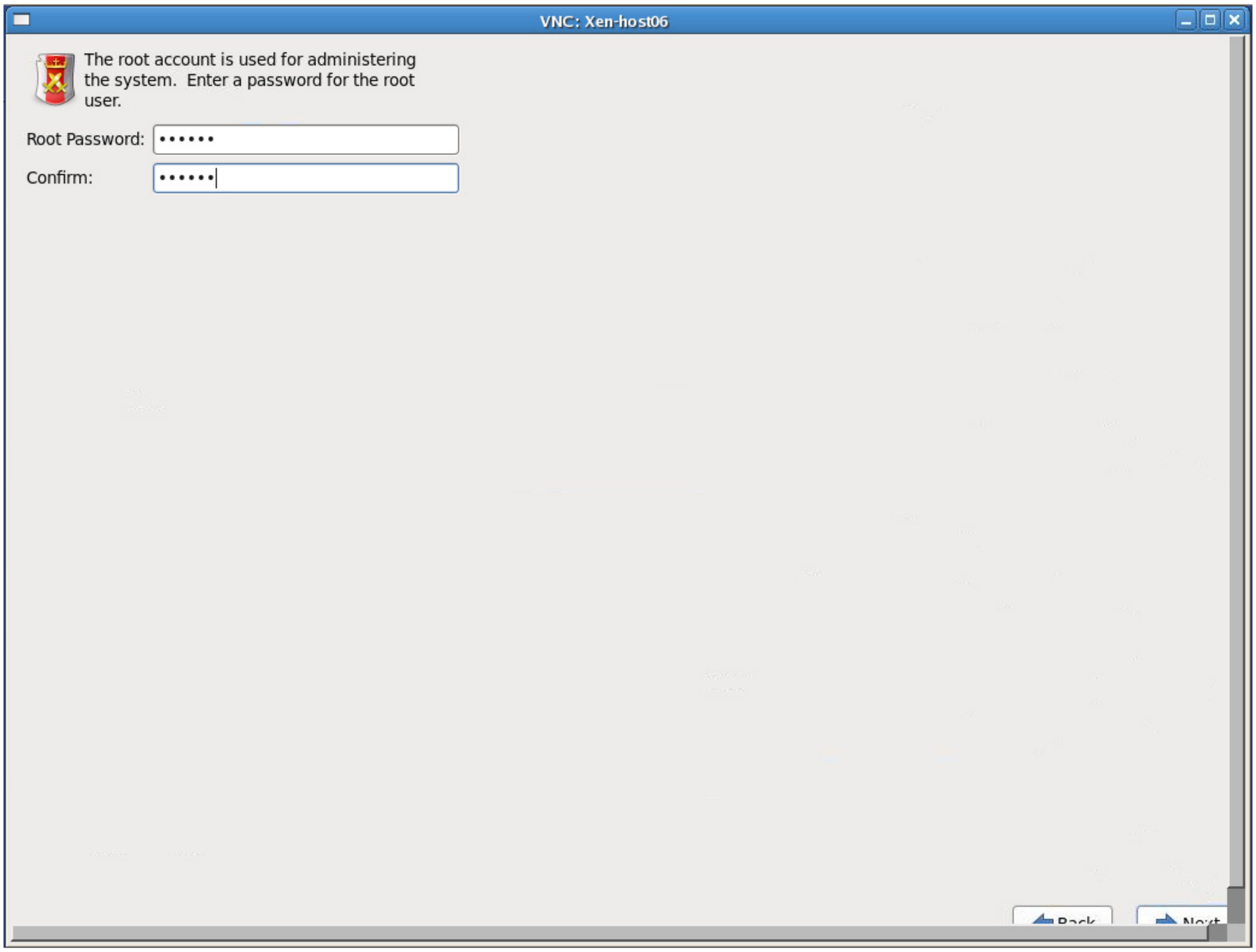
k. Select the nearest city in your time zone. Denver is selected in this example. Scroll down if necessary and click **Next**.

- Accept the default, “System clock uses UTC.”

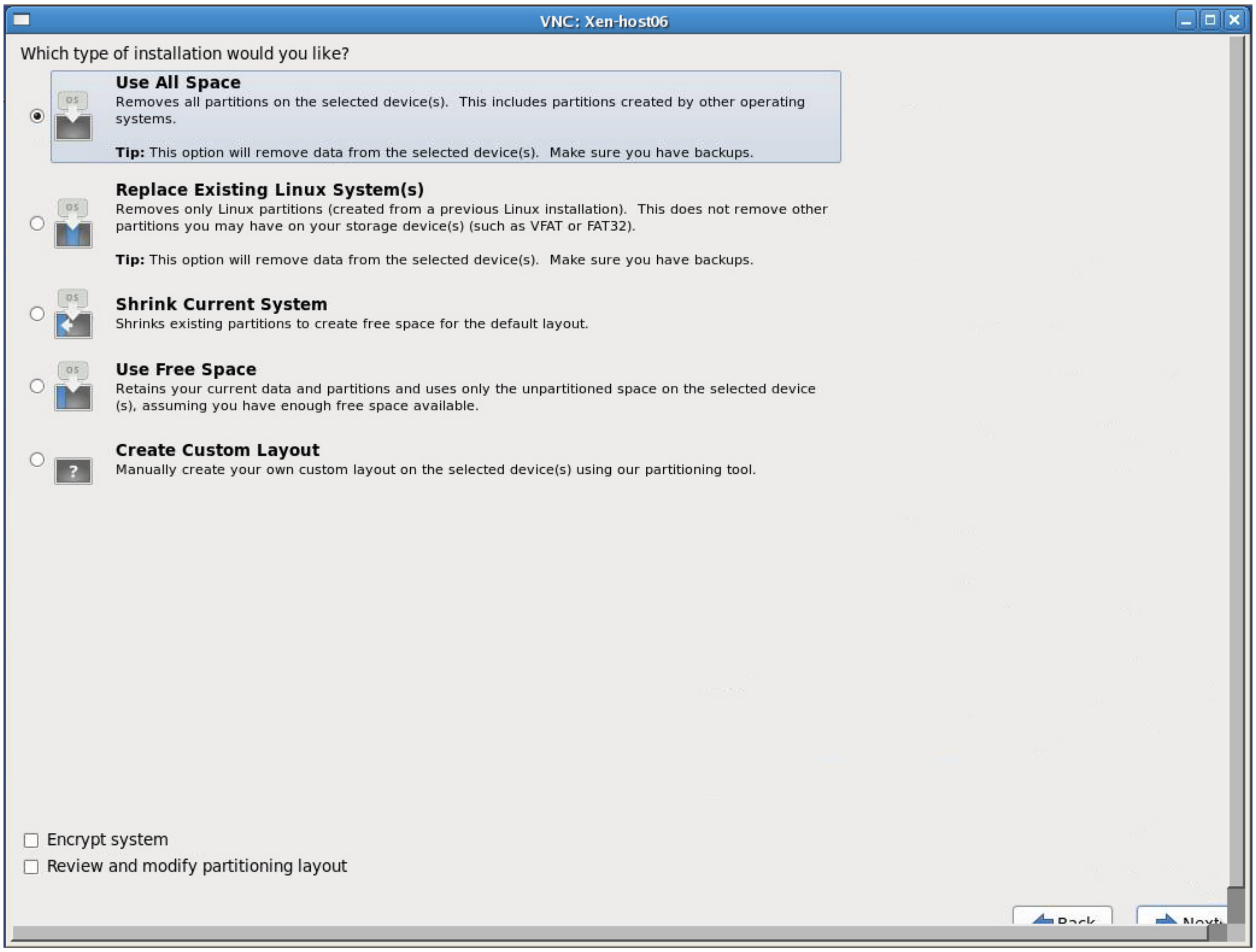


l. For the Root Password, enter `oracle`. Enter `oracle` to confirm. Scroll down if necessary and click **Next**.

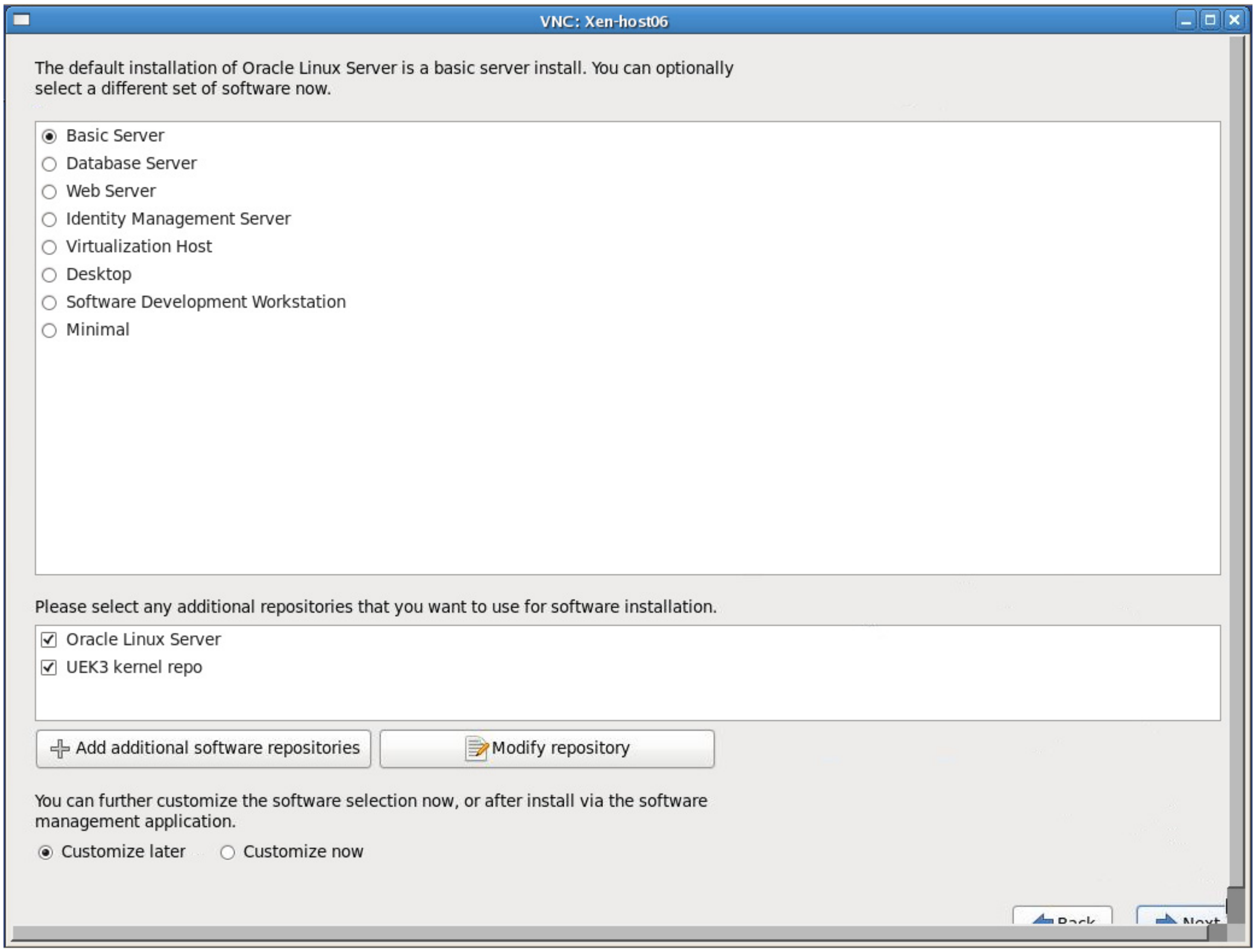
- Click "**Use Anyway**" when warned of a weak password.



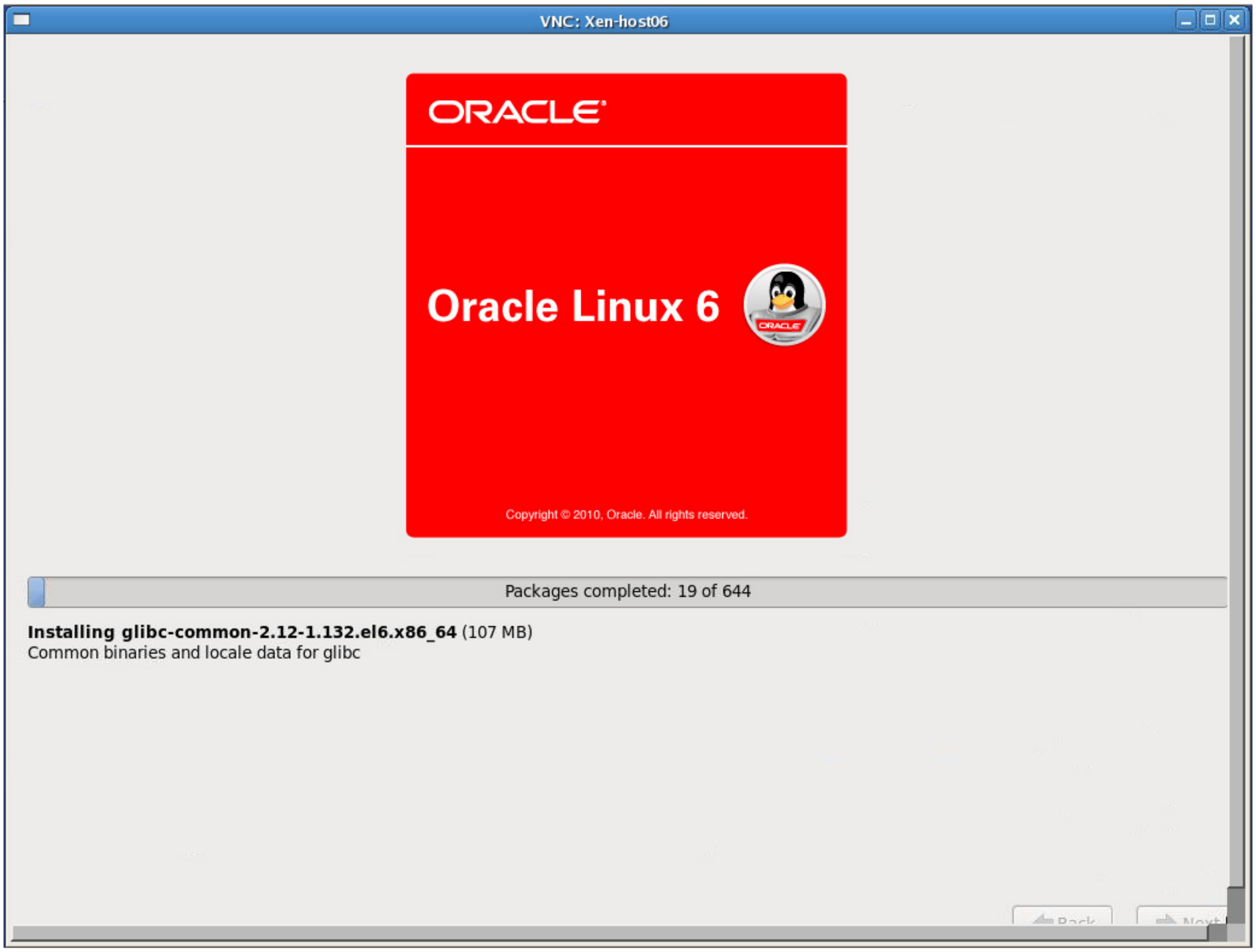
m. Select **Use All Space** for the type of installation. Scroll down if necessary and click **Next**.
- Do not click “Encrypt system” or “Review and modify portioning layout.”
- Click “**Write changes to disk**.”



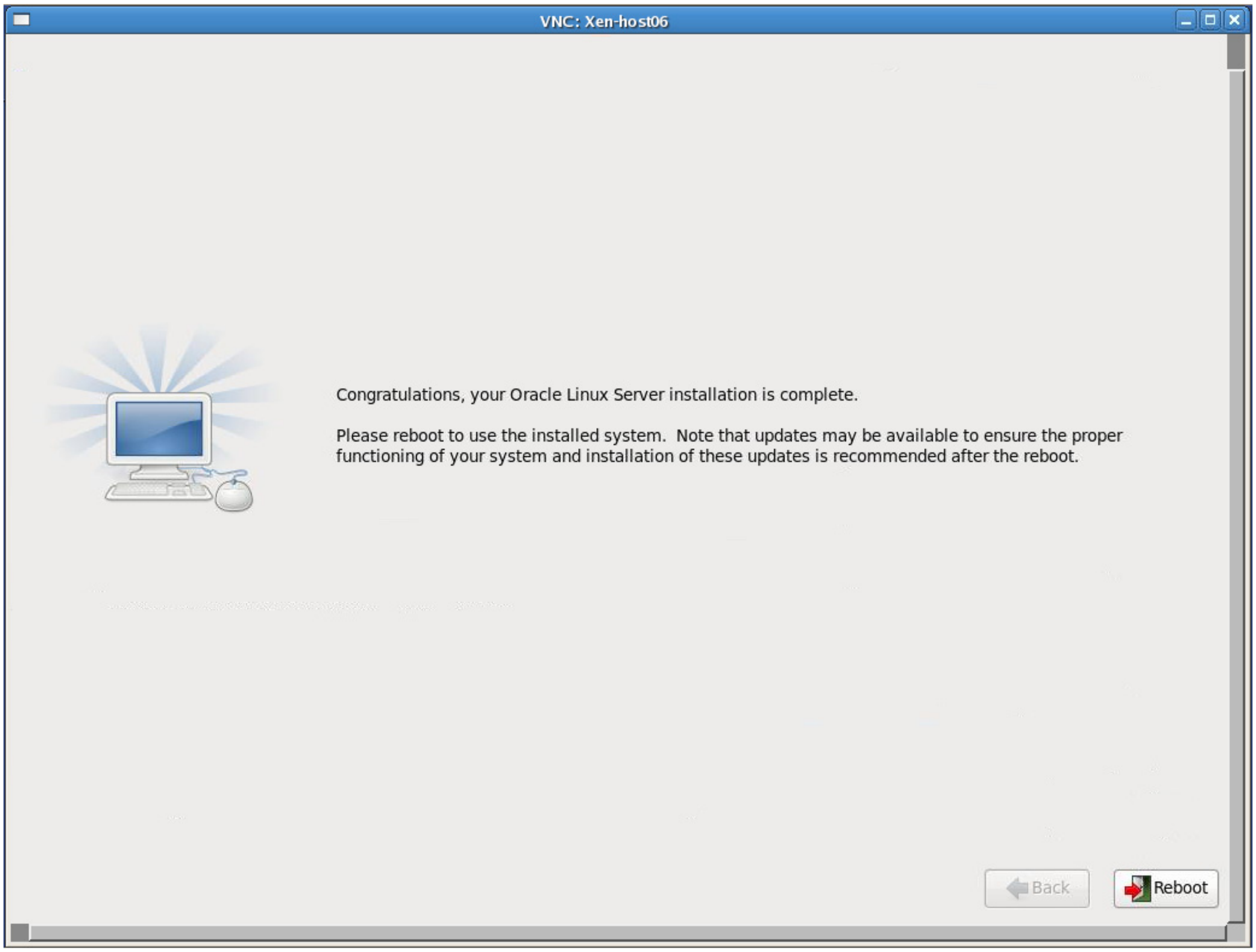
n. Select **Basic Server**. Scroll down if necessary and click **Next**.



o. The RPM packages are installed. No further action is required until all packages are installed.



p. Click **Reboot** when package installation has completed.



7. Log in to **host06** and explore the Btrfs root file system.
   - Wait a few seconds for **host06** to reboot before proceeding.

   a. From **dom0**, use the `ssh` command to log in to **host06** (`192.0.2.106`).
      - Because this is the first time you have logged in using `ssh`, the command checks to make sure that you are connecting to the host that you think you are connecting to. Enter **`yes`**.
      - The `root` password is `oracle`.

```
# ssh 192.0.2.106
The authenticity of host '192.0.2.106 (192.0.2.106)' can't be
established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.0.2.106' (RSA) to the list of
known hosts.
root@192.0.2.106's password: oracle
[root@host06 ~]# hostname
host06.example.com
```

      - The `hostname` command confirms whether you have successfully logged in to **host06**.

b. Use the `fdisk` command to view the storage devices.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes
/dev/xvda1   *      1      64     512000   83   Linux
/dev/xvda2         64    1567   12069888   8e   Linux LVM
Disk /dev/mapper/vg_host03-lv-root: 11.1 GB, 11068768256 bytes
Disk /dev/mapper/vg_host03-lv-swap: 1287 MB, 1287651328 bytes
```

- One 12 GB device is available, `/dev/xvda`.
- The `/dev/xvda` disk device has two partitions:
  - `/dev/xvda1` is the boot partition with a partition type `83 – Linux`.
  - `/dev/xvda2` is the remainder of the disk with a partition type of `8e – Linux LVM`.
- This system disk uses LVM volumes for the `root` and `swap` partitions.

c. Use the `df` command to list the mounted partitions.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/mapper/vg_host06-lv_root
               11G  2.1G   7.5G   22%  /
tmpfs         750M     0   750M    0%  /dev/shm
/dev/xvda1    477M   55M   398M   12%  /boot
```

- The root partition is mounted on the `vg_host06-lv-root` LVM volume.
- The boot partition is mounted on `/dev/xvda1`.

d. Use the `mount` command to display the mounted file systems.

```
# mount
/dev/mapper/vg_host06-lv_root on / type btrfs (rw)
...
```

- Notice that `/dev/mapper/vg_host06-lv-root` has a file system type of `btrfs`.

e. Use the `cat` command to display the contents of the `/etc/fstab` file.

```
# cat /etc/fstab
...
/dev/mapper/vg_host06-lv_root   /  btrfs  defaults  1 1
...
```

- Again, notice that the file system mounted on `/` is a Btrfs file system.

f. Use the `btrfs sub list` command to list the subvolumes and snapshots.

```
# btrfs sub list /
ID 260 gen 27 top level 5 path install
```

- The mounted root file system is a snapshot (named `install`) of the root file system taken at the end of the installation.
- In this example, the installation root subvolume has an ID of `5`.
- The subvolume with an ID of `260` (`install`) is currently mounted as `/`.

g. Use the `ls` command to display the contents of `/`.

```
# ls /
bin  cgroup  etc  lib   media  mnt   opt  root  selinux  sys
usr  boot    dev  home  lib64  misc  net  proc  sbin     srv
tmp  var
```

h. Use the `mkdir` command and the `mount` command to mount the installation root volume (`ID = 5`).

```
# mkdir /instroot
# mount -o subvolid=5 /dev/mapper/vg_host06-lv_root /instroot
```

i. Use the `ls` command to display the contents of `/instroot`.

```
# ls /instroot
bin  cgroup  etc  lib   media  mnt   opt  root  selinux  sys
usr  boot    dev  home  lib64  misc  net  proc  sbin     srv
tmp  var     install
```

- Notice the existence of the `install` directory, which was not listed before mounting the file system on `/instroot`.

8. Create a snapshot of the root file system.

a. Use the `cd` command to change to the `/instroot` directory.

```
# cd /instroot
```

b. Use the `btrfs sub` command to create a snapshot (named `root_shapshot`) of the root file system (`install`).

```
# btrfs sub snapshot install root_snapshot
Create a snapshot of ‘install’ in ‘./root_snapshot’
```

c. Use the `btrfs sub list` command to list the subvolumes and snapshots.

```
# btrfs sub list /
ID 260 gen 31 top level 5 path install
ID 261 gen 31 top level 5 path root_snapshot
```

- In this example, the `root_snapshot` snapshot has an ID of `261`.

9. Mount an alternate snapshot as the root file system.
   - To roll back changes to your system, you can mount a snapshot as the root file system by specifying the snapshot's ID as the default subvolume.

   a. Use the `btrfs sub` command to specify the `root_snapshot` snapshot (`ID = 261`) as the default subvolume for the root file system.

   ```
   # btrfs sub set-default 261 /
   ```

   - You would need to reboot the system to mount this snapshot as the root file system.

10. Shut down the **host06** VM and start **host03** VM.

    a. Use the `shutdown` command to shut down the **host06** VM.

    ```
    # shutdown -h now
    ...
    The system is going down for halt NOW!
    Connection to 192.0.2.106 closed by remote host.
    Connect to 192.0.2.106 closed.
    ```

    - The connection to **host06** closes and you are returned to **dom0**.

    b. Use the `xm list` command to list information about domains.

    ```
    # xm list
    Name ...
    Domain-0 ...
    host01 ...
    host02 ...
    ```

    - Notice that **host06** is not listed.
    - It might take a few seconds for **host06** to shut down.

    c. Use the `cd` command to change to the `/OVS/running_pool/host03` directory on **dom0**.

    ```
    # cd /OVS/running_pool/host03
    ```

    d. Use the `xm create vm.cfg` command to start the **host03** VM.

    ```
    # xm create vm.cfg
    Using config file "./vm.cfg".
    Started domain host03 (id=...)
    ```

Practices for Lesson 13: Control Groups (Cgroups)

Chapter 13

Practices for Lesson 13: Managing Resources with Control Groups (Cgroups)

Practices Overview

In these practices, you perform the following:

- Explore the default cgroup hierarchy.
- Create and delete child cgroups within the default cgroup hierarchy.
- Create a custom cgroup hierarchy.
- Use cgroups to allocate CPU use.

Practice 13-1: Exploring the Default Cgroup Hierarchy

Overview

In this practice, you:

- Set SELinux to Permissive mode
- Ensure that the cgroup package, `libcgroup`, is installed
- Explore the default `/etc/cgconfig.conf` file
- Start the `cgconfig` service and create the default hierarchy
- View the default hierarchy, the subsystems and parameters, and the lists of tasks (processes) associated with each hierarchy

Assumptions

- You are the `root` user on **dom0**.

Tasks

1. Use the `ssh` command to log in to **host03**.
   - The `root` password is `oracle`.

```
# ssh host03
root@host03's password: oracle
Last login: ...
[root@host03 ~]#
```

2. Use the `setenforce` command to set SELinux to Permissive mode.

```
# getenforce
Enforcing
# setenforce 0
# getenforce
Permissive
```

3. Use the `rpm` command to check whether the `libcgroup` package is installed.

```
# rpm -qa | grep libcgroup
libcgroup-0.40.rc1-5.el6_4.x86_64
```

   - In this example, the `libcgroup` package is installed.
   - If not installed, run the `yum install` command to install the package.

4. Use the `ls` command to display the contents of `/cgroup`.

```
# ls /cgroup
```

   - The `/cgroup` directory is created but contains no files.

5. Use the `cat` command to view the `/etc/cgconfig.conf` file.

```
# cat /etc/cgconfig.conf
...
mount {
     cpuset  = /cgroup/cpuset;
     cpu     = /cgroup/cpu;
```

```
    cpuacct = /cgroup/cpuacct;
    memory  = /cgroup/memory;
    devices = /cgroup/devices;
    freezer = /cgroup/freezer;
    net_cls = /cgroup/net_cls;
    blkio   = /cgroup/blkio;
}
```

- By default, all subsystems, or resource controllers, are mounted to `/cgroup/<subsystem>`.

6. Use the `service` command to start the `cgconfig` service.

```
# service cgconfig start
Starting cgconfig service:                    [  OK  ]
```

- Starting the `cgconfig` service reads the `/etc/cgconfig.conf` file and populates the `/cgroup` hierarchy.

7. Use the `ls -lR` command to display the contents of `/cgroup`. Only partial output is shown.

```
# ls -lR /cgroup
...
drwxr-xr-x ...  blkio
drwxr-xr-x ...  cpu
drwxr-xr-x ...  cpuacct
drwxr-xr-x ...  cpuset
drwxr-xr-x ...  devices
drwxr-xr-x ...  freezer
drwxr-xr-x ...  memory
drwxr-xr-x ...  net_cls

/cgroup/blkio
...
-r--r--r-- ... blkio.io_merged
...

/cgroup/cpu
...
-rw-r--r-- ... cpu.cfs_period_us
...

/cgroup/cpuacct
...
-r--r--r-- ... cpuacct.stat
...
```

```
/cgroup/cpuset
...
-rw-r--r-- ... cpuset.cpu_exclusive
...

/cgroup/devices
...
--w------- ... devices.allow
...

/cgroup/freezer
...

/cgroup/memory
...
-rw-r--r-- ... memory.failcnt
...

/cgroup/net_cls
...
-rw-r--r-- ... net_cls.classid
...
```

- Each subsystem directory contains parameters for controlling and reporting on the applicable resources or operations.
  - The `blkio` subsystem parameters control and report block I/O operations.
  - The `cpu` subsystem parameters control access to CPU resources.
  - The `cpuacct` subsystem parameters report usage of CPU resources.
  - The `cpuset` subsystem parameters control access to CPU cores and memory nodes (for systems with NUMA architectures).
  - The `devices` subsystem parameters control access to system devices.
  - The `freezer` subsystem parameters allow you to suspend and resume tasks.
  - The `memory` subsystem parameters control access to memory resources and report on memory use.
  - The `net_cls` subsystem parameters tag network packets for use by network traffic control.
- Each subsystem directory contains a `tasks` file, which contains all the process IDs (PIDs) assigned to the cgroup.

8. Use the `lssubsys -m` command to list subsystems and their respective mount points.

```
# lssubsys -m
cpuset  /cgroup/cpuset
cpu     /cgroup/cpu
cpuacct /cgroup/cpuacct
```

```
memory  /cgroup/memory
devices /cgroup/devices
freezer /cgroup/freezer
net_cls /cgroup/net_cls
blkio   /cgroup/blkio
```

- In this example, all subsystems are available.

9. Use the `lscgroup` command to list the cgroups on your system.

```
# lscgroup
cpuset:/
cpu:/
cpuacct:/
memory:/
devices:/
freezer:/
net_cls:/
blkio:/
```

  - In this example, each subsystem hierarchy has a root cgroup.

10. Run the following command to display the number of processes in the `tasks` file in each subsystem hierarchy. The `find` command is surrounded by backquotes.

- The sample output shows `220` tasks per cgroup. The number of tasks on your system might be different.

```
# wc -l `find /cgroup -name tasks`
220 /cgroup/freezer/tasks
220 /cgroup/cpuset/tasks
220 /cgroup/cpuacct/tasks
220 /cgroup/net_cls/tasks
220 /cgroup/memory/tasks
220 /cgroup/devices/tasks
220 /cgroup/blkio/tasks
220 /cgroup/cpu/tasks
...
```

- In this example, all processes (tasks) are associated with each cgroup hierarchy and associated subsystem parameters.

Practice 13-2: Creating Cgroups in the Default Hierarchy

Overview

In this practice, you:

- Create a child cgroup in one cgroup subsystem hierarchy
- Create a child cgroup in two cgroup subsystem hierarchies
- Delete the child cgroups

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Create a child group in the `/cgroup/cpu` hierarchy.
   a. Use the `cgcreate` command to create a child group named `cpu_group` in the `cpu` subsystem.

```
# cgcreate –g cpu:cpu_group
```

   - The `-g` option specifies the hierarchy in which the cgroup is created.
   - In this example, a cgroup named `cpu_group` is created in the `/cgroup/cpu` hierarchy.

   b. Use the `ls -l` command to list the contents of the `/cgroup/cpu` hierarchy.

```
# ls –l /cgroup/cpu
...
-rw-r--r--...  cgroup.clone_children
--w--w--w-...  cgroup.event_control
-rw-r--r--...  cgroup.procs
-rw-r--r--...  cpu.cfs_period_us
-rw-r--r--...  cpu.cfs_quota_us
drwxr-xr-x...  cgu_group
-rw-r--r--...  cpu.rt_period_us
-rw-r--r--...  cpu.rt_runtime_us
-rw-r--r--...  cpu.shares
-rw-r--r--...  cpu.stat
-rw-r--r--...  notify_on_release
-rw-r--r--...  release_agent
-rw-r--r--...  tasks
```

   - Notice that the new `cpu_group` directory exists in `/cgroup/cpu`.

   c. Use the `ls –R` command to list the contents of the `/cgroup/cpu` hierarchy.

```
# ls –R /cgroup/cpu
/cgroup/cpu:
cgroup.clone_children  cpu.cfs_quota_us   cpu.shares    tasks
cgroup.event_control   cgu_group          cpu.stat
cgroup.procs           cpu.rt_period_us   notify_on_release
```

```
cpu.cfs_period_us      cpu.rt_runtime_us  release_agent

/cgroup/cpu/cpu_group:
cgroup.clone_children  cpu.cfs_quota_us   cpu.stat
cgroup.event_control   cpu.rt_period_us   notify_on_release
cgroup.procs           cpu.rt_runtime_us  tasks
cpu.cfs_period_us      cpu.shares
```

- Notice that the new `cpu_group` child cgroup inherited the characteristics from its parent cgroup.

d. Run the following command to display the number of processes in the `tasks` file in each subsystem hierarchy.

```
# wc -l `find /cgroup -name tasks`
220 /cgroup/freezer/tasks
220 /cgroup/cpuset/tasks
220 /cgroup/cpuacct/tasks
220 /cgroup/net_cls/tasks
220 /cgroup/memory/tasks
220 /cgroup/devices/tasks
220 /cgroup/blkio/tasks
0 /cgroup/cpu/cpu_group/tasks
220 /cgroup/cpu/tasks
...
```

- Notice that the child group did not inherit the tasks.
- Tasks are not automatically assigned to a child group.

2. Create a single group in two different subsystem hierarchies.

a. Use the `cgcreate` command to create a child group named `cpu_mem` in both the `cpu` and `memory` subsystems.

```
# cgcreate -g cpu,memory:cpu_mem
```

- The `-g` option specifies the hierarchy in which the cgroup is created.
- In this example, a cgroup named `cpu_mem` is created in both the `/cgroup/cpu` and `/cgroup/memory` hierarchies.

b. Use the `lscgroup` command to list the cgroups on your system.

```
# lscgroup
cpuset:/
cpu:/
cpu:/cpu_mem
cpu:/cpu_group
cpuacct:/
memory:/
memory:/cpu_mem
devices:/
```

```
freezer:/
net_cls:/
blkio:/
```

- In this example, each subsystem hierarchy has a root cgroup.
- The `cpu` subsystem has two child cgroups: `cpu_group` and `cpu_mem`.
- The `memory` subsystem has one child cgroup: `cpu_mem`.

c. Use the `find` command to display `cpu_mem` entries in the `/cgroup` hierarchy.

```
# find /cgroup -name cpu_mem
/cgroup/memory/cpu_mem
/cgroup/cpu/cpu_mem
```

- The cgroup named `cpu_mem` exists in both the `/cgroup/memory` and `/cgroup/cpu` hierarchies.

3. Delete the child cgroups.

a. Use the `cgdelete` command to delete the `cpu_group` group from the `cpu` hierarchy.

```
# cgdelete cpu:cpu_group
```

b. Use the `lscgroup` command to list the cgroups on your system.

```
# lscgroup
cpuset:/
cpu:/
cpu:/cpu_mem
cpuacct:/
memory:/
memory:/cpu_mem
devices:/
freezer:/
net_cls:/
blkio:/
```

- Notice that the `cpu_group` group no longer exists.

c. Use the `cgdelete` command to delete the `cpu_mem` groups from the `cpu` and `memory` hierarchies.

```
# cgdelete cpu,memory:cpu_mem
```

d. Use the `lscgroup` command to list the cgroups on your system.

```
# lscgroup
cpuset:/
cpu:/
cpuacct:/
memory:/
devices:/
freezer:/
net_cls:/
blkio:/
```

- Notice that there are no child groups.
- Each subsystem hierarchy has only a root cgroup.

Practice 13-3: Creating a Custom Cgroup Hierarchy

Overview

In this practice, you:

- Modify the `/etc/cgconfig.conf` file to create a custom hierarchy
- Delete the hierarchy and re-create it from the command line

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Make a backup copy of the `/etc/cgconfig.conf` file.
   a. Use the `cp` command to copy `/etc/cgconfig.conf` to `/etc/cgconfig.conf.org`.

```
# cp /etc/cgconfig.conf /etc/cgconfig.conf.org
```

2. Use the `vi` editor to edit the "mount" section of `/etc/cgconfig.conf` as follows:
   - Delete the `freezer =` line and the `net_cls =` line.

```
# vi /etc/cgconfig.conf
...
mount {
     cpuset  = /cgroup/cpu-ram;
     cpu     = /cgroup/cpu-ram;
     cpuacct = /cgroup/cpu-ram;
     memory  = /cgroup/cpu-ram;
     devices = /cgroup/devlist;
     blkio   = /cgroup/iolimit;
}
```

   - In this example, four subsystems, `cpuset`, `cpu`, `cpuacct`, and `memory` are mounted to a single hierarchy, `/cgroup/cpu-ram`.
   - The `devices` subsystem is mounted on `/cgroup/devlist`.
   - The `blkio` subsystem is mounted on `/cgroup/iolimit`.
3. Use the `service` command to restart the `cgconfig` service.

```
# service cgconfig restart
Stopping cgconfig service:                [  OK  ]
Starting cgconfig service:                [  OK  ]
```

4. Use the `lscgroup` command to list the cgroups on your system.

```
# lscgroup
cpuset,cpu,cpuacct,memory:/
devices:/
blkio:/
```

   - This example shows the three cgroups, one of which comprises four subsystems, but the command does not show the mount points.

5. Use the `lssubsys -m` command to list subsystems and their respective mount points.

```
# lssubsys -m
cpuset,cpu,cpuacct,memory  /cgroup/cpu-ram
devices /cgroup/devlist
blkio   /cgroup/iolimit
```

- This output confirms the cgroup hierarchies and mount points as configured in `/etc/cgconfig.conf`.

6. Use the `ls` command to display the contents of `/cgroup/cpu-ram`.

```
# ls /cgroup/cpu-ram
...
cpuacct.stat
...
cpu.cfs_period_us
...
cpuset.cpu_exclusive
...
memory.failcnt
...
```

- Notice that parameters from four different subsystems—`cpu`, `cpuset`, `cpuacct`, and `memory`—are all available within the same hierarchy, `/cgroup/cpu-ram`.

7. Create the same hierarchy from the command line.

   a. Use the `service` command to stop the `cgconfig` service.

```
# service cgconfig stop
Stopping cgconfig service:                [  OK  ]
```

   b. Use the `ls -R` command to display the contents of `/cgroup`.

```
# ls -R /cgroup
/cgroup/:
blkio  cpuacct  cpuset   devlist  iolimit  net_cls
cpu    cpu-ram  devices  freezer  memory

/cgroup/blkio

/cgroup/cpu

/cgroup/cpuacct

/cgroup/cpu-ram

/cgroup/cpuset

/cgroup/devices
```

```
/cgroup/devlist

/cgroup/freezer

/cgroup/iolimit

/cgroup/memory

/cgroup/net_cls
```

- In this example, the mount points exist but no subsystems are mounted and no subsystem parameters are available.

c. Use the `/bin/rm -r /cgroup` command to remove all the mount points.

```
# /bin/rm -r /cgroup
```

d. Use the `mkdir` command to make the `/cgroup/cpu-ram` directory.

```
# mkdir -p /cgroup/cpu-ram
```

e. Use the `mount` command to mount the four subsystems—`cpu`, `cpuset`, `cpuacct`, and `memory`—onto the `/cgroup/cpu-ram` mount point.

```
# mount -t cgroup -o cpu,cpuset,cpuacct,memory cpu-ram
/cgroup/cpu-ram
```

f. Use the `ls` command to display the contents of `/cgroup/cpu-ram`. (Only a partial output is shown.)

```
# ls /cgroup/cpu-ram
...
cpuacct.stat
...
cpu.cfs_period_us
...
cpuset.cpu_exclusive
...
memory.failcnt
...
```

- Notice that parameters from four different subsystems—`cpu`, `cpuset`, `cpuacct`, and `memory`—are all available in the same hierarchy, `/cgroup/cpu-ram`.
- This is identical to the hierarchy that you created by editing `/etc/cgconfig.conf` and starting the `cgconfig` service.

8. Unassign all cgroup resources.

a. Use the `mount` command to display the cgroup mounts.

```
# mount | grep cgroup
cpu-ram on /cgroup/cpu-ram type cgroup (rw,cpu,cpuset,...)
```

b. Use the `umount` command to unmount `/cgroup/cpu-ram`.

```
# umount /cgroup/cpu-ram
```

c. Use the `mount` command to display any cgroup mounts.

```
# mount | grep cgroup
```

- The cgroups are no longer mounted.

d. Use the `ls -R` command to display the contents of `/cgroup`.

```
# ls -R /cgroup
/cgroup/:
cpu-ram

/cgroup/cpu-ram
```

- The mount point exists but no subsystems are mounted and no subsystem parameters are available.

Practice 13-4: Using Cgroups to Allocate CPU Resources

Overview

In this practice, you:

- Create four new users
- Create two new groups, `eng` and `sales`
- Assign the new users to the new groups
- Configure cgroups to allocate a 3:1 relative share of the available CPU time to members of the `eng` group, with members of the `sales` group getting the lesser share
- Configure the cgroup rules definition file
- Execute CPU-intensive commands as members of the `eng` and `sales` groups
- View CPU usage using the `top` utility

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Add users and groups.
   a. Use the `useradd` command to add users `jim`, `john`, `joan`, and `susan`.

```
# useradd jim
# useradd john
# useradd joan
# useradd susan
```

   b. Use the `passwd` command to assign a password of "`password`" to users `jim`, `john`, `joan`, and `susan`.
      - Ignore the BAD PASSWORD warning message.

```
# passwd jim
Changing password for user jim.
New password: password
BAD PASSWORD: it is based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
# passwd john
Changing password for user john.
New password: password
BAD PASSWORD: it is based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
# passwd joan
Changing password for user joan.
New password: password
BAD PASSWORD: it is based on a dictionary word
```

```
Retype new password: password
passwd: all authentication tokens updated successfully.
# passwd susan
Changing password for user susan.
New password: password
BAD PASSWORD: it is based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
```

c. Use the `groupadd` command to add groups `eng` and `sales`.

```
# groupadd eng
# groupadd sales
```

d. Use the `tail /etc/group` command to determine the GIDs for the `eng` and `sales` groups.

```
# tail /etc/group
...
eng:x:565:
sales:x:566:
```

- In this example, the GID for `eng` is `565` and the `GID` for `sales` is `566`.

e. Use the `usermod` command to add users `jim` and `joan` to the `eng` group. Add users `john` and `susan` to the `sales` group.

```
# usermod -aG 565 jim
# usermod -aG 565 joan
# usermod -aG 566 john
# usermod -aG 566 susan
```

2. Use the `tail` command to view the `/etc/group` file.

```
# tail /etc/group
...
eng:x:565:jim,joan
sales:x:566:john,susan
```

- Users `jim` and `joan` are members of the `eng` group.
- Users `john` and `susan` are members of the `sales` group.

3. Use the `vi` editor to modify the `/etc/cgconfig.conf` file as follows:

- In the "`mount`" section, delete the `cpuset =` line, the `memory =` line, the `devices =` line, and the `blkio =` line.
- Create two "`group`" sections to define the `eng` and `sales` child groups as shown.

Note: A pre-configured `cgconfig.conf` file exists on **dom0** in the `/OVS/seed_pool/sfws` directory.

- You can edit the `cgconfig.conf` file as follows using the `vi` command, or you can use the `sftp` command and copy `/OVS/seed_pool/sfws/cgconfig.conf` from **dom0** to `/etc/cgconfig.conf` on **host03**.

```
# vi /etc/cgconfig.conf
...
mount {
     cpu     = /cgroup/cpu-ram;
     cpuacct = /cgroup/cpu-ram;
}
group eng {
     cpu {
          cpu.shares="750";
     }
     cpuacct {
          cpuacct.usage="0";
     }
}
group sales {
     cpu {
          cpu.shares="250";
     }
     cpuacct {
          cpuacct.usage="0";
     }
}
```

- The "mount" section mounts both the cpu and cpuacct subsystems on the single /cgroup/cpu-ram hierarchy.
- The "group" section creates two child groups in this hierarchy, eng and sales.
- Processes started in the eng group get three times the relative share of CPU time:
  - cpu.shares is set to 750 for the eng group.
  - cpu.shares is set to 250 for the sales group.
- The cpuacct.usage parameters report the total CPU time in nanoseconds for all tasks in the cgroup.
  - Setting this parameter to 0 resets its value and also the value of the cpuacct.usage_percpu parameter.

4. Use the vi editor to modify the /etc/cgrules.conf file as follows:

```
# vi /etc/cgrules.conf
...
@sales      cpu,cpuacct       sales
@eng        cpu,cpuacct       eng
```

- The first rule assigns the sales cgroup and the cpu and cpuacct subsystems to all processes originating from the sales group.
- The second rule assigns the eng cgroup and the cpu and cpuacct subsystems to all processes originating from the eng group.

5. Use the `service` command to restart the `cgconfig` service.

```
# service cgconfig restart
Stopping cgconfig service:                    [  OK  ]
Starting cgconfig service:                    [  OK  ]
```

- Restarting the `cgconfig` service reads the `/etc/cgconfig.conf` file and updates the `/cgroup` hierarchy.

6. Use the `lscgroup` command to list the cgroups on your system.

```
# lscgroup
cpu,cpuacct:/
cpu,cpuacct:/sales
cpu,cpuacct:/eng
```

- This example shows the root cgroup and the two child groups, `sales` and `eng`.

7. Use the `lssubsys -m` command to list the available subsystems and their respective mount points.

```
# lssubsys -m
cpu,cpuacct  /cgroup/cpu-ram
```

- This output confirms that only the `cpu` and `cpuacct` subsystems are available and that they are mounted on a single hierarchy, `/cgroup/cpu-ram`.

8. Use the `ls –l` command to display the contents of `/cgroup/cpu-ram`. Only partial output is shown.

```
# ls –R /cgroup/
/cgroup/:
cpu-ram

/cgroup/cpu-ram:
...

/cgroup/cpu-ram/eng:
...

/cgroup/cpu-ram/sales:
...
```

- Notice that all parameters from both the `cpu` and `cpuacct` subsystems are available in the parent cgroup and the children cgroups.

9. Use the `cgget` command to view the value of the `cpu.shares` parameter.

```
# cgget -r cpu.shares /
/:
cpu.shares: 1024
# cgget -r cpu.shares /eng
/eng:
cpu.shares: 750
# cgget -r cpu.shares /sales
/sales:
cpu.shares: 250
```

- Notice the default value of `1024` for the root cgroup.
- Notice that `cpu.shares` is set to `750` for the `eng` group, and `250` for the `sales` group, as configured in `/etc/cgconfig.conf`.

10. Use the `service` command to start the `cgred` service.

```
# service cgred start
Starting CGroup Rules Engine Daemon:     [  OK  ]
```

11. In a new terminal window, log in as user `john` on **host03**.
   - User `john` is a member of the `sales` group.
   - The `sales` group has been allocated only 25% of the CPU.

   a. From **dom0**, open a new terminal window and use the `ssh john@host03` command to log on to **host03** as user `john` (password is `password`).
      - Use the `hostname` command and the `whoami` command to verify that you have logged on to **host03** as user `john`.

```
[dom0]$ ssh john@host03
john@hostw03's password: password
[john@host03 ~]$ hostname
host03.example.com
[john@host03 ~]$ whoami
john
```

12. In a new terminal window, log in as user `joan` on **host03**.
   - User `joan` is a member of the `eng` group.
   - The `eng` group has been allocated 75% of the CPU.

   a. From **dom0**, open a new terminal window and use the `ssh joan@host03` command to log on to **host03** as user `joan` (password is `password`).
      - Use the `hostname` command and the `whoami` command to verify that you have logged on to **host03** as user `joan`.

```
[dom0]$ ssh joan@host03
joan@hostw03's password: password
[joan@host03 ~]$ hostname
host03.example.com
[joan@host03 ~]$ whoami
joan
```

13. Execute CPU-intensive commands as users `john` and `joan`. View processes using the `top` command.

    a. In the terminal window where you are logged in as the `root` user, run the `top` command.

    ```
    [root@host03]# top
    ```

    b. In the terminal window where you are logged in as user `john`, run the following CPU-intensive command:

    ```
    [john@host03]$ dd if=/dev/zero of=/dev/null bs=1024
    ```

    c. In the terminal window where you are logged in as user `joan`, run the following CPU-intensive command:

    ```
    [joan@host03]$ dd if=/dev/zero of=/dev/null bs=1024
    ```

14. View the output of the `top` command. The following is only a sample:

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
top - 18:27:04 up 1 day,  3:43,  3 users,  load average: 1.09, 0.32, 0.14
Tasks: 101 total,   3 running,  98 sleeping,   0 stopped,   0 zombie
Cpu(s): 17.9%us, 82.1%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1536988k total,   378576k used,  1158412k free,    45532k buffers
Swap:  3080188k total,        0k used,  3080188k free,   187892k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6242 joan      20   0  102m  512  428 R 75.1  0.0   0:42.83 dd
 6244 john      20   0  102m  508  428 R 24.9  0.0   0:09.66 dd
    1 root      20   0 19400 1560 1244 S  0.0  0.1   0:00.59 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.03 ksoftirqd/0
    5 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u:0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    7 root      RT   0     0    0    0 S  0.0  0.0   0:01.26 watchdog/0
    8 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 cpuset
    9 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 khelper
   10 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 netns
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 xenwatch
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.00 xenbus
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.55 sync_supers
   14 root      20   0     0    0    0 S  0.0  0.0   0:00.01 bdi-default
   15 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kintegrityd
   16 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kblockd
```

- In this example, the user `joan` is a member of the `eng` group, which has a `3:1` relative share over members of the `sales` group. The `dd` process for this user has 75.1% CPU use.
- User `john` is a member of the `sales` group. The `dd` process for this user has 24.9% CPU use.
- This shows that allocating a larger share of CPU time to users in the `eng` group using cgroups is successful.

15. Exit the `top` and `dd` processes.
    a. In the `root` user's terminal window in which the `top` process is running, press `q` to quit `top`.
    b. In the `john` user's terminal window, press Ctrl + C to abort the `dd` command.
    c. In the `joan` user's terminal window, press Ctrl + C to abort the `dd` command.
16. Stop the `cgred` service and restart the `cgconfig` service.
    a. In the `root` user's terminal window, use the `service` command to stop the `cgred` service.

```
# service cgred stop
Stopping CGroup Rules Engine Daemon:        [  OK ]
```

    b. In the `root` user's terminal window, use the `service` command to restart the `cgconfig` service.

```
# service cgconfig restart
Stopping cgconfig service                   [  OK ]
Starting cgconfig service                   [  OK ]
```

17. Repeat steps 13 and 14. The following is a sample output of the `top` command:

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
top - 20:00:49 up 1 day,  5:17,  3 users,  load average: 1.11, 1.44, 1.00
Tasks: 100 total,   3 running,  97 sleeping,   0 stopped,   0 zombie
Cpu(s): 19.0%us, 81.0%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1536988k total,   382032k used,  1154956k free,    46544k buffers
Swap:  3080188k total,        0k used,  3080188k free,   190604k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6601 joan      20   0  102m  508  428 R 50.2  0.0   0:09.01 dd
 6600 john      20   0  102m  508  428 R 49.9  0.0   0:11.24 dd
 6598 root      20   0 15076 1084  832 R  0.3  0.1   0:00.04 top
    1 root      20   0 19400 1560 1244 S  0.0  0.1   0:00.60 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.04 ksoftirqd/0
    5 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u:0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    7 root      RT   0     0    0    0 S  0.0  0.0   0:01.32 watchdog/0
    8 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 cpuset
    9 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 khelper
   10 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 netns
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 xenwatch
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.00 xenbus
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.57 sync_supers
   14 root      20   0     0    0    0 S  0.0  0.0   0:00.01 bdi-default
   15 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kintegrityd
```

- Notice that after stopping the `cgred` service, the `dd` processes for user `joan` and user `john` now have a relatively even amount of CPU use.

    a. Repeat step 15 to stop the `dd` and `top` processes.

18. Log out as users `joan` and `john` and close the respective terminal windows.

Practices for Lesson 14: Virtualization with Linux

Chapter 14

Practices for Lesson 14: Virtualization with Linux

Practices Overview

In these practices, you get familiar with the tools to create and manage virtual guests in a KVM environment.

When active, KVM turns a physical machine into a virtualization host.

In the practices for this lesson, you use a Linux host called **host05**, which is not a physical machine in your lab environment; **host05** is a virtual machine running on your lab PC. For this reason, KVM is available in **host05** but is not active. All the tasks in these practices work the same, whether KVM is active or not. The side effect of KVM not being active is that virtual guests deployed from **host05** will perform very slowly. This side effect has little impact on the exercises.

During the practices for this lesson, you use the following commands to manage your virtual guests: `virt-manager` and `virsh`. These two commands are part of the `libvirt` toolkit. With the `libvirt` toolkit, you can manage the virtualization capabilities offered by KVM on your virtualization host, **host05**.

Practice 14-1: Preparing the Virtualization Host for KVM

Overview

In this practice, you prepare your virtualization host, **host05**, to run a virtual guest. The virtualization host runs Oracle Linux 6.5.

Assumptions

This practice makes the following assumptions about **host05**, your virtualization host:

- The virtualization package groups are already installed:
  - Virtualization Client
  - Virtualization Platform
  - Virtualization Tools
- SElinux is enabled, and the firewall is enabled
- IP forwarding is enabled for NAT networking.

This practice also assumes that you are the `root` user on to **dom0**. You might be logged in to **host03** from Practice 13.

Tasks

1. If you are logged on to **host03**, use the `exit` command to log off.
   - Log off any other VM you might be logged on to.

```
# exit
logout
Connection to host03 closed.
```

2. Shut down all VMs.
   - You need to release the memory allocated to the running VMs before starting **host05**.

   a. From **dom0**, run the `xm list` command to display the running VMs.

```
# xm list
Name               ID    Mem    VCPUs    State    Time(s)
Domain-0            0   2048        2    r-----       ...
host01            242   1536        1    -b----       ...
host02            243   1536        1    -b----       ...
host03            244   1536        1    -b----       ...
```

   - In this example, **host01**, **host02**, and **host03**, in addition to **Domain-0**, are running.

   b. Use the `xm destroy <host#>` command to shut down each running VM.
   - This example shuts down all three VMs.
   - Run the `xm list` command again to verify all three VMs have been shut down.

```
# xm destroy host01
# xm destroy host02
# xm destroy host03
# xm list
Name               ID    Mem    VCPUs    State    Time(s)
Domain-0            0   2048        2    r-----       ...
```

- In this example, all three VMs have been shut down.

3. Start the **host05** virtual machine, which acts as your virtualization host for the practices in this lesson.
   a. Use the `cd` command to change to the `/OVS/running_pool/host05` directory.

```
# cd /OVS/running_pool/host05
```

   b. Run the `xm create` command to start the **host05** virtual machine.

```
# xm create vm.cfg
```

4. Log in to **host05**.
   a. Determine the VNC port number for **host05** by running the `xm list -l host05 | grep location` command.

```
# xm list -l host05 | grep location
            (location 0.0.0.0:5900)
            (location 3)
```

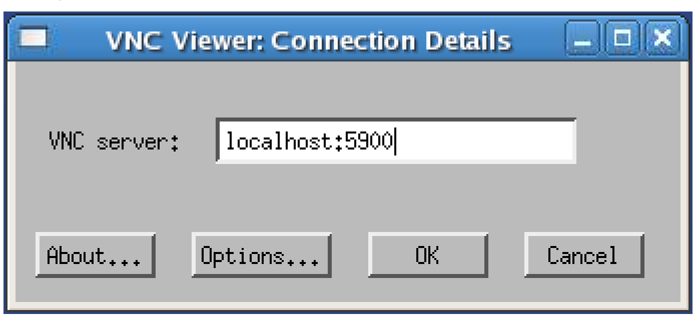
   - The sample shown indicates that the port number is **5900**. Your port number might be different.

   b. Run the `vncviewer&` command.

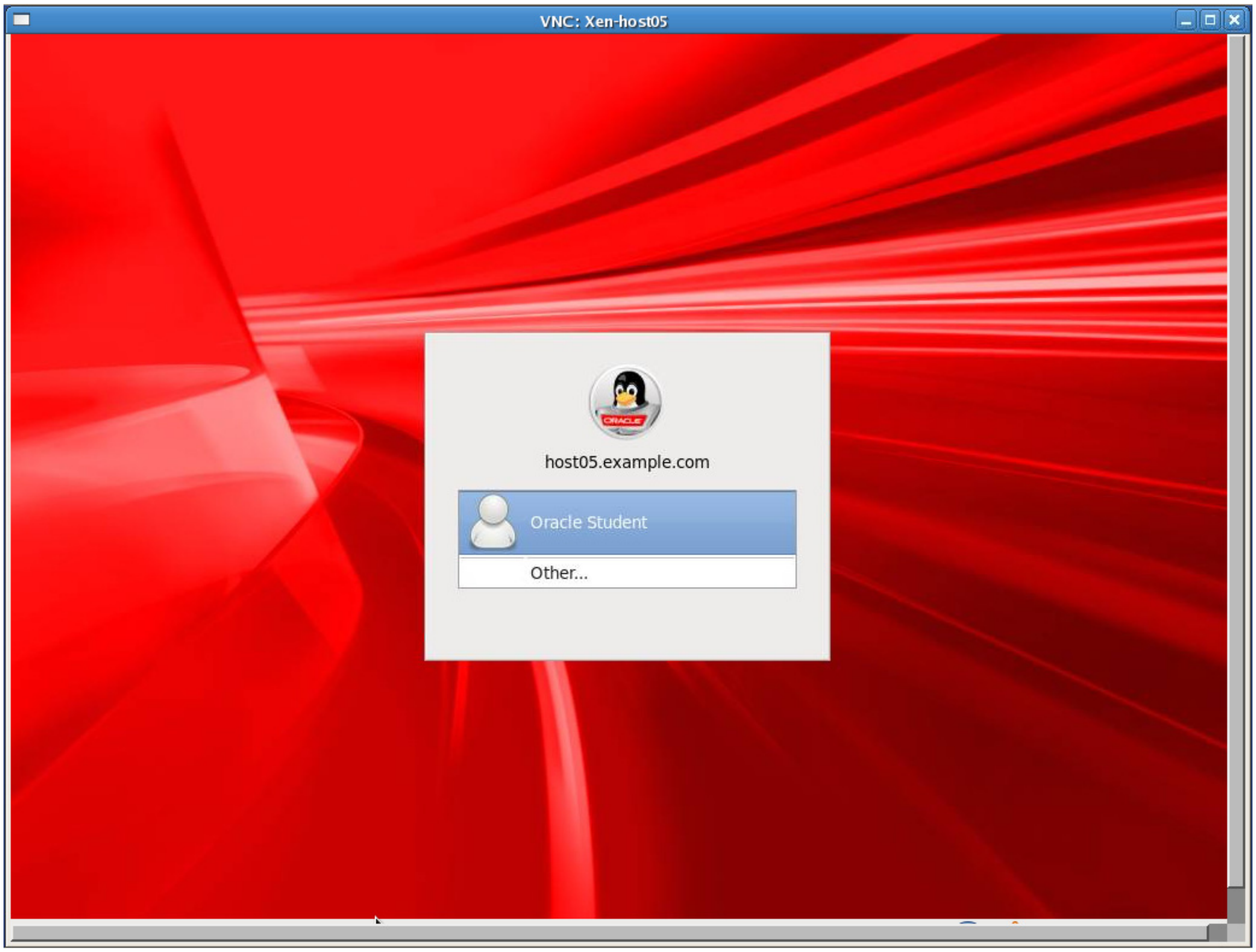
```
# vncviewer&
```

   - The VNC Viewer: Connection Details dialog box is displayed.

   c. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list -l host05 | grep location` command. For example, if the port number is **5900**, enter `localhost:5900` and click **OK**.



   - The GNOME login screen appears:

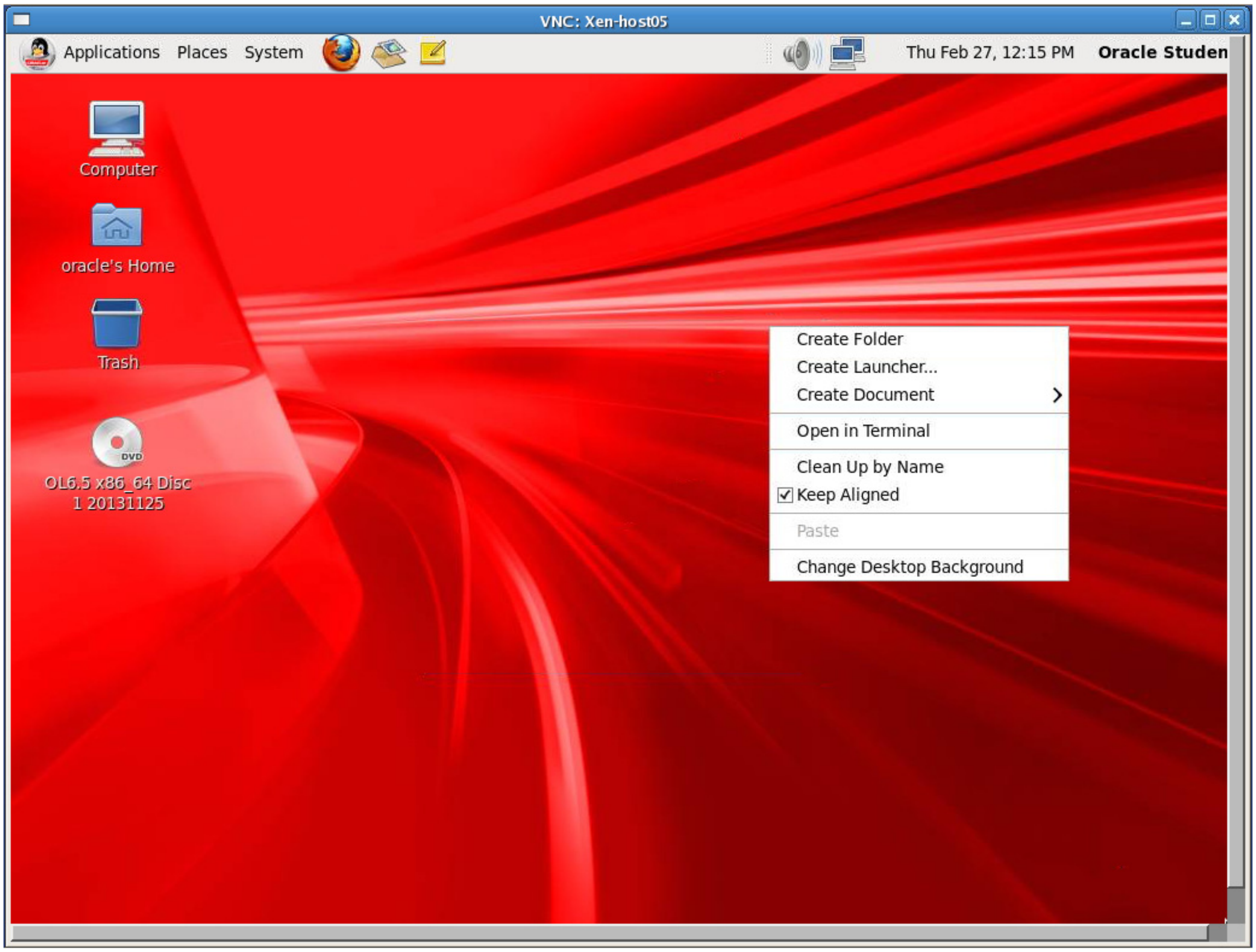


d. Click **Oracle Student** in the list of users. You are prompted for the **Password**.

e. Enter `oracle` for the Password and click **Log In**.

- The GNOME desktop appears.

f. Right-click on the desktop to display the pop-up menu:



g. From the pop-up menu, select **Open in Terminal**.
   - A terminal window appears.

h. In the terminal window, use the `su -` command to become the `root` user.
   - The `root` password is `oracle`.

```
$ su –
Password: oracle
#
```

5. List and examine the packages that are installed to support KVM operations.

a. Use the `rpm -ql` command to list the executable files in the `libvirt` package.
   - Pipe the output to `grep bin` to display only the commands and utilities.

```
# rpm -ql libvirt | grep bin
/usr/sbin/libvirtd
```

   - The `libvirt` package contains the `libvirtd` daemon, which communicates with the hypervisor on the virtualization host.

b. Use the `rpm -ql` command to list the executable files in the `libvirt-client` package.

- Pipe the output to `grep bin` to display only the commands and utilities.

```
# rpm -ql libvirt-client | grep bin
/usr/bin/virsh
/usr/bin/virt-host-validate
/usr/bin/virt-pki-validate
/usr/bin/virt-xml-validate
```

- The `libvirt-client` package contains the `virsh` command, an important command that you use to manage virtual machines and their resources.

c. Use the `rpm -ql` command to list the executable files in the `virt-manager` package.

- Pipe the output to `grep bin` to display only the commands and utilities.

```
# rpm -ql virt-manager | grep bin
/usr/bin/virt-manager
```

- The `virt-manager` package contains the `virt-manager` command that launches the Virtual Machine Manager graphical user interface.

d. Use the `rpm -ql` command to list the executable files in the `python-virtinst` package.

- Pipe the output to `grep bin` to display only the commands and utilities.

```
# rpm -ql python-virtinst | grep bin
/usr/bin/virt-clone
/usr/bin/virt-convert
/usr/bin/virt-image
/usr/bin/virt-install
/usr/sbin/virt-install
```

- The `python-virtinst` package contains the `virt-install` command that you can use to create virtual machines from the command line or from within scripts.

6. Examine the status of KVM.

a. Use the `lsmod` command to list the loaded KVM modules.

```
# lsmod | grep kvm
```

- In this example, the KVM modules are not loaded.

b. Use the `modprobe` command to load the KVM module.

- Repeat the lsmod command to verify that the KVM module is loaded.

```
# modprobe kvm
# lsmod | grep kvm
kvm                    436463  0
```

- The `kvm` module is now loaded.
- The `kvm` module is not enough to use KVM successfully. KVM needs the hardware acceleration provided by the processor on the physical host machine.

c. Examine the `/etc/sysconfig/modules/kvm.modules` file for information about the KVM hardware-specific module that is loaded at boot time:

```
# cat /etc/sysconfig/modules/kvm.modules
#!/bin/sh

if [ $(grep -c vmx /proc/cpuinfo) -ne 0 ]; then
    modprobe -b kvm-intel >/dev/null 2>&1
fi

if [ $(grep -c svm /proc/cpuinfo) -ne 0 ]; then
    modprobe -b kvm-amd >/dev/null 2>&1
fi

modprobe -b vhost_net >/dev/null 2>&1

exit 0
```

- At boot time, in addition to the `kvm` module, an additional `kvm` module associated with either the Intel (`kvm-intel`) or AMD (`kvm-amd`) processor is loaded to provide full KVM capabilities.

d. Attempt to load the `kvm` module associated with either Intel or AMD:

```
# modprobe kvm-intel
FATAL: Error inserting kvm_intel (/lib/modules/3.8.13-16.2.1.el6uek.x86_64/kernel/arch/x86/kvm/kvm-intel.ko): Operation not supported
# modprobe kvm-amd
FATAL: Error inserting kvm_amd (/lib/modules/3.8.13-16.2.1.el6uek.x86_64/kernel/arch/x86/kvm/kvm-amd.ko): Operation not supported
#
```

- In your lab environment, you cannot load the Intel or AMD KVM module in **host05** because **host05** is not running directly on the physical machine. As a result, you cannot use the KVM acceleration provided by the processor. This is a limitation of your lab environment, not a limitation of KVM.
- Without KVM acceleration, **host05** uses the QEMU virtual machine emulator exclusively to run virtual guests. All the practices in this lesson work the same, whether the acceleration is present or not. Without the KVM acceleration, the performance of the virtual machine that you deploy in Practice 14-3 is going to be poor but this situation does not affect the outcome of the practice.
- At your own site, with proper support by your Intel or AMD-based processor, you can take full advantage of the KVM acceleration provided by the hardware extensions in your physical machine.

7. Ensure that the `libvirtd` daemon is running and starts at boot time.

a. Use the `service` command to verify that the `libvirtd` service is running

```
# service libvirtd status
libvirtd (pid ...) is running...
```

Practice 14-2: Starting the Virtual Machine Manager and Preparing to Create a Virtual Machine

Overview

In this practice, you start the `libvirt` graphical user interface by using the `virt-manager` command. The `libvirt` graphical user interface is called the Virtual Machine Manager.

With the Virtual Machine Manager, you manipulate two types of entities:

- Connections
- Virtual machines deployed using each connection.

A connection is used to connect to a specific hypervisor:

- The connection can be local (the hypervisor is located on the host where the Virtual Machine Manager runs) or remote.
- Supported hypervisors include KVM-QEMU, Linux Containers (LXC), and Xen.

In all the practices for this lesson, you use a connection to a KVM-QEMU type hypervisor.

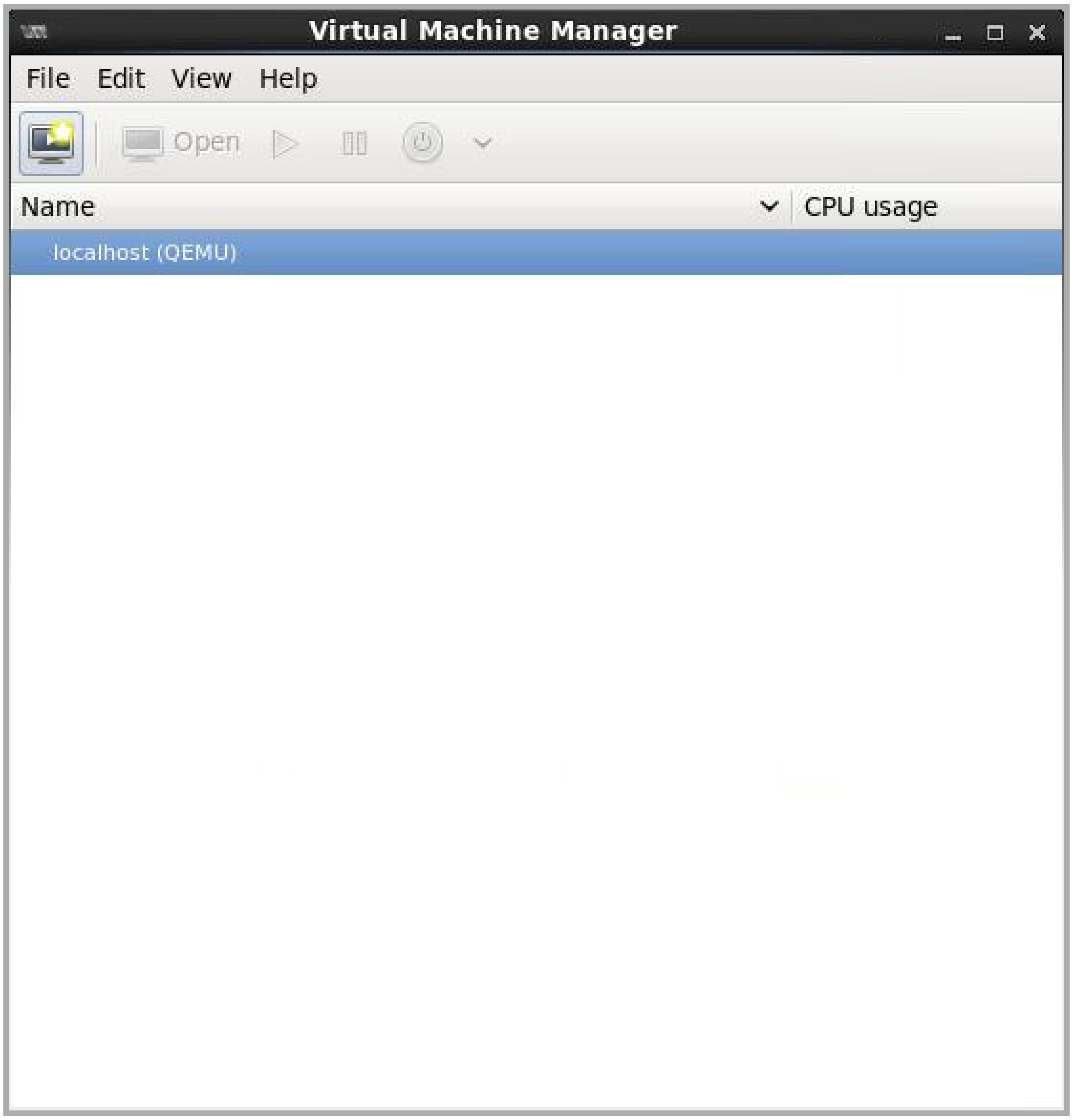
Before attempting to create virtual machines, ensure that the networking and storage resources are in place. You perform this verification from the Virtual Machine Manager.

Tasks

1. Start the Virtual Machine Manager.
   a. From **host05**, run the `virt-manager` command to launch the Virtual Machine Manager.

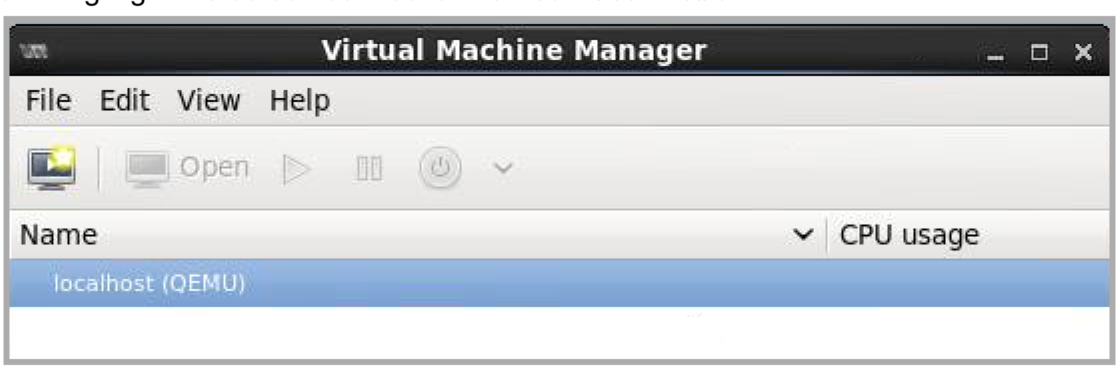
```
# virt-manager&
```

- The Virtual Machine Manager main window appears.

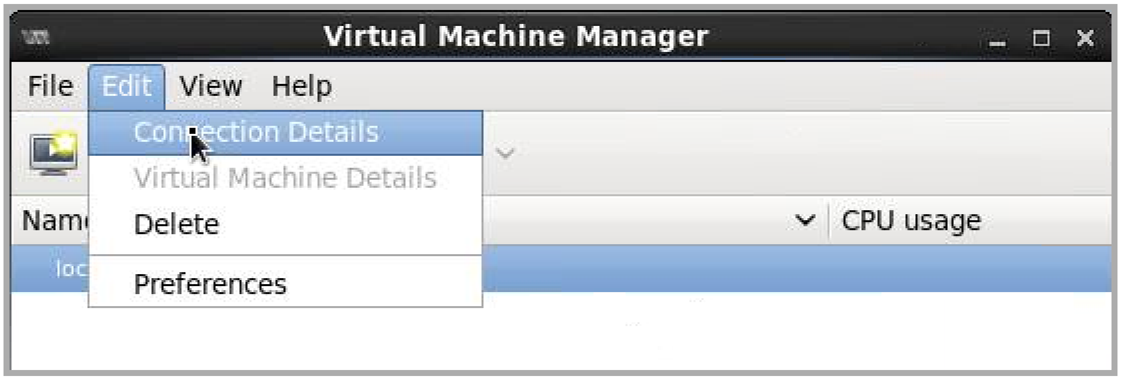


- If you get an error stating that the `libvirt` package is not installed, run the `virt-manager` command again.

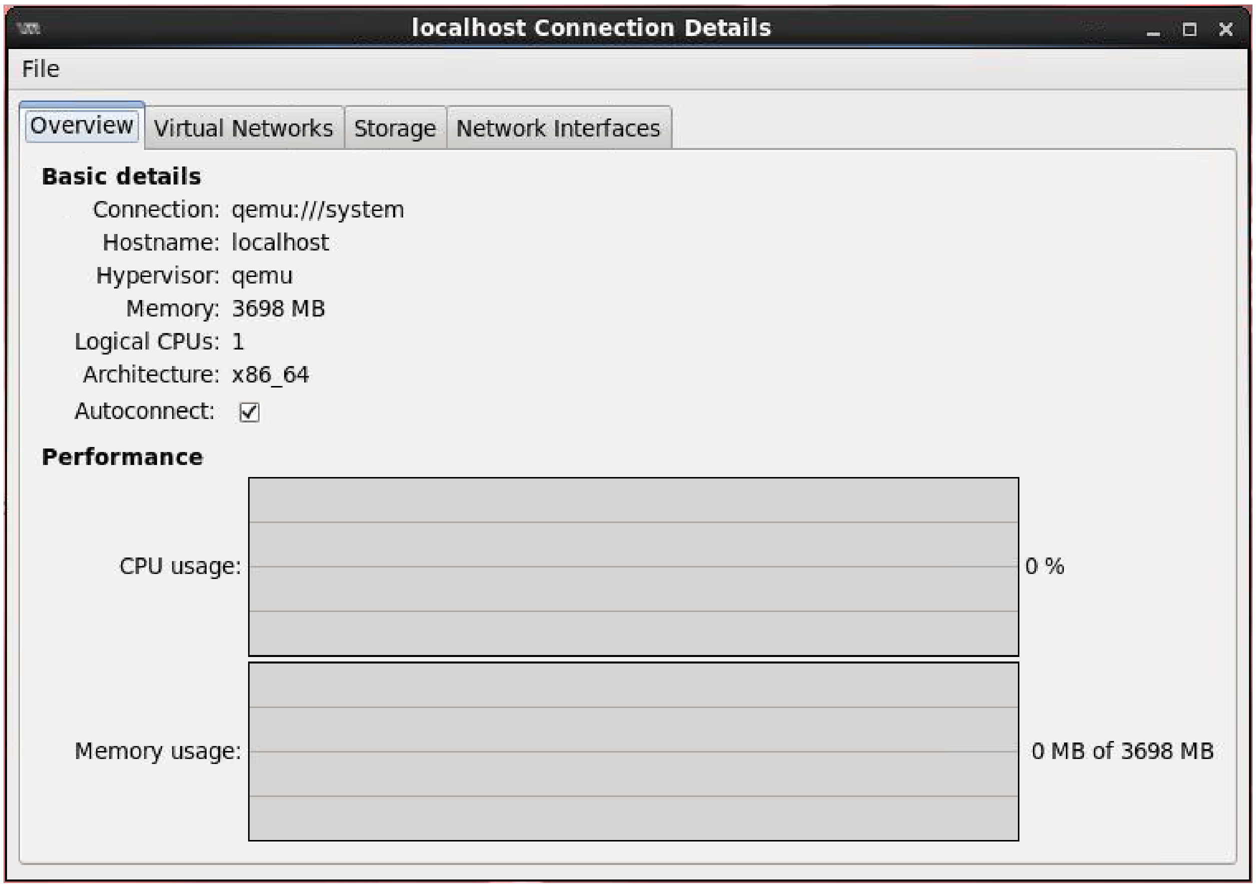
2. Examine the connection and connection details.
   a. Highlight the default connection named `localhost`.



b. Select **Edit > Connection Details** from the menu bar.

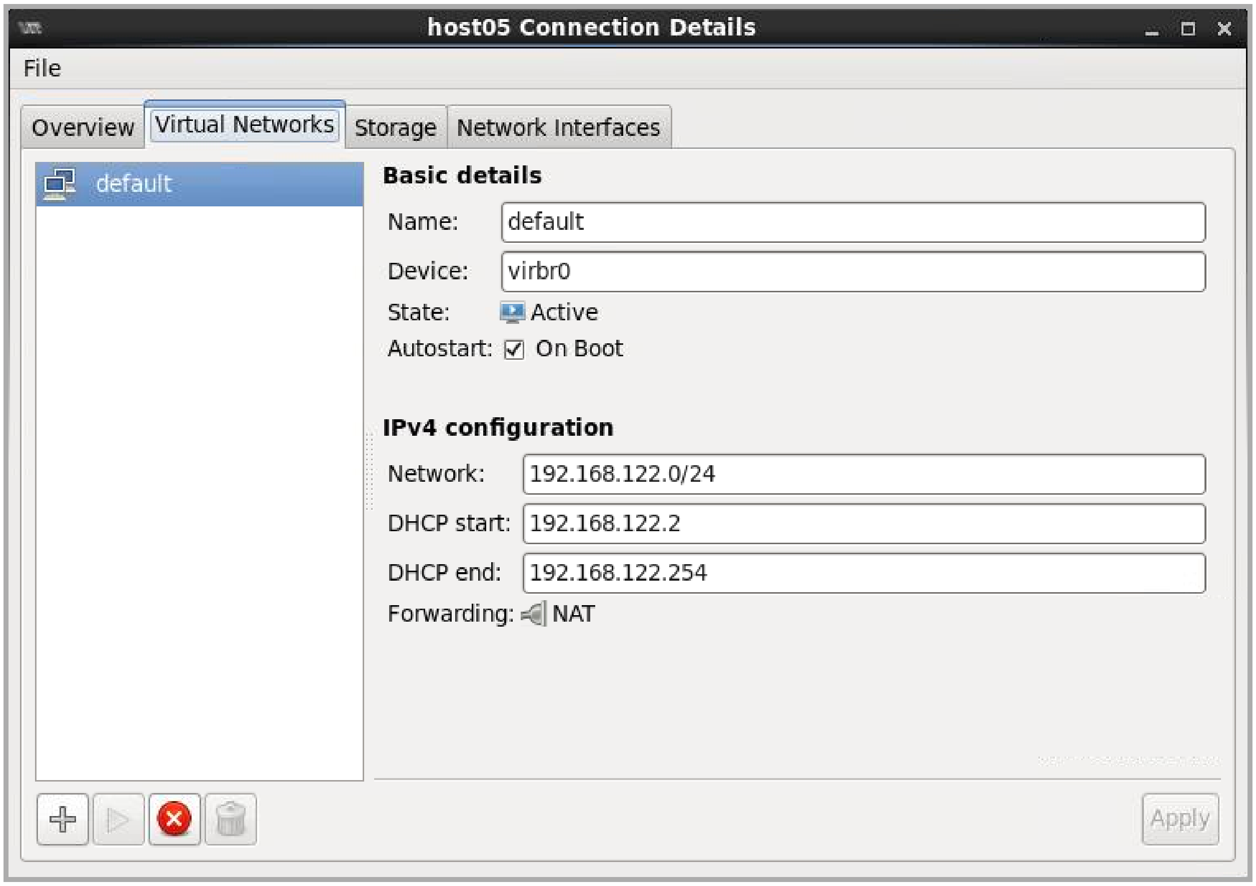


- The information for the **Overview** tab appears.



- This tab shows the type of hypervisor used, and the memory and virtual CPUs available to create virtual machines.
- Even when KVM is fully implemented, the Hypervisor field shows `qemu`. KVM and QEMU work together to run virtual machines.

c. Click the **Virtual Networks** tab to display information about virtual networks.



- There is only one virtual network available. This virtual network is called `default` and is configured automatically on **host05** when you start the Virtual Machine Manager for the first time.
- You use the `default` virtual network when configuring networking for the virtual machine that you create in the next practice.
- The default virtual network is associated with network device `virbr0`, which is a bridge. The network bridge acts like a virtual switch to which virtual machines connect.
- The `virbr0` bridge is not attached to any physical NIC (like `eth0` or `eth1`) on the virtualization host. Instead, NAT and IP forwarding are used to forward packets from the virtual guests to the external network. You can find more information about virtual networking at this site: http://wiki.libvirt.org/page/VirtualNetworking.

d. From a terminal window on **host05**, use the `brctl` command to display the bridge information for `virbr0`.

```
# brctl show
bridge name    bridge id           STP enabled      interfaces
virbr0         8000...             yes              virbr0-nic
```

e. You can also obtain information for `virbr0` by using the `ifconfig -a` command.

```
# ifconfig -a
...
eth0       Link encap:Ethernet
           inet addr:192.0.2.105
...
virbr0     Link encap:Ethernet
           inet addr:192.168.122.1
...
virbr0-nic Link encap:Ethernet
...
```

f. Use the `iptables -L` command to view the `iptables` rules that set up NAT.

- With NAT, any guests connected through the virtual network switch use the host IP address for communication to the outside world.

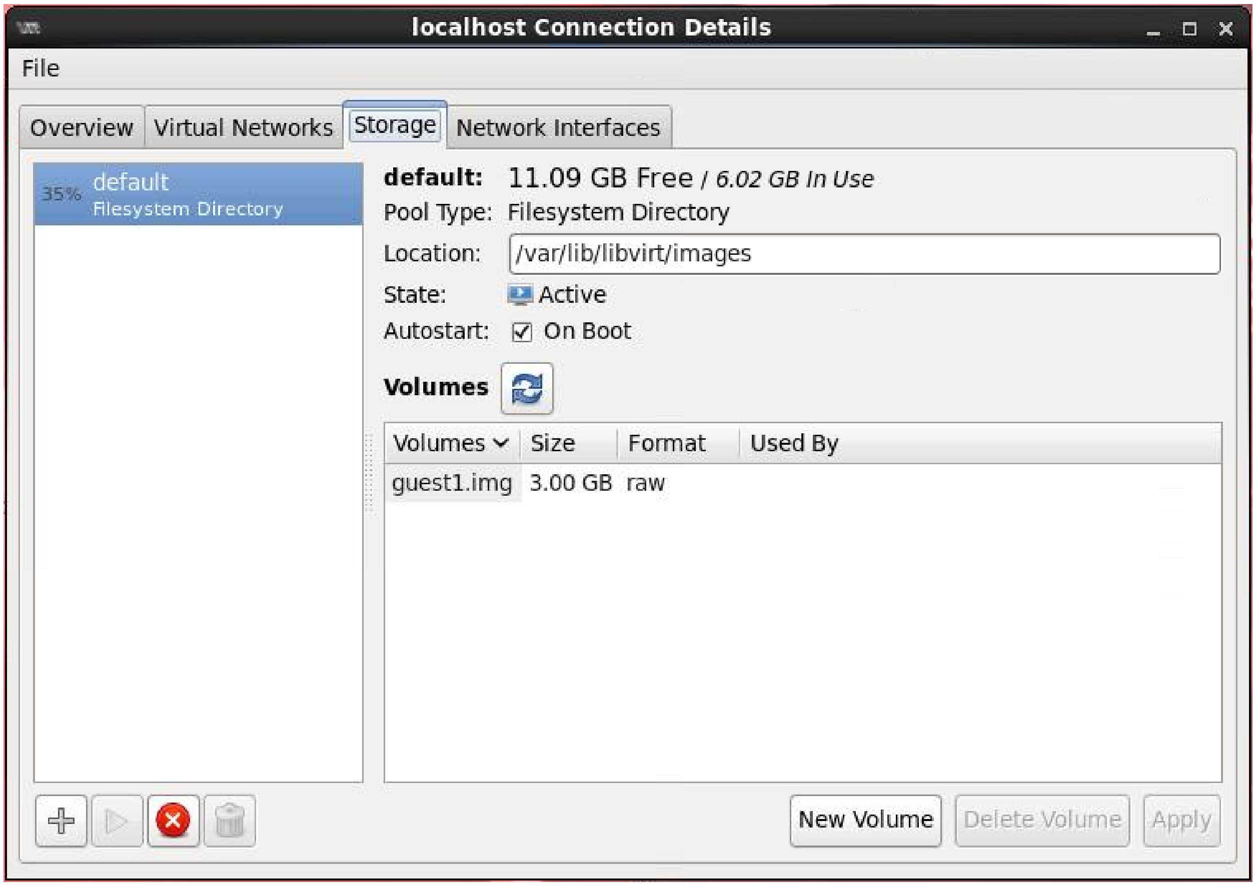
```
# iptables -L
...
Chain FORWARD (policy ACCEPT)
Target   prot opt source
ACCEPT   all  --  anywhere          192.168.122.0/24  state ...
ACCEPT   all  --  192.168.122.0/24  anywhere
...
```

g. Use the `ps -ef` command and pipe the output to `grep` to display the `dnsmasq` process.

- The `libvirt` API uses the `dnsmasq` program to provide each virtual network switch with a range of IP addresses that are provided to guests through DHCP.

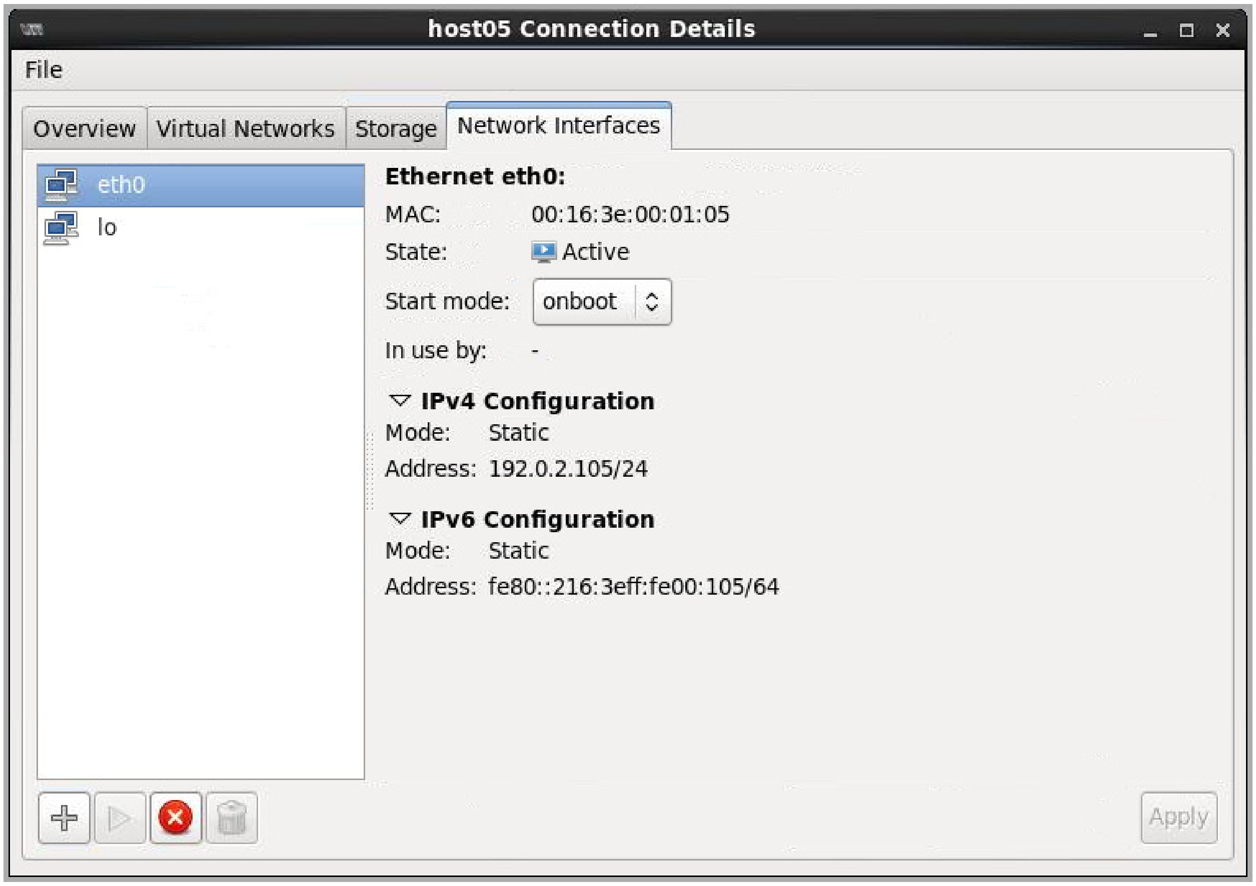
```
# ps -ef | grep dns
nobody ... /usr/sbin/dnsmasq ... --listen-address 192.168.122.1
--dhcp-range 192.168.122.2,192.168.122.254 ...
...
```

h. Click the **Storage** tab to display information about available storage and volumes in your KVM environment.



- By default, disk images for virtual machines are created in the `/var/lib/libvirt/images` directory, which is a storage pool of type file system. You can create additional pools of storage and these pools can reside on NFS shares or iSCSI volumes.
- There is one volume, `guest1.img`, in the Volumes area in the right pane for the default pool. As you create new virtual machines and assign storage to these virtual machines, new volumes appear in this window.
- You use volume `guest1.img` when creating your new virtual machine in the next practice for this lesson.

i. Click the **Network Interfaces** tab to display information about the NICs on **host05**.



- There is only one NIC available on **host05**. If other NICs are available, they show up in the left pane, and if a NIC is not configured, it is disabled.
- You can use available NICs to create new virtual networks, using NAT with IP forwarding, or bridged virtual networks in routed mode as described in the slides for this lesson.

j. From a terminal window on **host05**, verify the status of IP forwarding on **host05** by examining the contents of the `/etc/sysct.conf` file.

```
# grep forward /etc/sysctl.conf
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
```

- IP forwarding was enabled for you in **host05**. In other environments, you have to enable IP forwarding because `virt-manager` does not do it automatically for you. IP forwarding and NAT are needed for networking of virtual machines connecting to the `default` virtual network.

k. Close the Virtual Machine Manager **localhost Connection Details** window by clicking the `X` in the top-right corner of the window.

Practice 14-3: Creating a Virtual Machine

Overview

In this practice, you create a new virtual machine and import a disk image for this new virtual machine. The disk image already contains an installed OS, which means that you do not need to perform an OS installation for your new virtual machine.

Assumptions

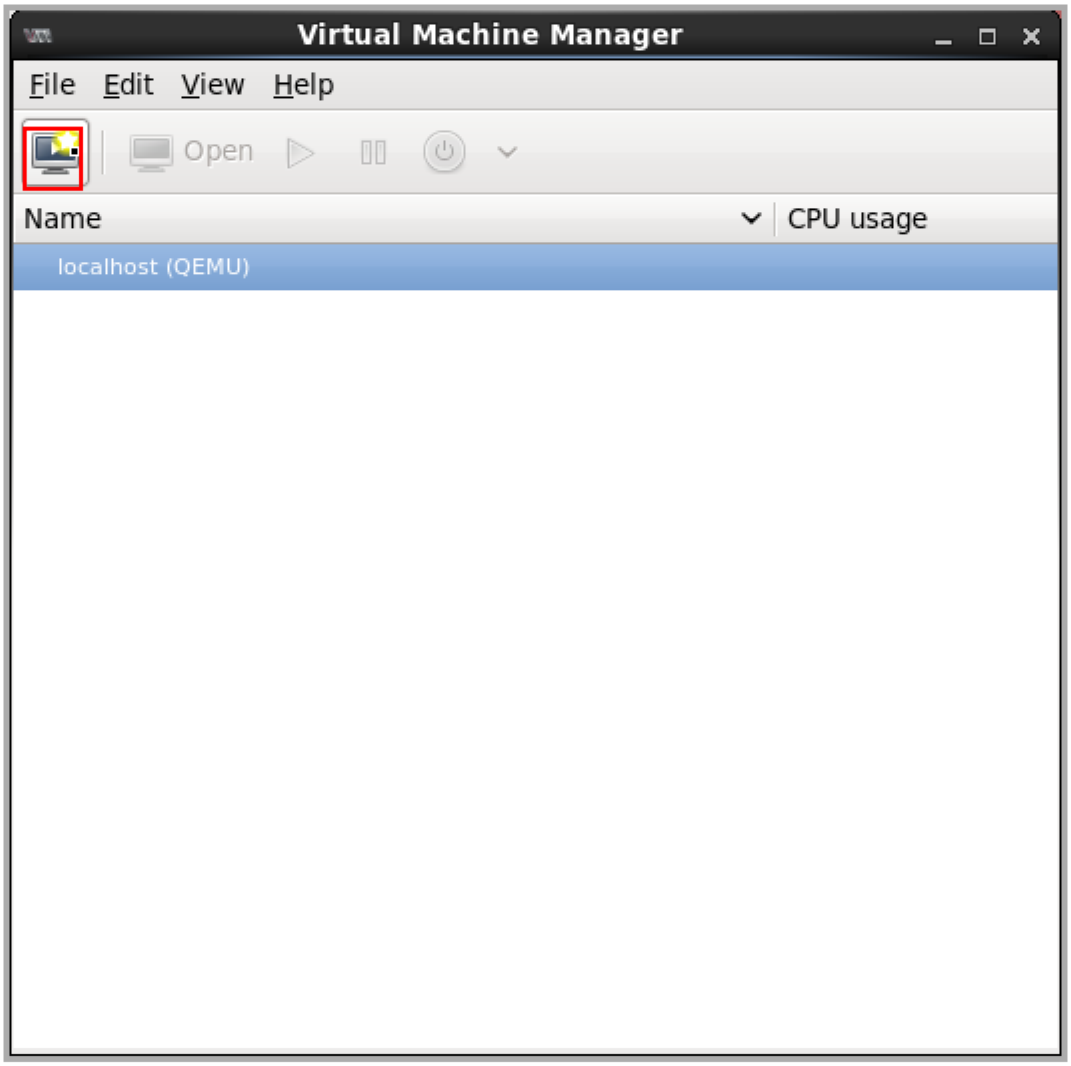
The existing disk image is called `guest1.img` and resides in the `/space` directory on **host05**.

Tasks

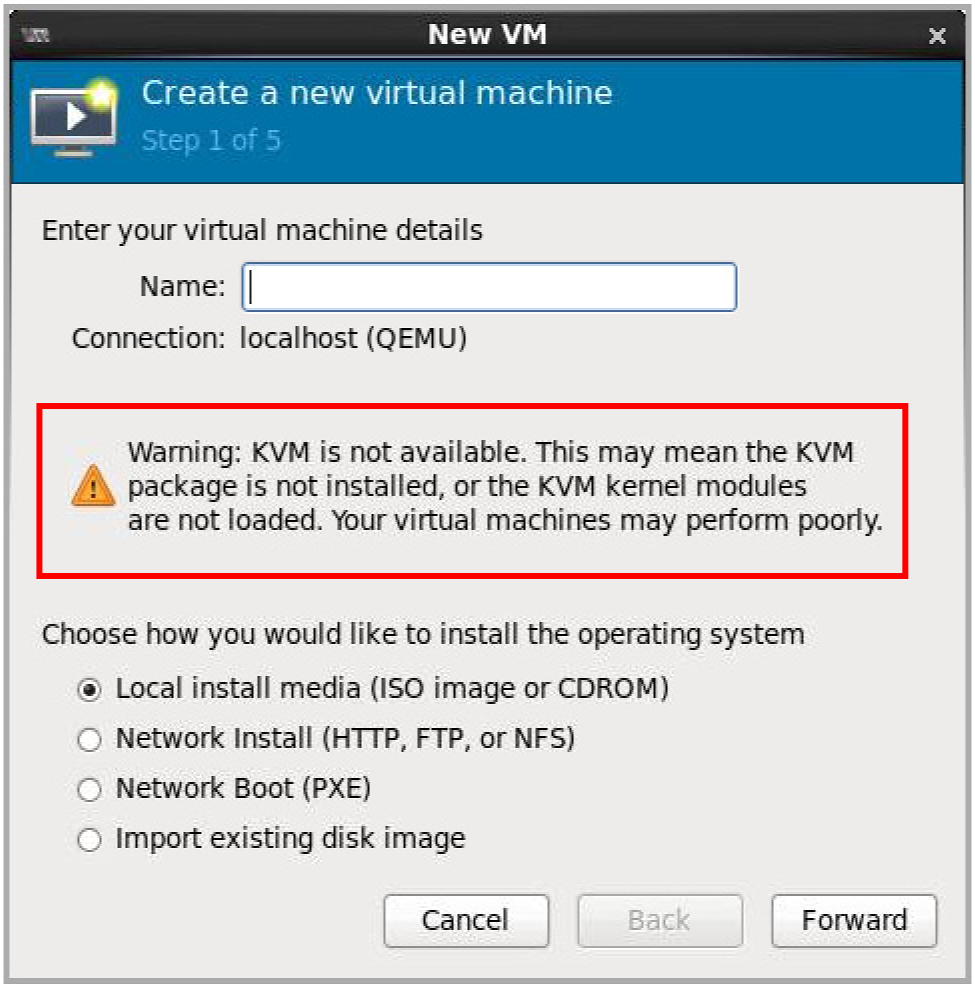
1. From a terminal window on **host05**, start the Virtual Machine Manager if it is not already started.

```
# virt-manager&
```

2. Click the "**Create a new virtual machine**" icon in the toolbar.

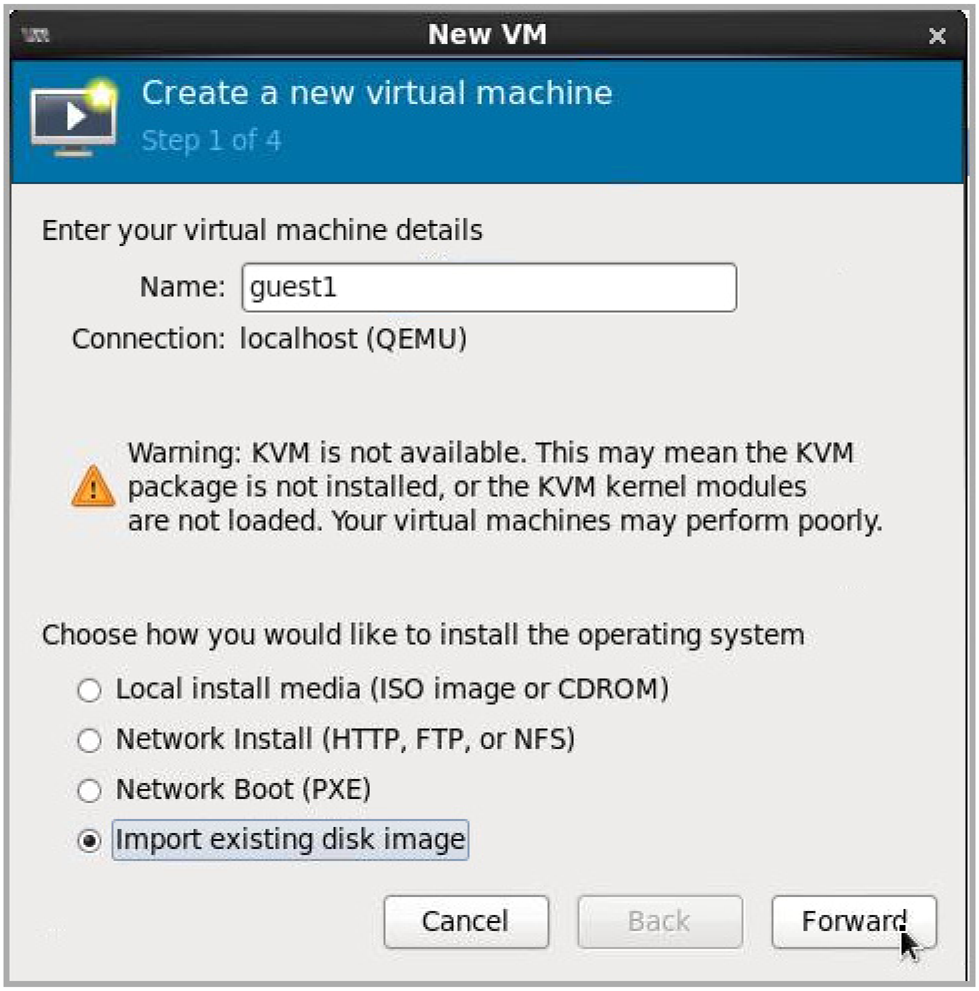


- Notice the warning in the window that appears.



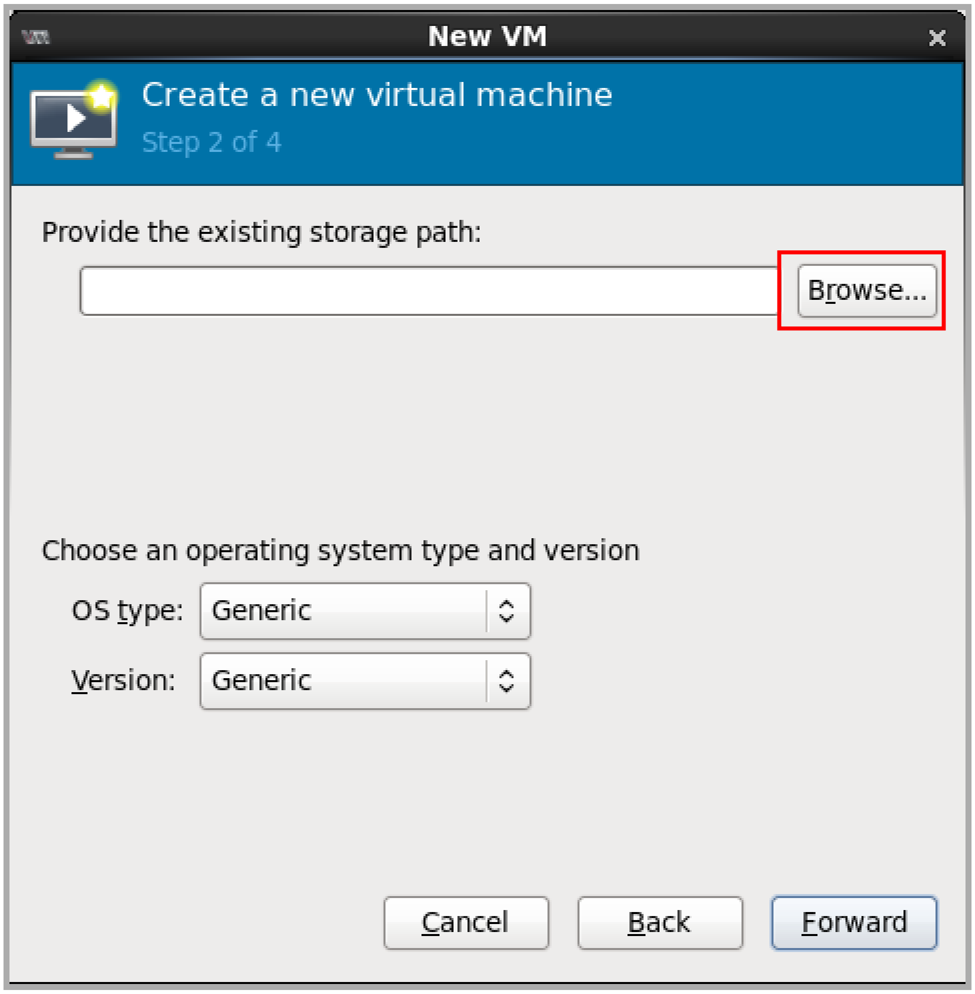
- KVM is not available, as discussed in the Practices Overview. You are still able to create and manage virtual machines, but the virtual machines will perform poorly.

3. In the “Create a new virtual machine, Step 1 of 5” window:
   - Enter `guest1` in the Name field for your virtual machine
   - Click the “**Import existing disk image**” radio button.

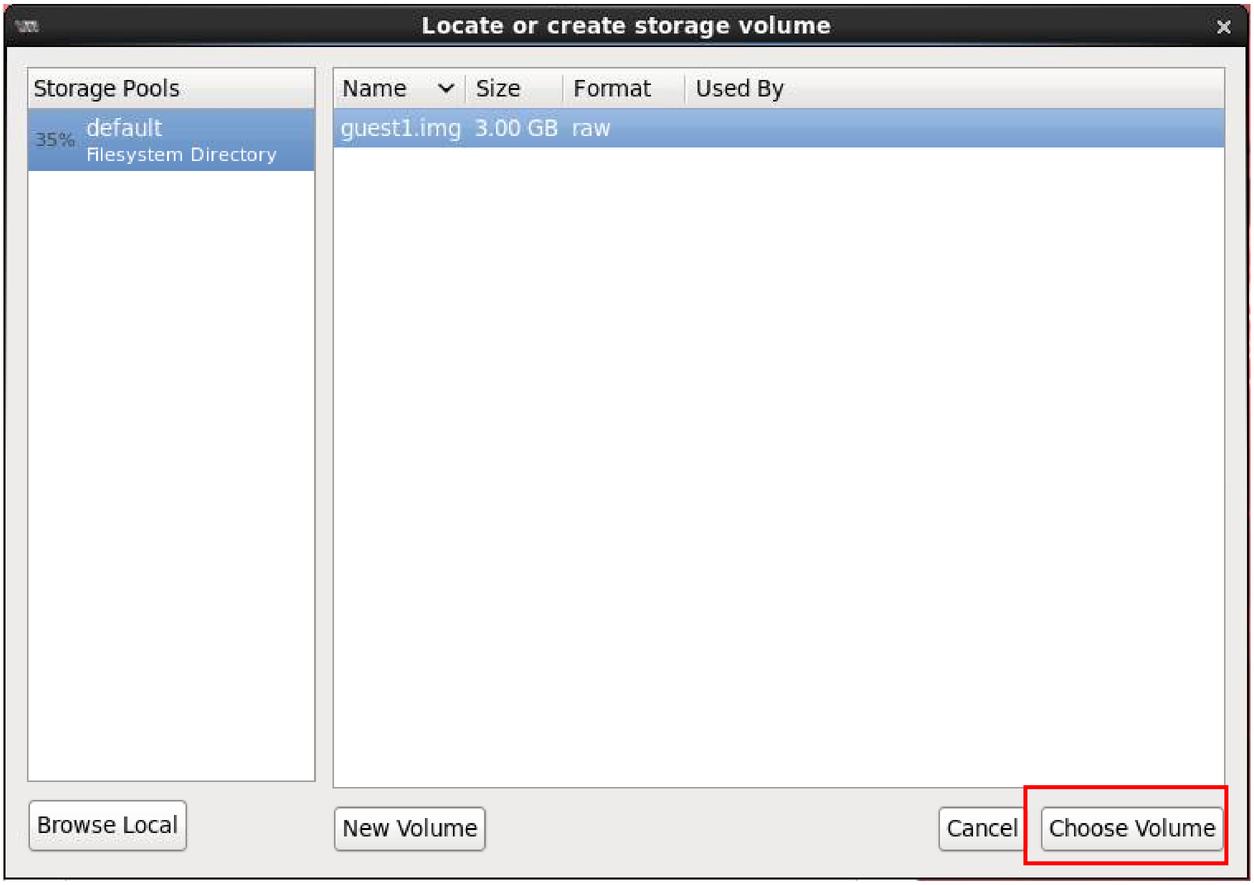


   - With this installation selection, steps go from Step 1 to 5 to Step 1 to 4.
   - Click the **Forward** button to continue.

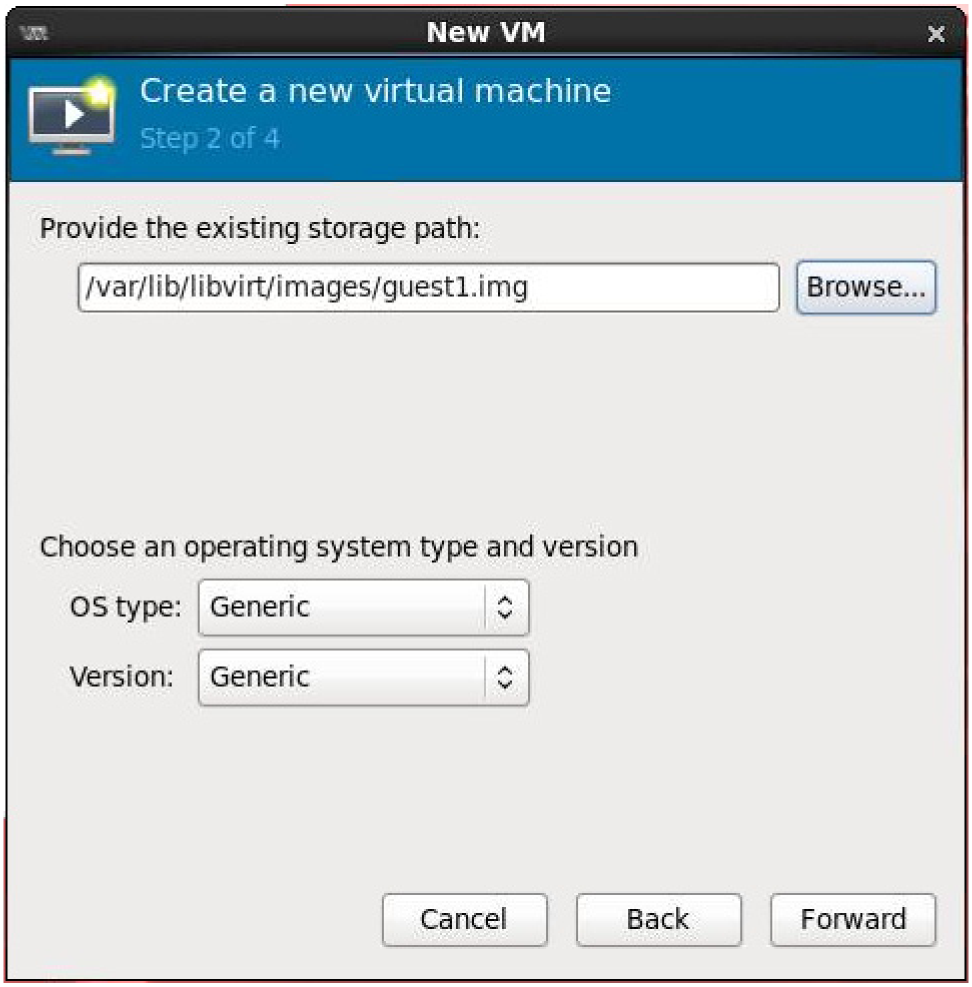
4. In the "Create a new virtual machine, Step 2 of 4" window, click the **Browse** button to locate the existing disk image for `guest1`.



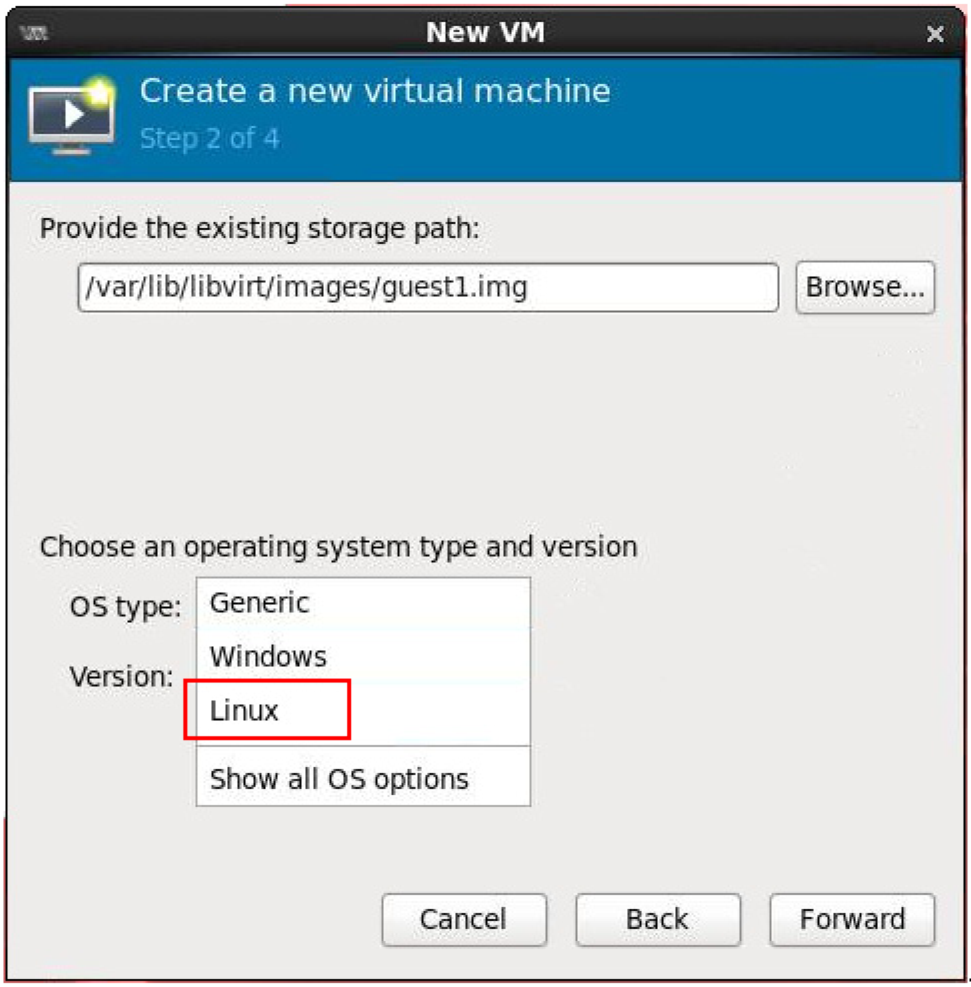
5. In the "Locate or create storage volume" window:
   - Highlight volume `guest1.img` in the right pane
   - Click the **Choose Volume** button



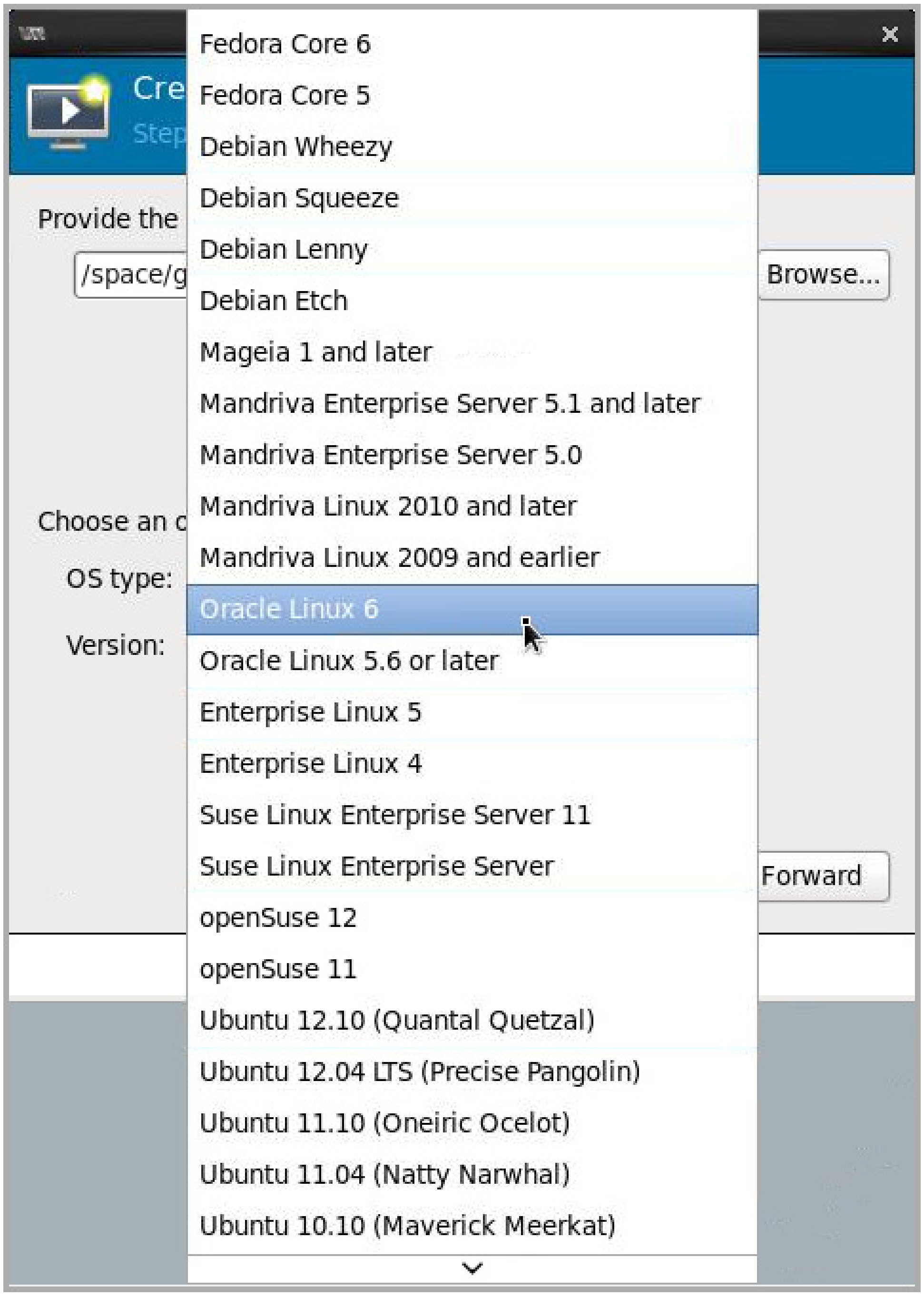
- You are returned to the Step 2 of 4 window, and the selected volume appears in the storage path field.



6. To choose the appropriate operating system type and version, click the spinbox for **OS type** and find the **Linux** selection.



7. Select the **OS Version**.
   a. Select the appropriate variant for Linux by clicking the **Version** spinbox and locating **Oracle Linux 6**.
      - You might have to click "**Show all OS options**" to force the display of all variants of Linux.
      - Use the down arrow at the bottom of the list to scroll to **Oracle Linux 6**.

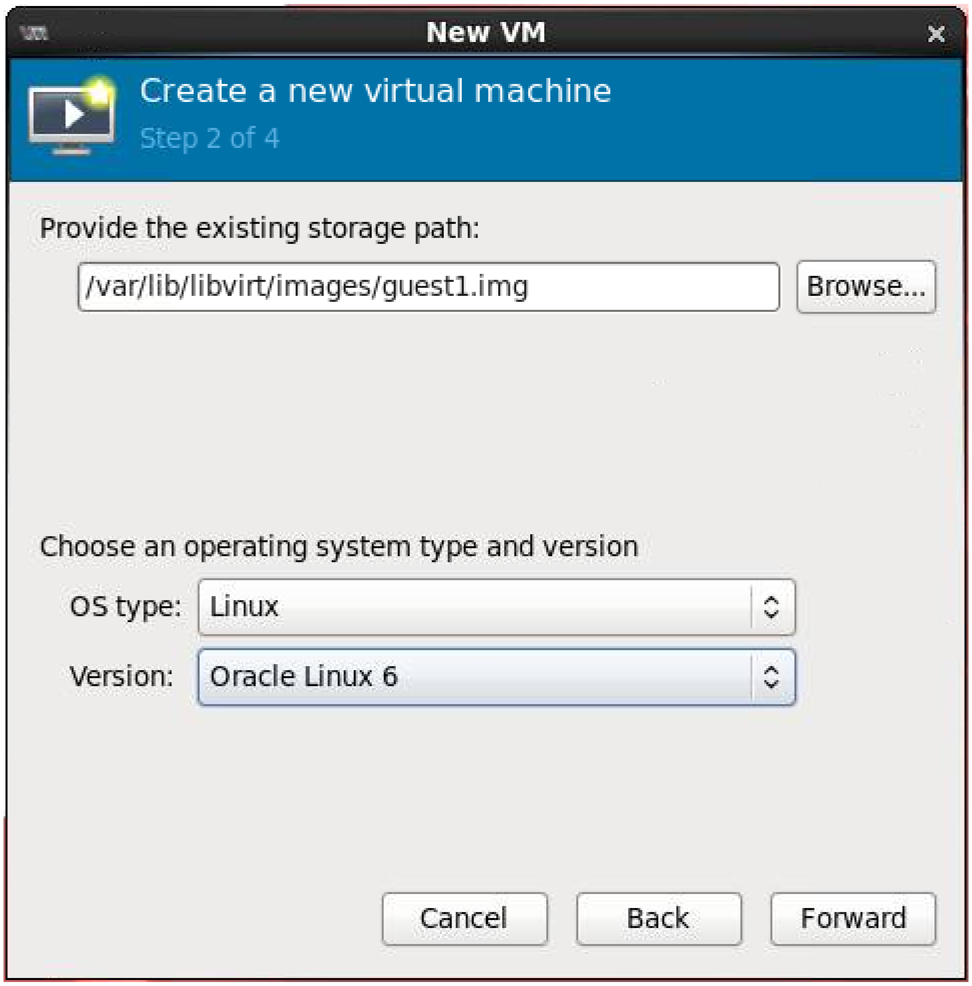


b. Run the following command from a terminal window to display the list of available OS versions.

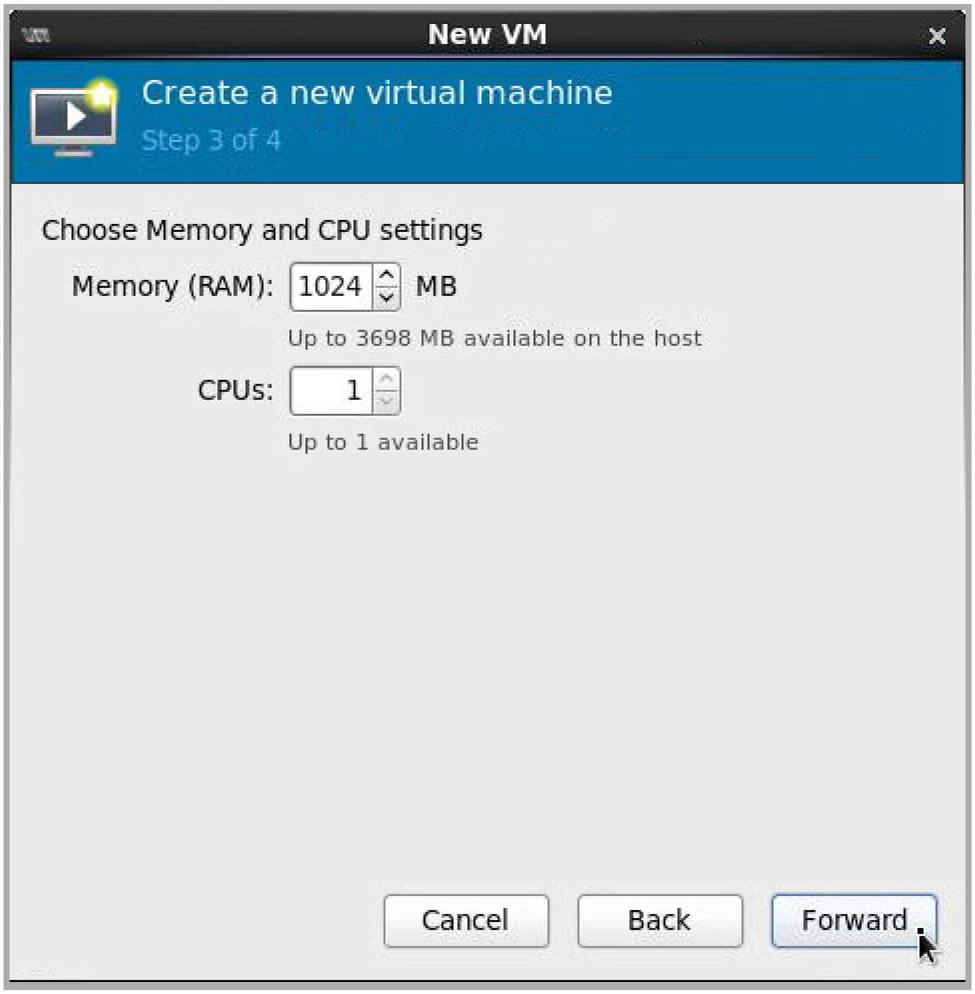
```
# virt-install --os-variant list
win7                    : Microsoft Windows 7
vista                   : Microsoft Windows Vista
winxp64                 : Microsoft Windows XP (x86_64)
winxp                   : Microsoft Windows XP
...
oel6                    : Oracle Linux 6    ← Select this variant
```

```
oel5.6                    : Oracle Linux 5.6 or later
oel5                      : Enterprise Linux 5
oel4                      : Enterprise Linux 4
rhel7                     : Red Hat Enterprise Linux 7
...
generic26                 : Generic 2.6.x kernel
generic24                 : Generic 2.4.x kernel
```

c. After selecting the **OS type** as **Linux** and the **Version** as **Oracle Linux 6**, click the **Forward** button.

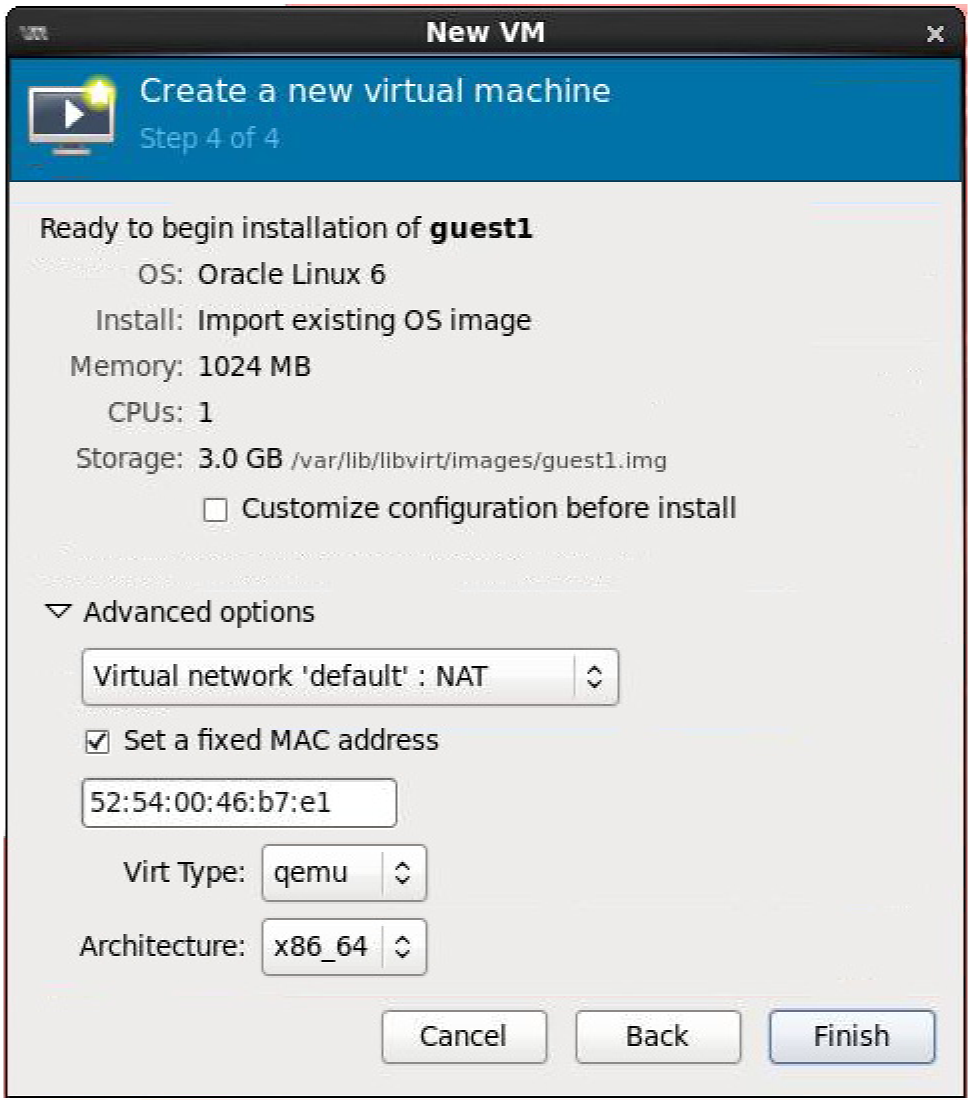


8. In the “Create a new virtual machine, Step 3 of 4” window, accept the default Memory size and CPUs number for the new virtual machine.

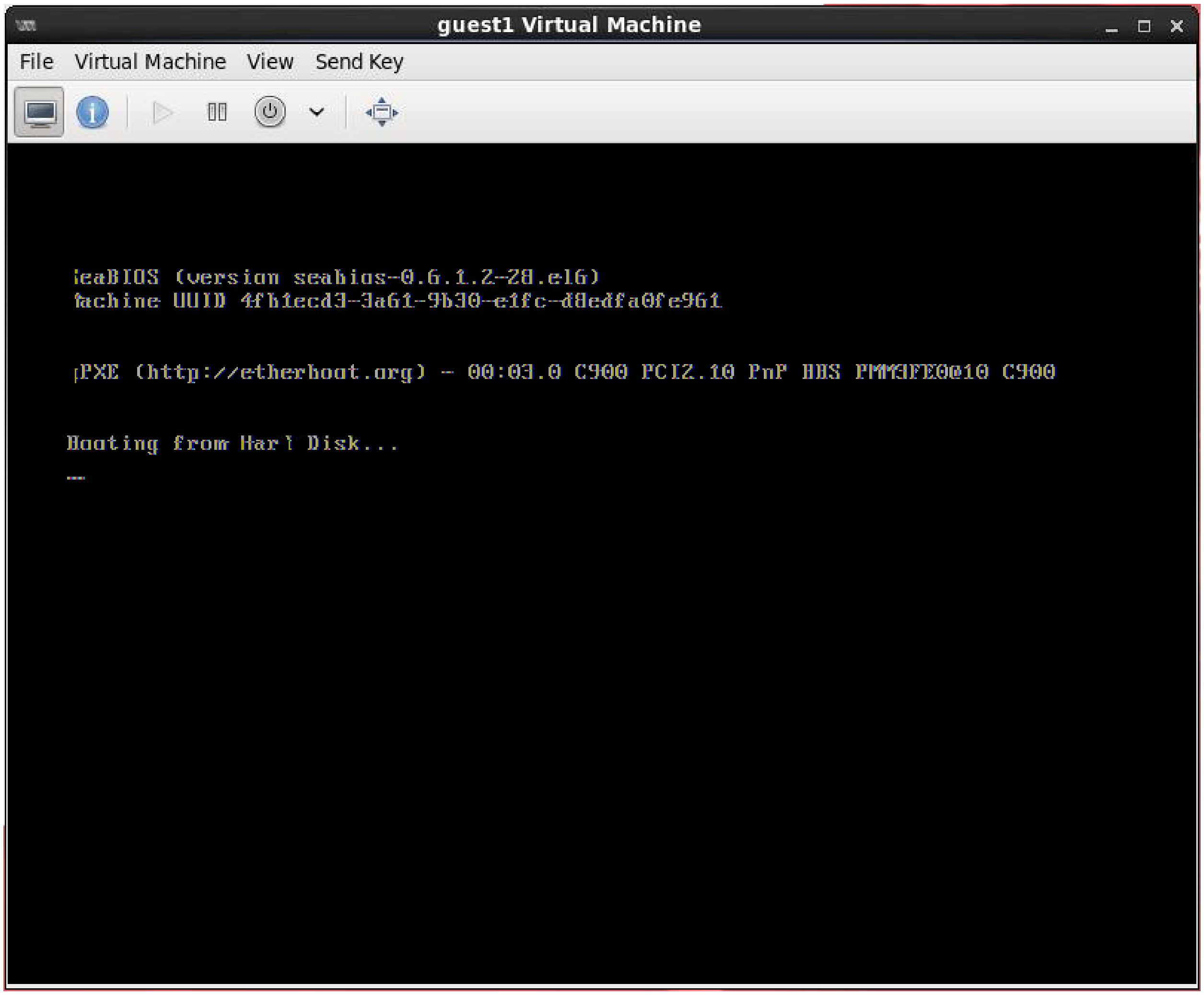


- Click **Forward** to continue.

9. In the “Create a new virtual machine, Step 4 of 4” window, click the expand button to display **Advanced** options.



- Notice the MAC address assigned to your new virtual machine. You can change this MAC address if necessary; however, in this task, leave the MAC address to its automatically assigned value. The value of the MAC address is different in your environment.
- Click **Finish** to complete the creation of your virtual machine.
- The virtual machine is created and starts to boot.
- You are automatically connected to the virtual machine console.



- In your lab environment, your virtual machine is not taking advantage of the acceleration provided by KVM. This limitation of your lab environment is discussed in the Practices Overview section Without the KVM acceleration, the boot process is very slow.
- In the next practice, you manipulate your virtual machine while it is still booting.
- If you no longer have control of your cursor because it is used by the virtual machine, press the Ctrl + Alt key combination to regain control of the cursor.

Practice 14-4: Managing Your Virtual Machine

Overview

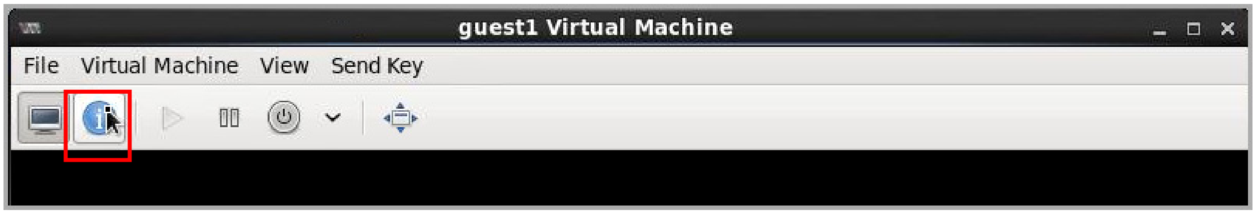
In this practice, you perform various operations on your virtual machine.

Assumptions

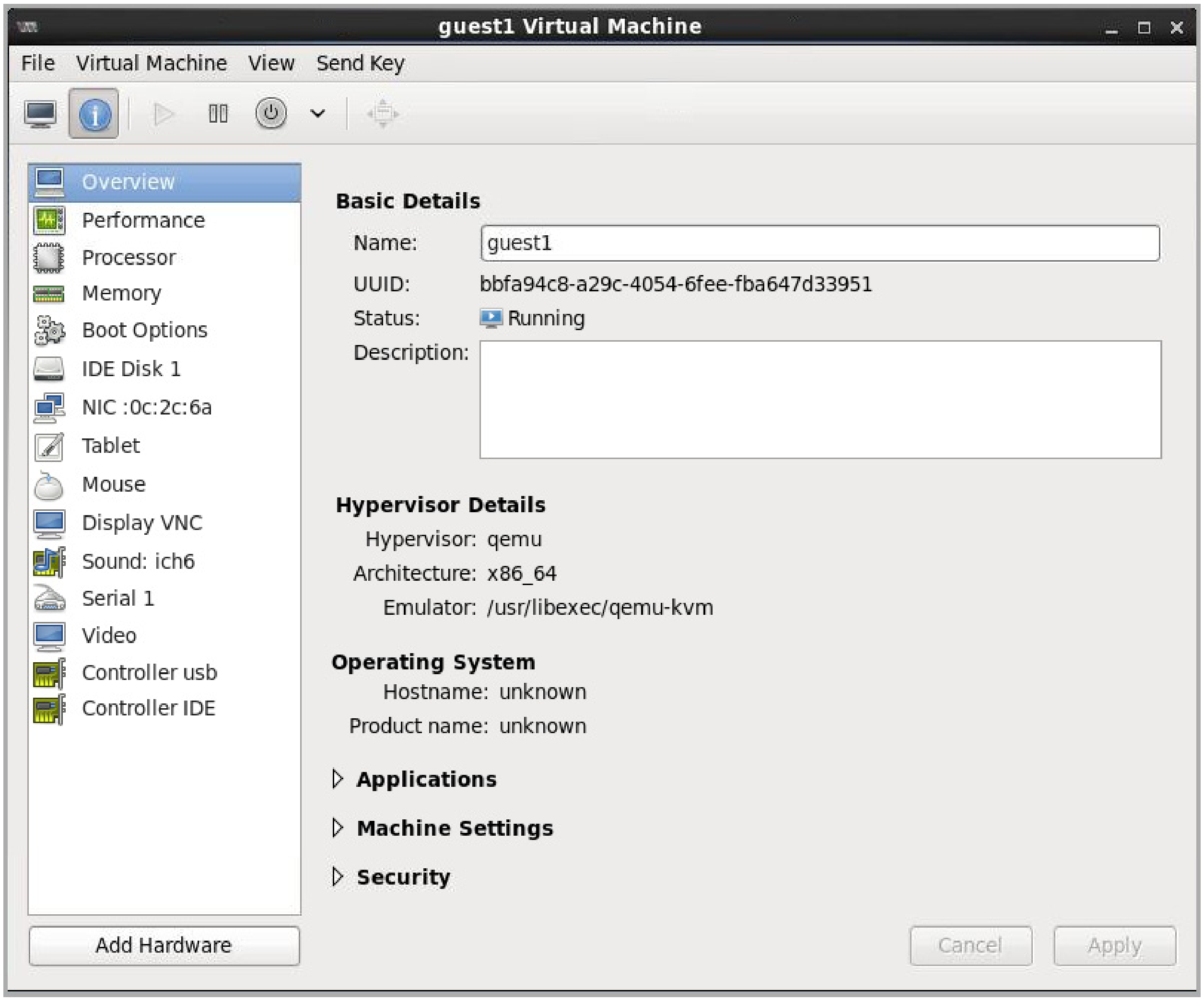
This practice assumes that you have the virtual machine console window opened, as shown in the last step of the previous practice.

Tasks

1. Using the Virtual Machine Manager, display information about your `guest1` virtual machine.
    a. In the console window of your virtual machine, click the information icon to show the virtual hardware details.

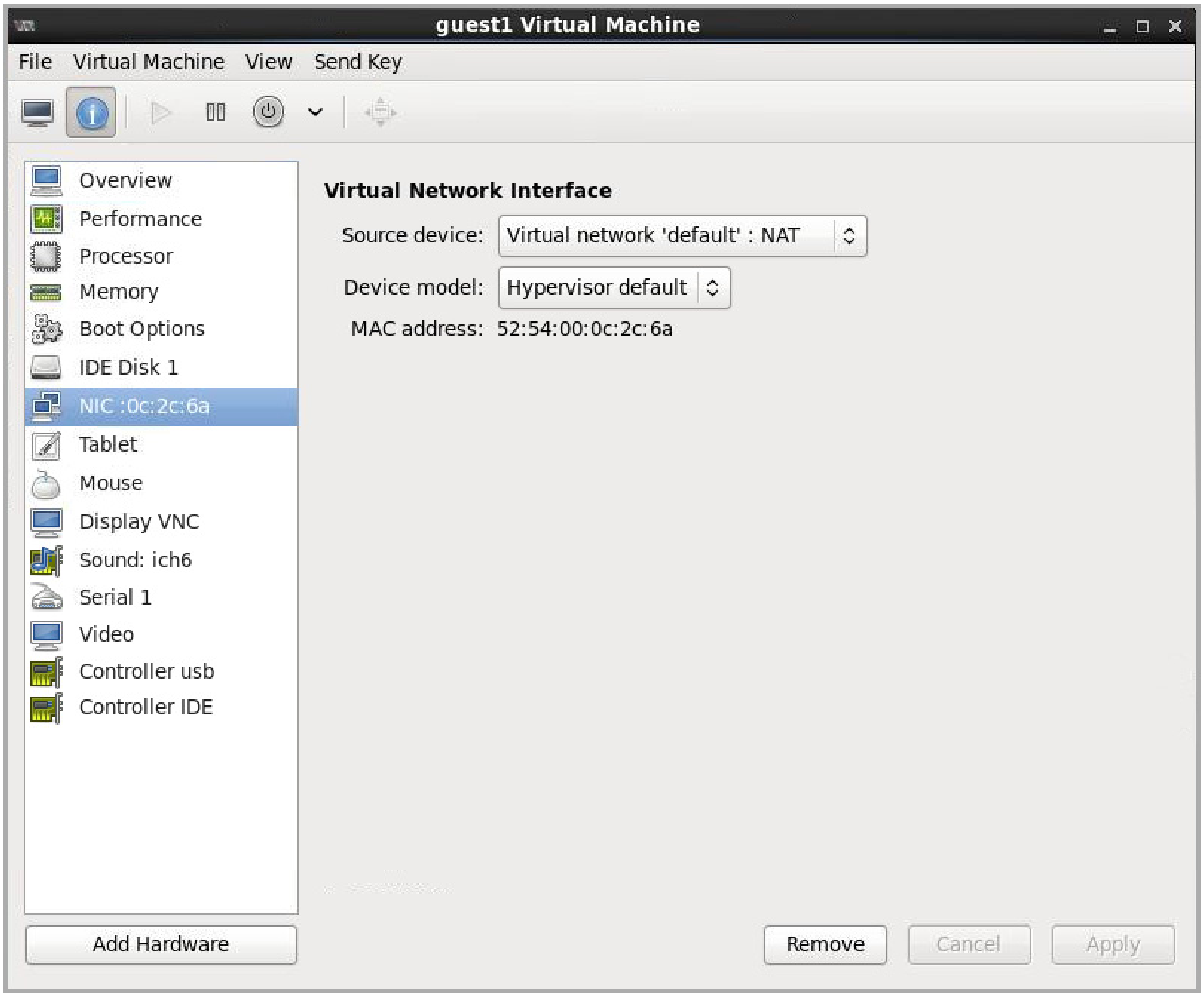


- The **Overview** perspective appears.



- Note the following items:
  - The UUID located in the Basic Details section. The UUID is a unique identification number that is assigned automatically to your virtual machine.
  - The status of the virtual machine: Running.
  - The hypervisor, which shows as `qemu`. If your virtualization host supports virtualization hardware assist, the hypervisor shows as `kvm`.

b. Click the **NIC** perspective in the left pane.



- The single NIC is connected to the virtual network called `default`.
- The Device Model indicates Hypervisor default. You change this setting later in this practice.

c. Click other perspectives and examine the information available from these perspectives.

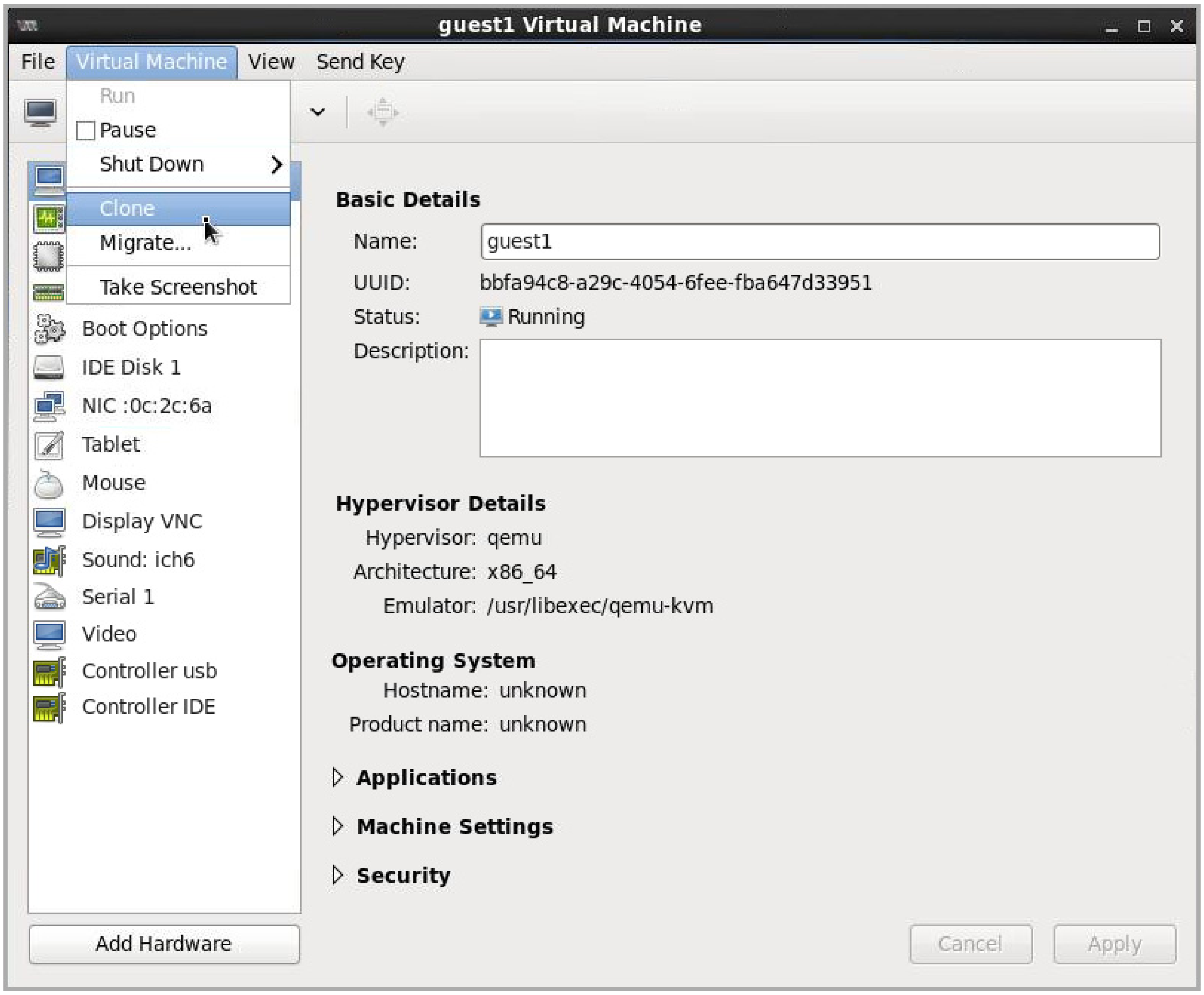
d. Leave the Virtual Machine Details window open.

2. Clone the virtual machine `guest1`.

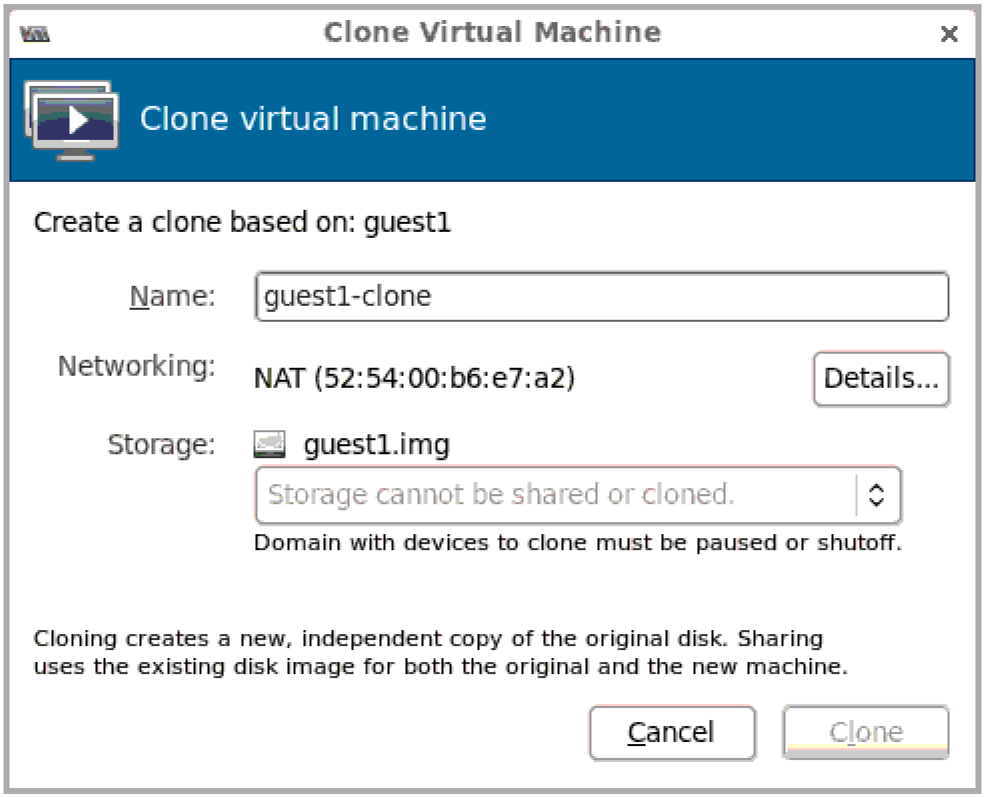
- The cloning operation creates a new virtual machine and assigns it a new MAC address and a new UUID. The clone has a virtual disk that is a copy of the virtual disk owned by the source virtual machine used for the cloning operation.
- Before cloning a virtual machine, you install all the applications and perform the necessary configuration in the operating system of the virtual machine. After customizing your virtual machine, you remove machine-specific information from the virtual machine so that clones of the virtual machine acquire their own identity. You can find more information about this process at this location: https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.0/html/Evaluation_Guide/Evaluation_Guide-Create_RHEL_Template.html

- In this task, you clone the `guest1` virtual machine from the Virtual Machine Manager. You can also use the `virt-clone` command to create a clone from an existing virtual machine.

a. From the Virtual Machine Details window, either showing the console or virtual machine information, select **Virtual Machine > Clone** from the menu bar.



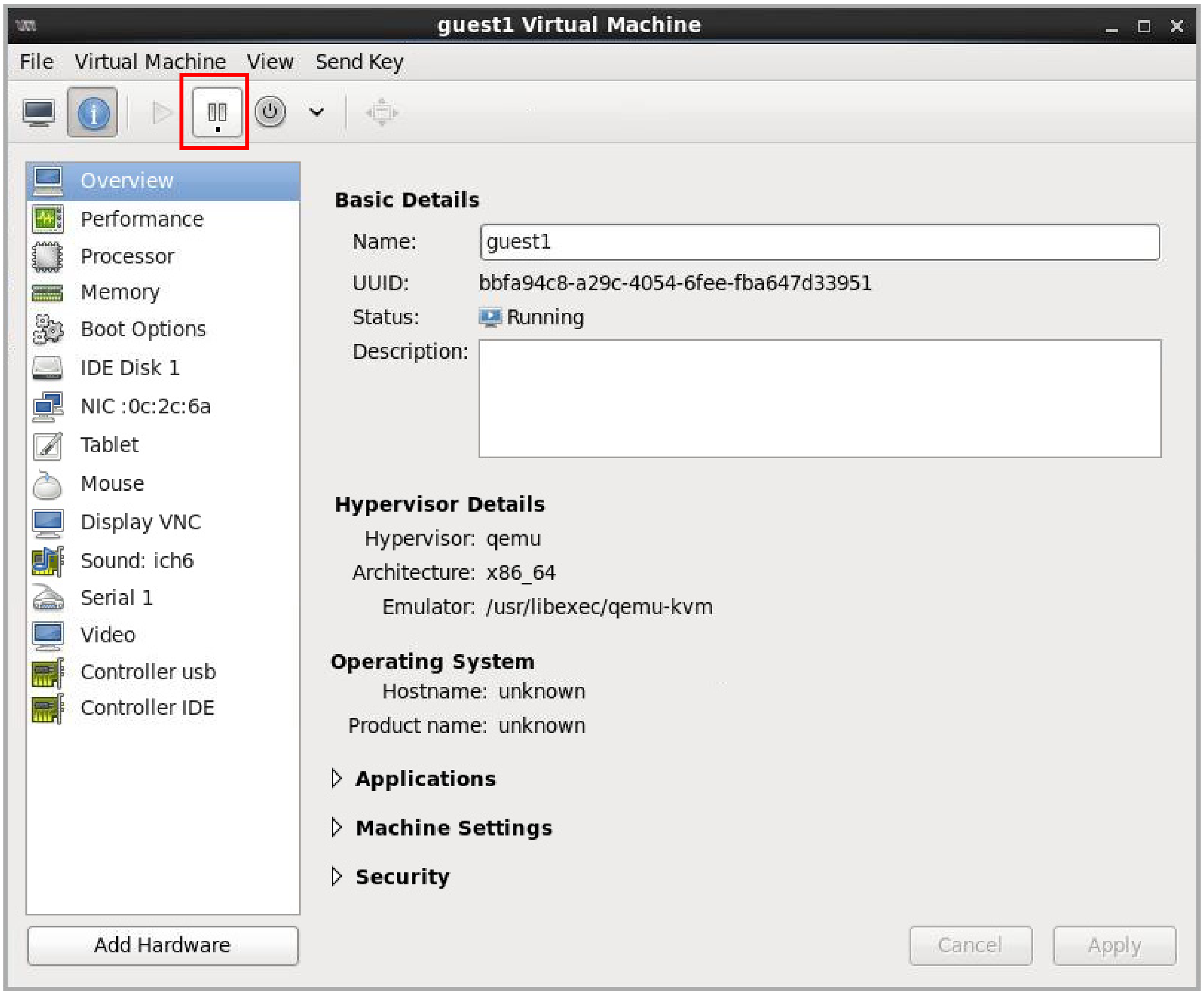
- The Clone Virtual Machine window appears.



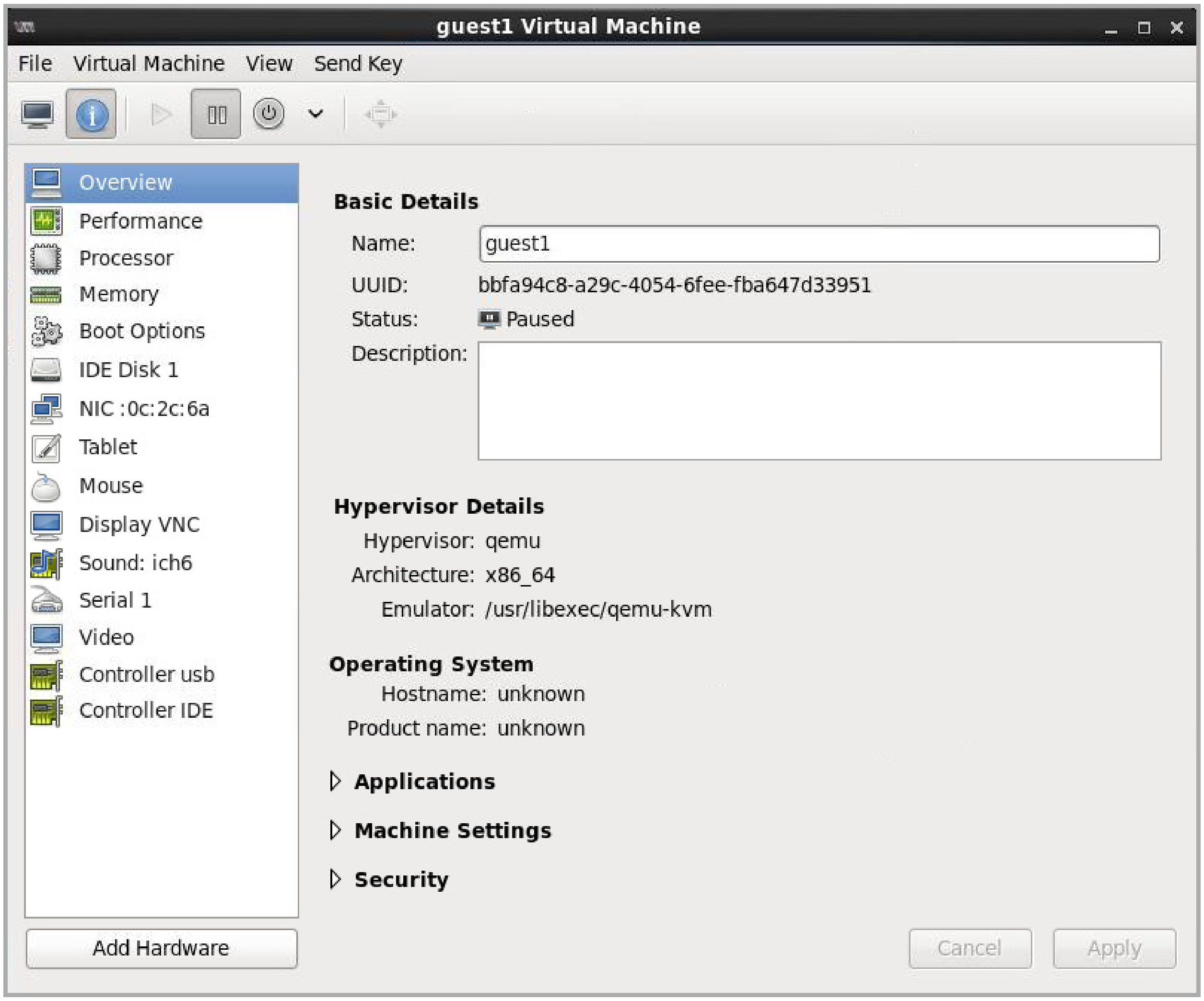
- You cannot clone the virtual machine, because it is in a running state. You must either pause it or shut it down to clone it.
- To capture a valid state when cloning a virtual machine, it is preferable to pause the virtual machine when it is idle. Because your virtual machine is still booting, the clone created with the following operation might not boot properly.
- The Red Hat Enterprise Virtualization solution offers hot cloning, which means that you can clone a running virtual machine.

b. Click the **Cancel** button on the Clone Virtual Machine window.

c. From the Virtual Machine Details window, either showing the console or virtual machine information, click the **Pause** button on the toolbar.

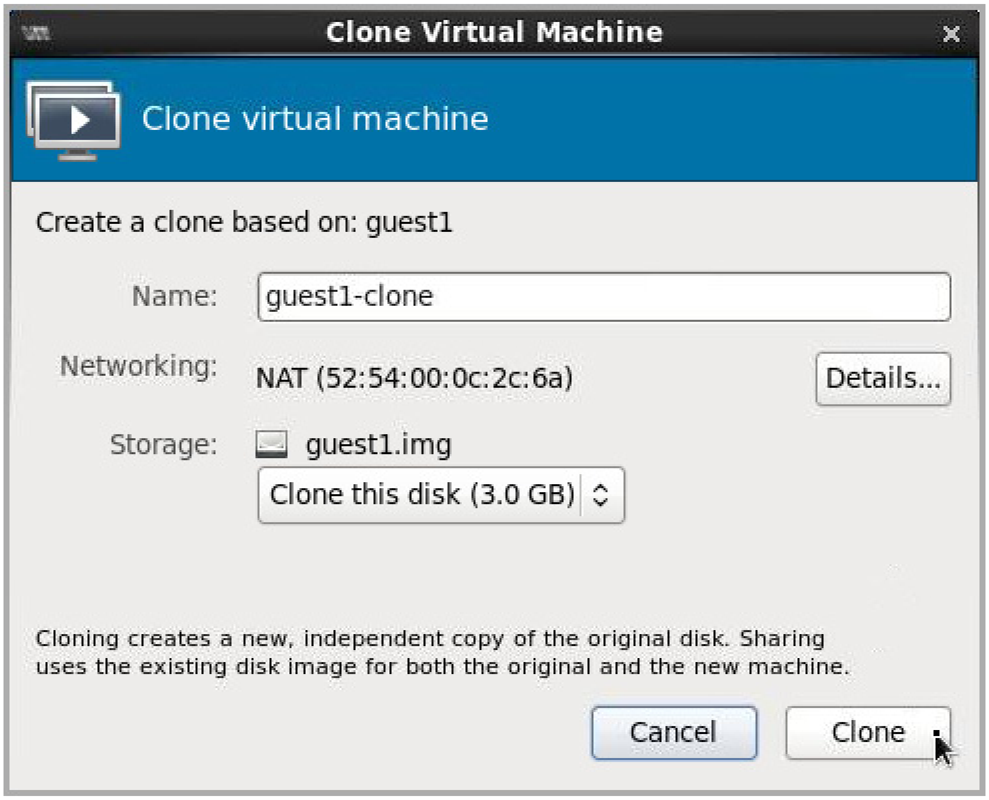


- The status of the virtual machine changes from Running to Paused.

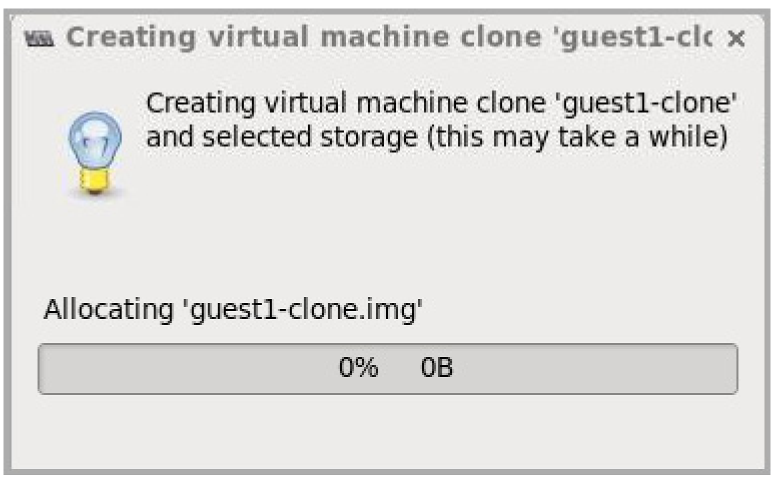


d. From the Virtual Machine Details window, attempt to clone the virtual machine again by selecting **Virtual Machine > Clone** from the menu bar.

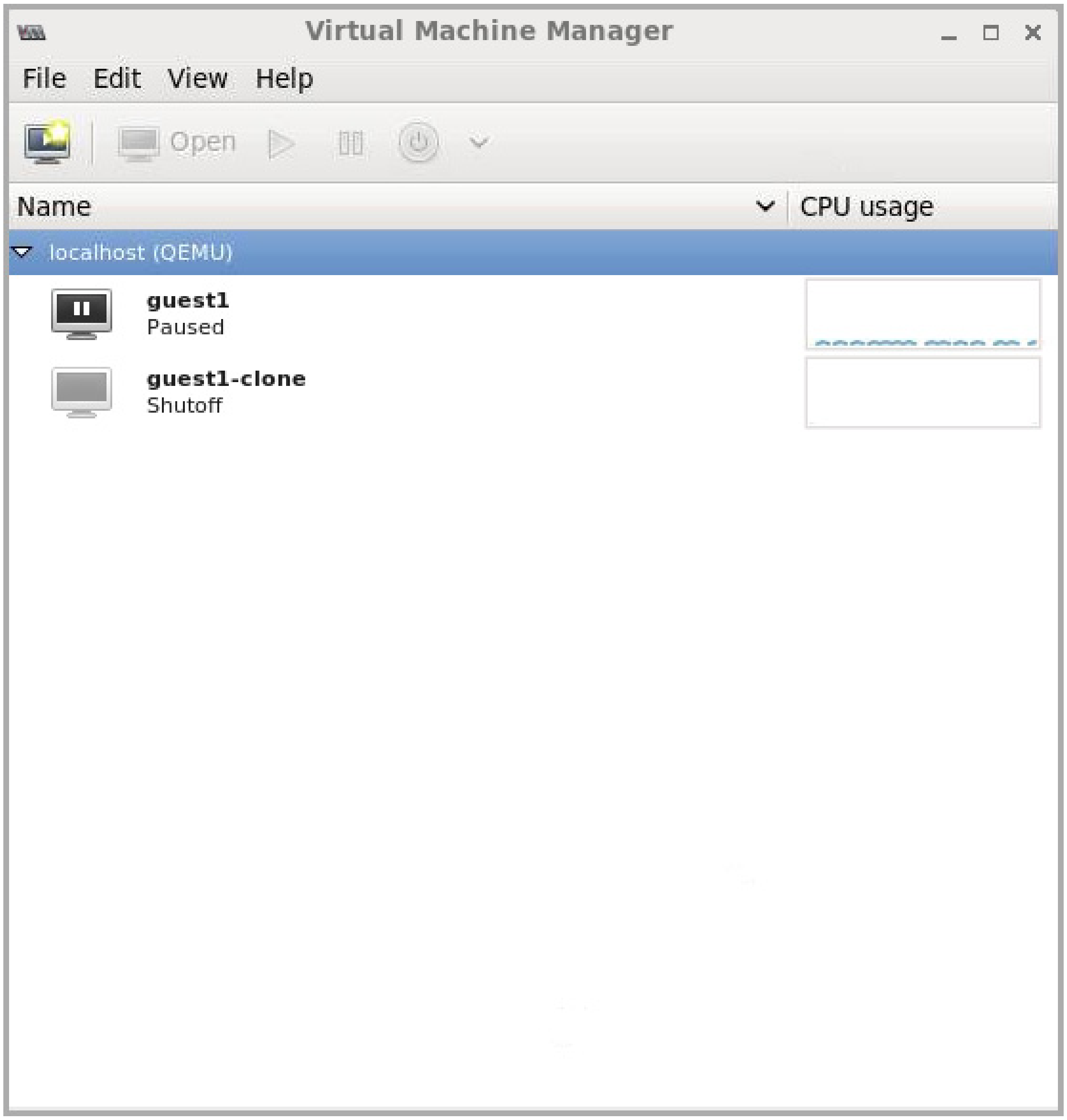
e. Click the **Clone** button in the Clone Virtual Machine window.



- The cloning operation starts.



f. After the cloning completes, select **File > View Manager** from the menu bar to return to the main Virtual Machine Manager window.
   - The window displays the new clone.

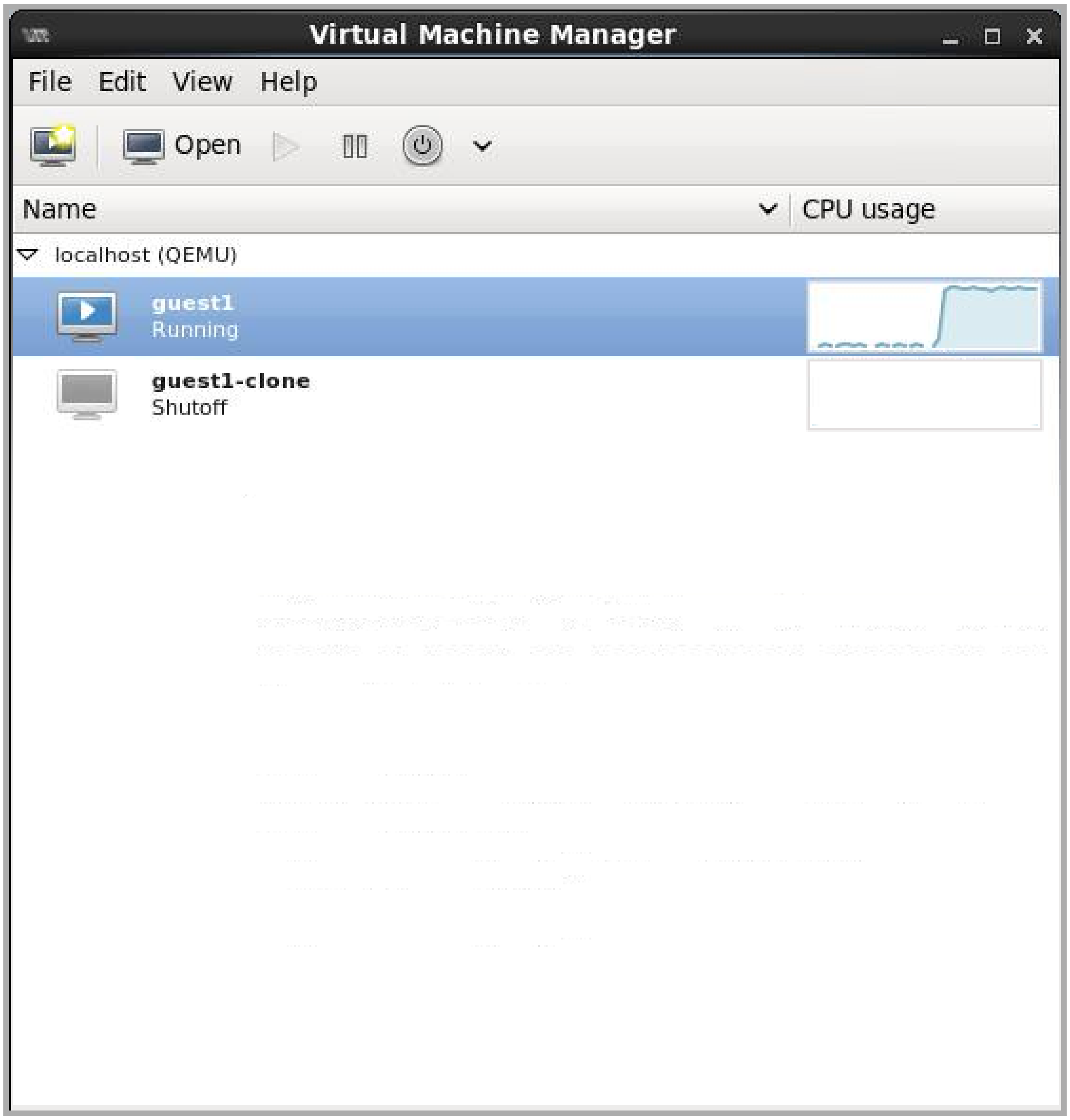


- The new clone, called `guest1-clone`, is independent from `guest1`, the virtual machine used to create the clone. You can change the parameters and options for the clone without affecting the virtual machine used for the cloning operation.

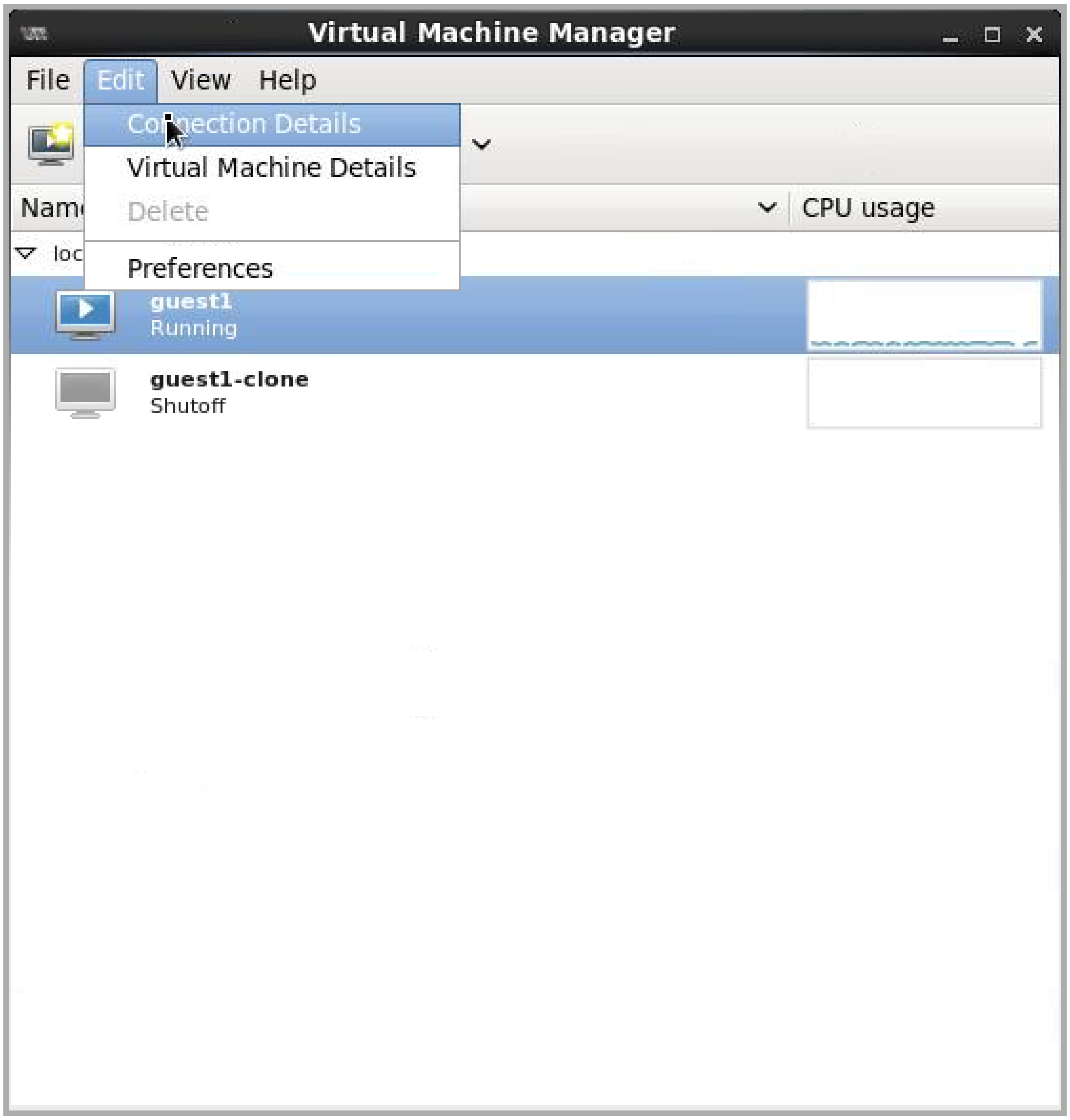
g. Resume operation for the `guest1` virtual machine, by choosing one of the following two methods:
   - Right-click `guest1` in the Virtual Machine Manager main window, and select Resume from the shortcut menu.
   - Open a terminal window on **host05** and execute the following command: `virsh resume guest1`.

```
# virsh resume guest1
Domain guest1 resumed
```

h. The `guest1` virtual machine resumes its booting.

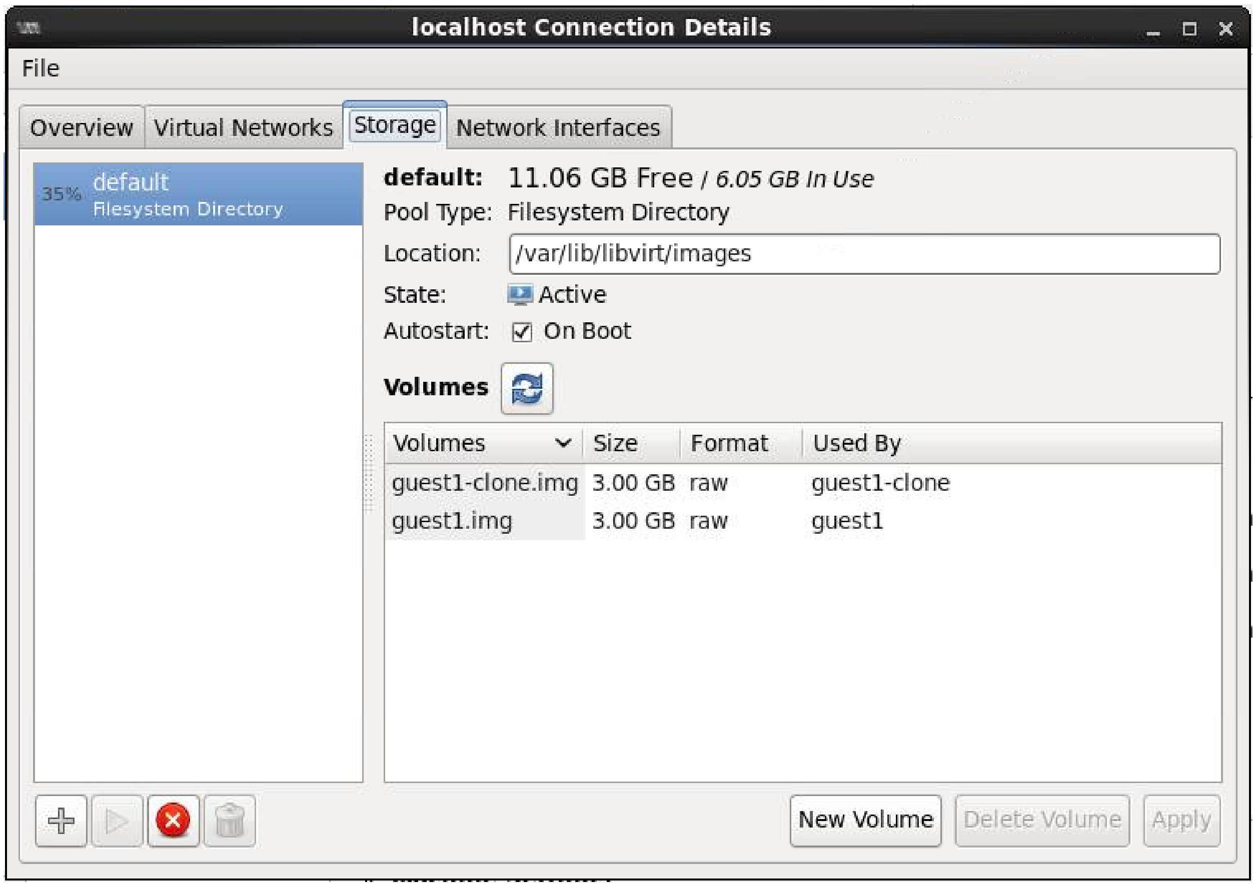


i. Select **Edit > Connection Details** in the menu bar.



j. In the **host05** Connection Details window:

- Click the **Storage** tab



- The virtual disk for the clone appears in the Volumes pane.

3. Add paravirtualization for the virtual machine.
   - In this task, you add paravirtualization support for the virtual NIC in the `guest1` virtual machine. A paravirtual driver is virtualization aware and cooperates with the hypervisor for improved performance.

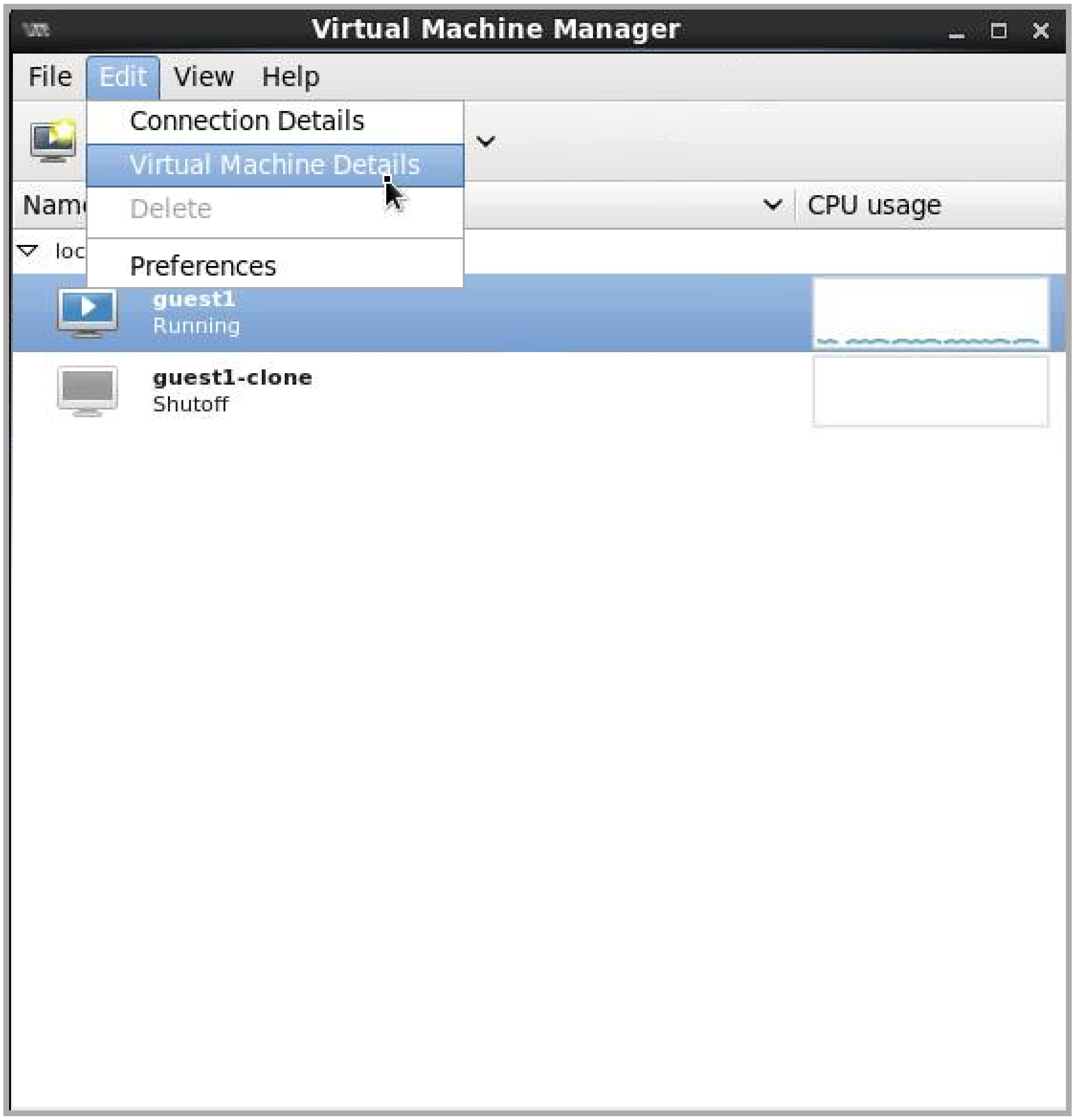
   a. Dump the virtual machine configuration file by using the `virsh dumpxml` command.

```
# virsh dumpxml guest1
<domain type='qemu' id='1'>
...
    <interface type='network'>
      <mac address='52:54:00:0c:2c:6a'/>
      <source network='default'/>
      <target dev='vnet0'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00'
    slot='0x03' function='0x0'/>
    </interface>
...
</domain>
```

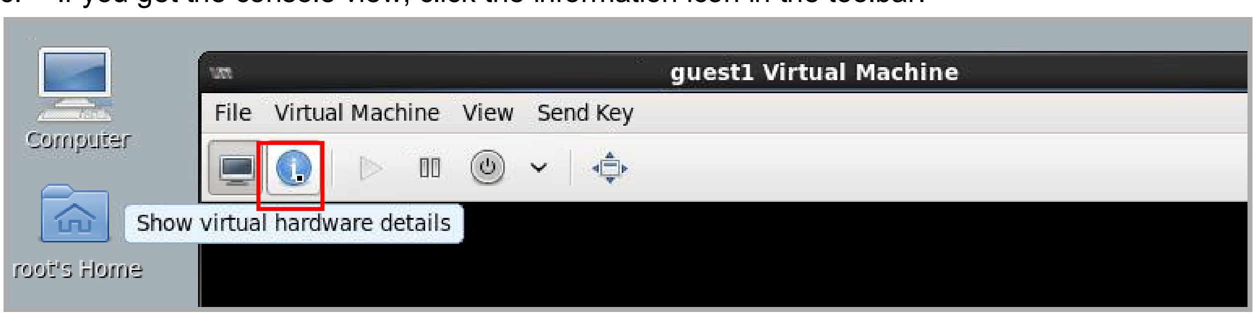
   - The configuration file for the `guest1` virtual machine is in XML format.

- Note the network interface element of type `network` shown in the previous display. The NIC operations are fully emulated.

b. In the Virtual Machine Manager main window, highlight `guest1`. Select **Edit > Virtual Machine Details** in the menu bar.



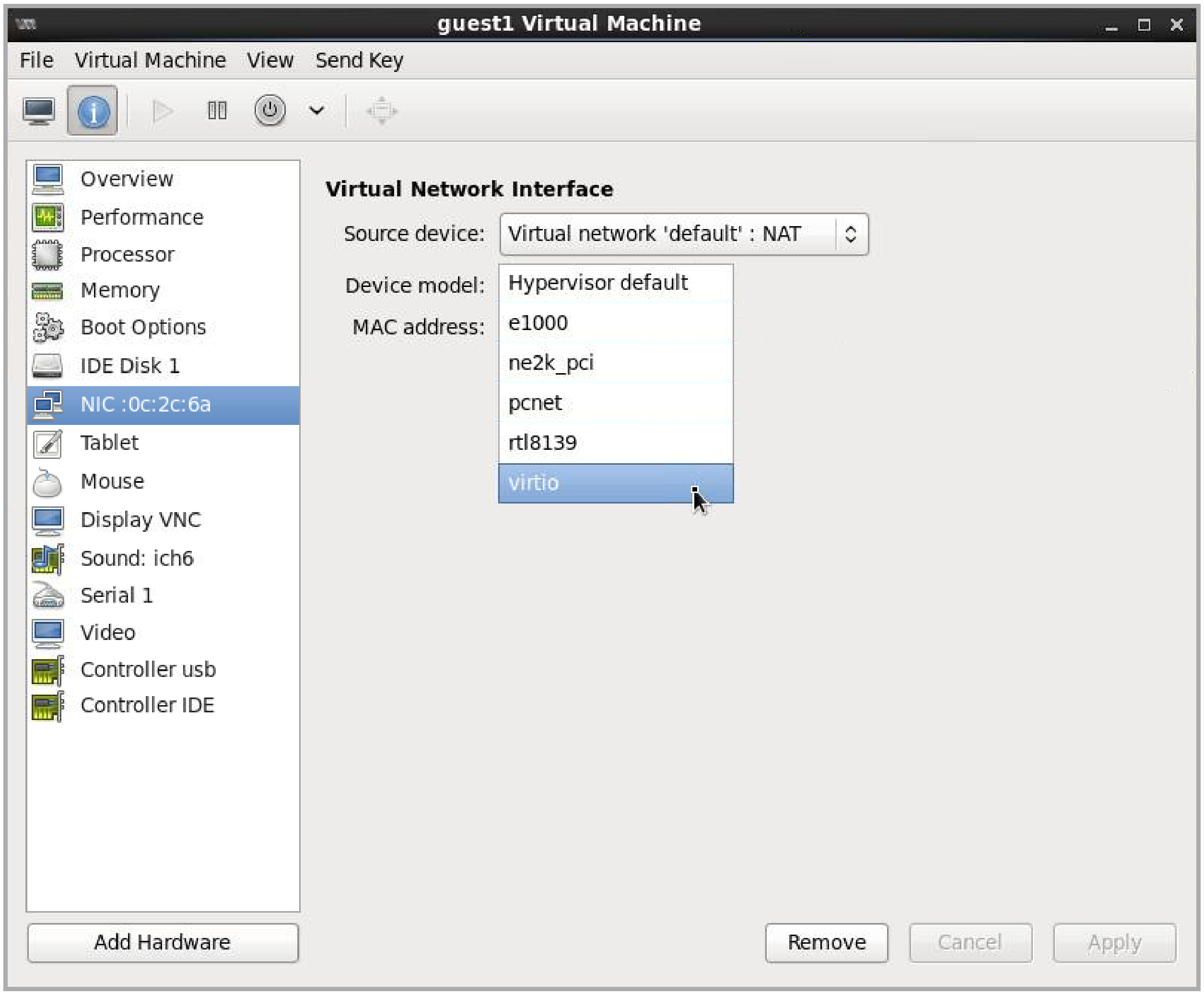
c. If you get the console view, click the information icon in the toolbar.



d. In the `guest1` Virtual Machine window, perform the following steps:

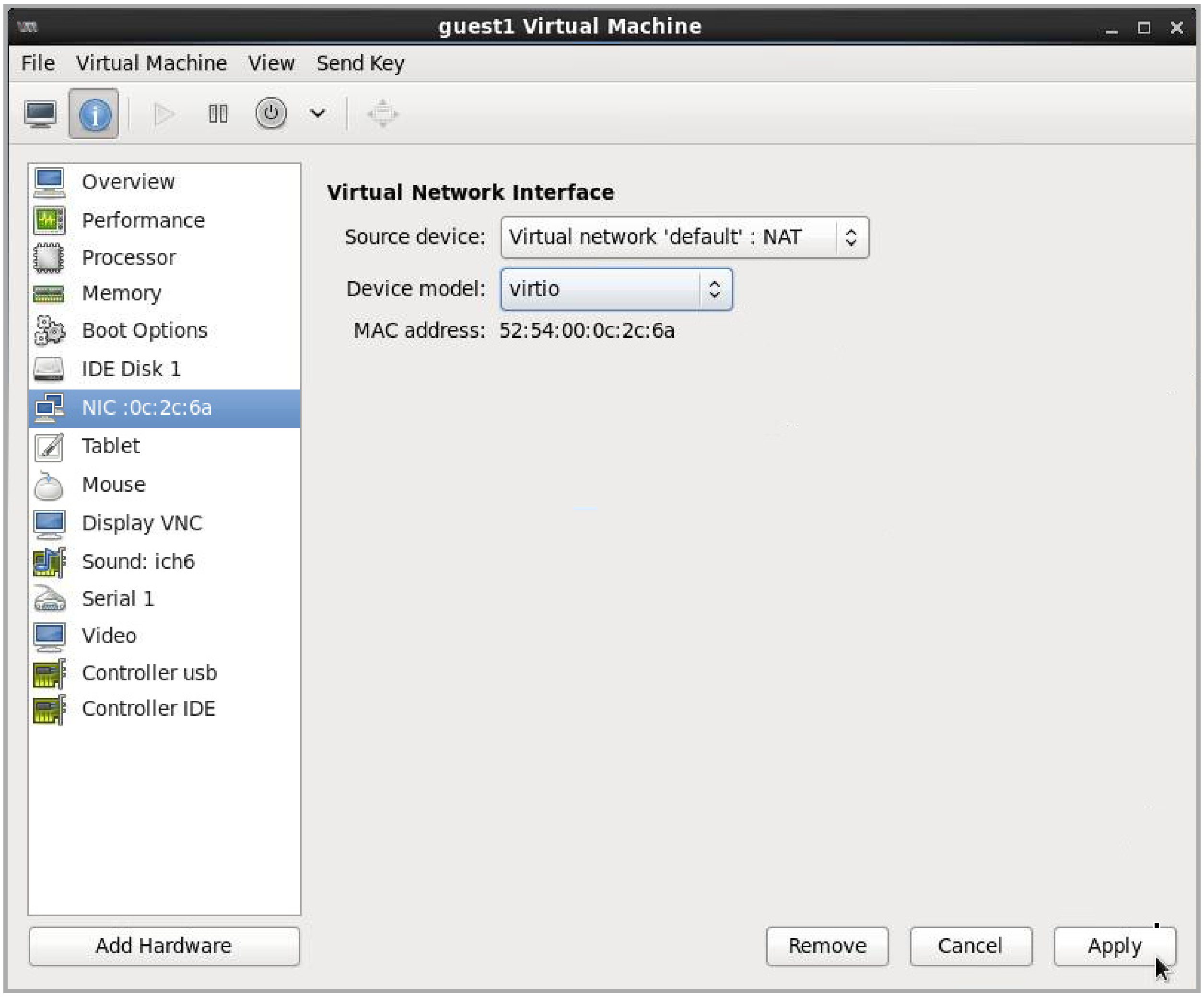
- Highlight the **NIC** in the left pane

- Click the spinbox for the Device Model
- Change the Device Model from Hypervisor default to **virtio**.

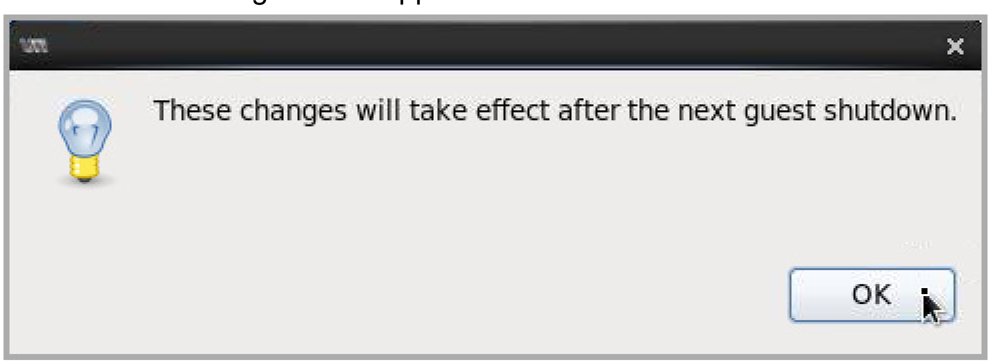


e. Click the **Apply** button to confirm your change.

- You might have to maximize the Virtual Machine window to see the Apply button.



- The following window appears. Click **OK**.



f. In your terminal window on **host05**, dump the virtual machine configuration file again with the `virsh dumpxml` command.

```
# virsh dumpxml guest1
<domain type='qemu' id='1'>
...
    <interface type='network'>
      <mac address='52:54:00:0c:2c:6a'/>
      <source network='default'/>
```

```
      <target dev='vnet0'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00'
    slot='0x03' function='0x0'/>
    </interface>
...
</domain>
```

- The change to the device model for the network interface has not yet taken effect.

g. Use the `cd` command to change to the default location for all virtual machine XM configuration files, `/etc/libvirt/qemu`.

- Use the `ls -l` command to list the contents of the directory.

```
# cd /etc/libvirt/qemu
# ls -l
-rw-------. ...  guest1-clone.xml
-rw-------. ...  guest1.xml
drwx------. ...  networks
```

h. Use the `cat` command to display the contents of the `guest1.xml` file.

```
# cat guest1.xml
...
    <interface type='network'>
      <mac address='52:54:00:0c:2c:6a'/>
      <source network='default'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00'
    slot='0x03' function='0x0'/>
    </interface>
...
</domain>
```

- Note that this file contains the change to the model type for the `guest1` network interface.
- You can also enable paravirtualization for the disk driver.

i. In your terminal window on **host05**, use the `virsh list` command to display active domains.

```
# virsh list
 Id    Name                           State
----------------------------------------------------
 1     guest1                         running
```

- The `virsh` command refers to virtual machines as domains. You can find more information about the life cycle of domains at this location: http://wiki.libvirt.org/page/VM_lifecycle.

j. Use the `virsh list --all` command to display all domains, regardless of their state.

```
# virsh list --all
 Id    Name                           State
----------------------------------------------------
 -     guest1                         running
 -     guest1-clone                   shut off
```

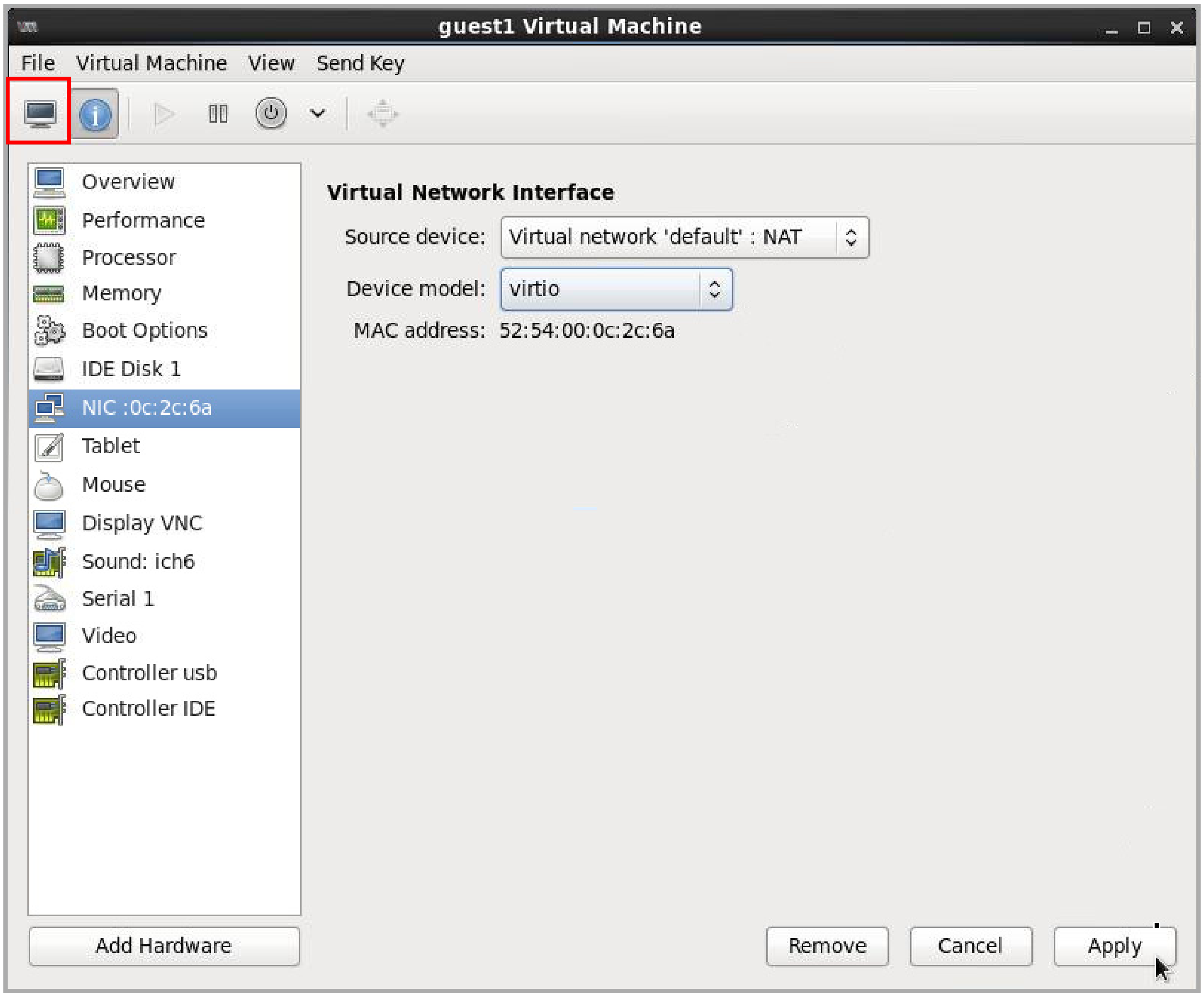
Note: As mentioned earlier, in your lab environment your virtual machine is not taking advantage of the acceleration provided by KVM. Without the KVM acceleration, the boot process is very slow.

You can read through the remaining tasks in this practice, or you can come back to this practice in an hour or so, at which time the boot process will have completed.

You need to eventually shut down **host05** so that you can start the required VMs for the remaining practices. The remaining practices require the following VMs:

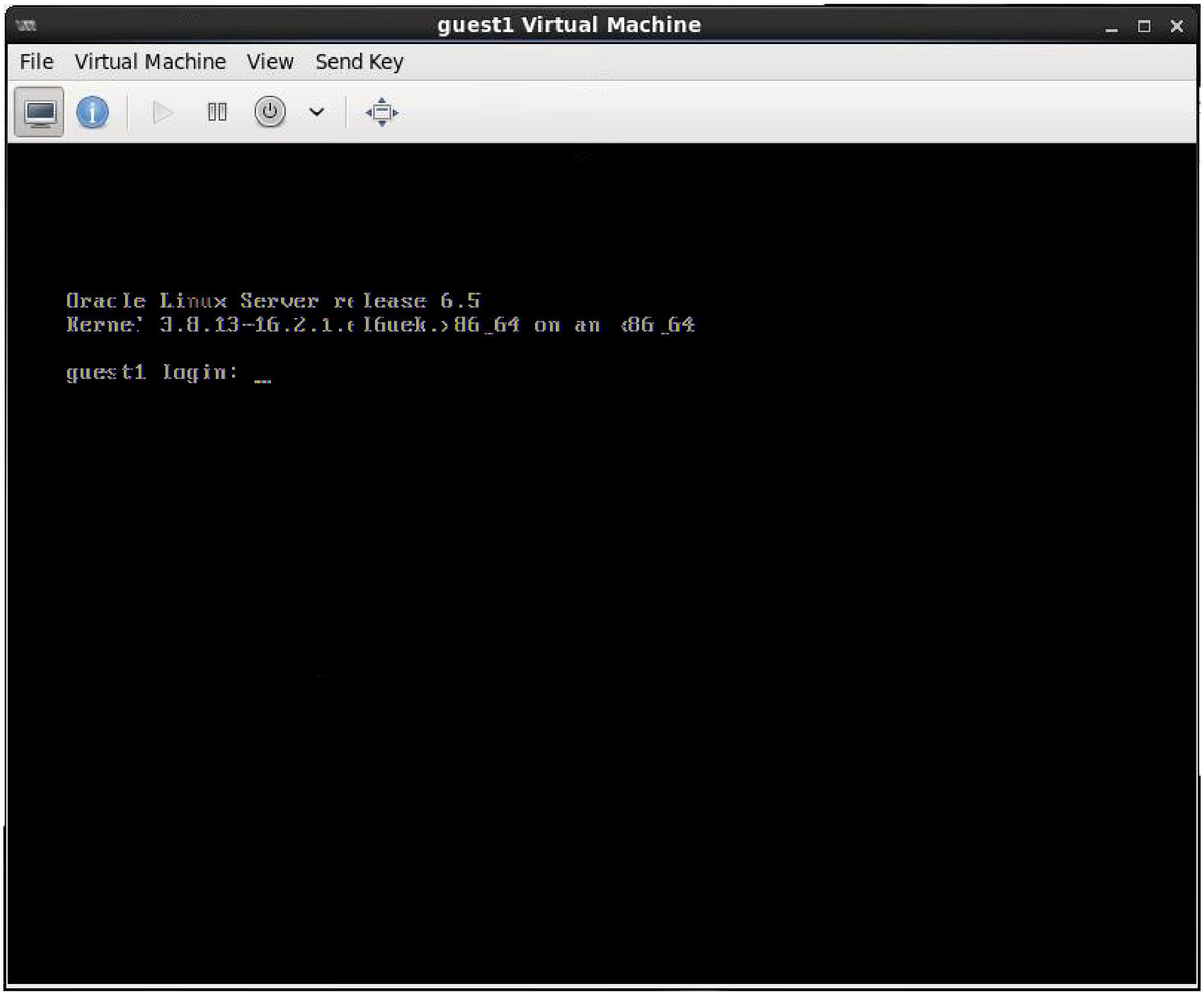
- *Practice 15 – Linux Container* uses **host02**
- *Practice 16 – SELinux* uses **host01** and **host03**
  - Practice 16 also uses **host05** again
- *Practice 17 – Core Dump Analysis* uses **host03**
- *Practice 18 – DTrace* uses **host03**

4. Log in to the **guest1** KVM virtual machine.
   a. Click the console tool from the Virtual Manager window toolbar.



b. The console window appears.

- Press the ENTER key to display the login prompt.



c. Log in as the `root` user.

- The root password on **guest1** is `0racle` (leading zero).

```
guest1 login: root
Password: 0racle
Last login: ...
[root@guest1 ~]#
```

d. Use the `df -h` command to list the mounted partitions.

```
# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/mapper/vg_guest1-lv_root ...       /
tmpfs                         ...
/dev/sda1                     ...       /boot
```

e. Use the `ifconfig` command to display the network interface configuration.

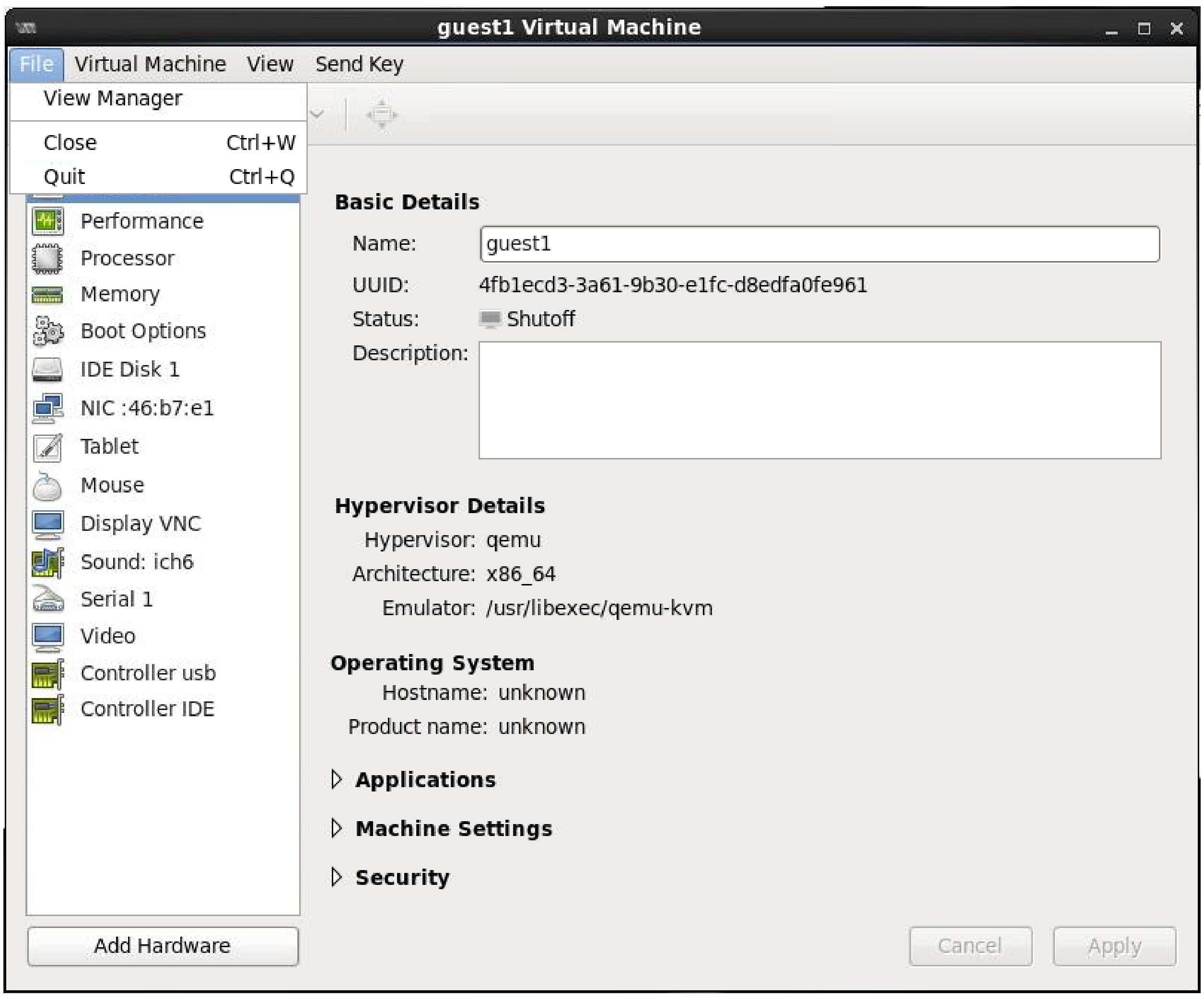
```
# ifconfig
eth0        Link encap:Ethernet
            inet addr:192.168.122.131
...
```

f. Run any other commands you like to explore the configuration.

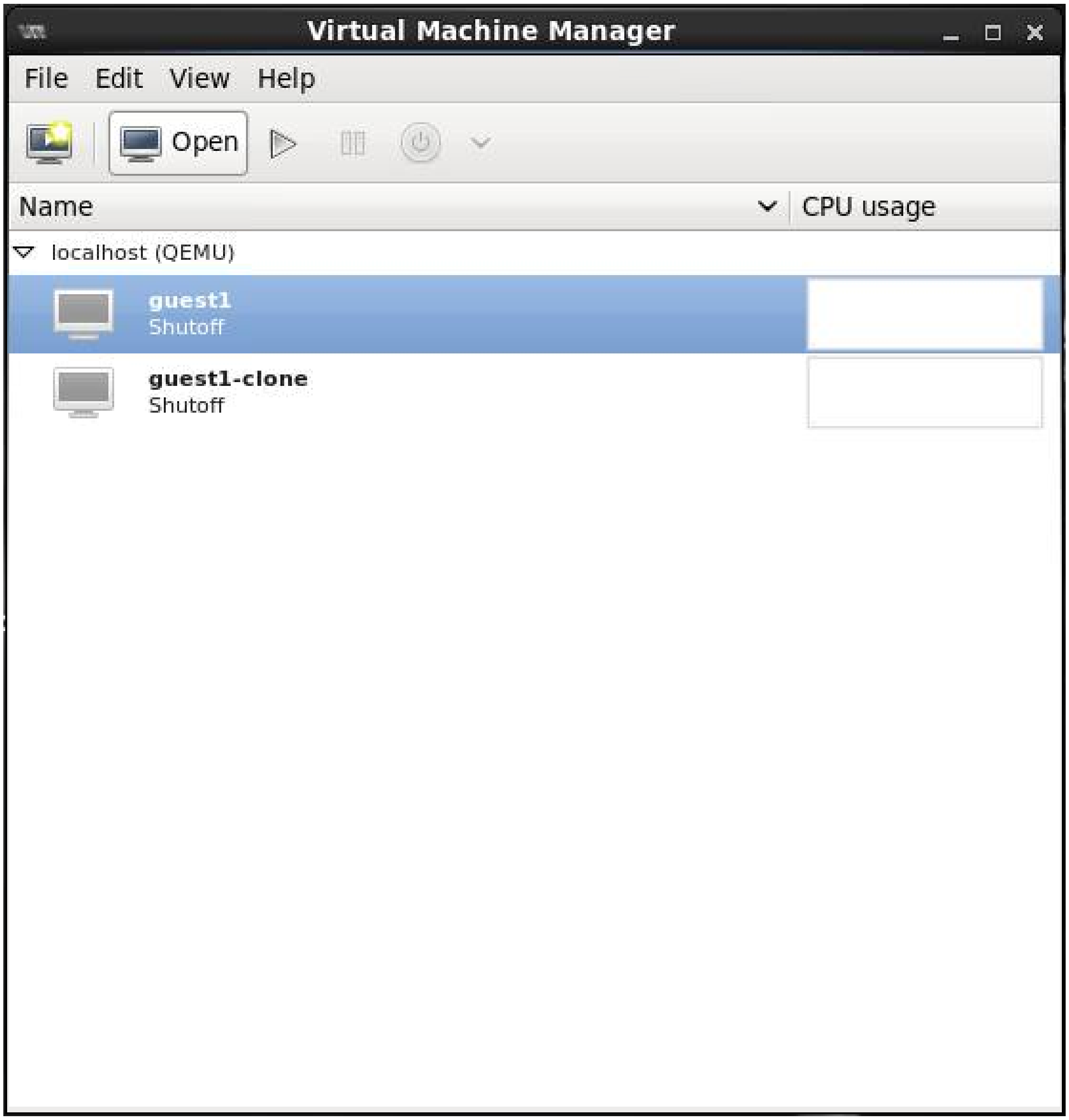
g. Use the `shutdown -h now` command to shut down **guest1**.

```
# shutdown -h now
The system is now going down.
...
```

h. Close the Virtual Machine window by selecting **File > Close** from the menu bar.



i. Notice that **guest1** is now Shutoff.



j. Close the Virtual Machine Manager by selecting **File > Quit**.

k. Run the `shutdown` command to shut down **host05**.

```
# shutdown -h  now
```

5. From **dom0**, start the **host01**, **host02**, and **host03** VMs.

   a. Use the `cd` command to change to the `/OVS/running_pool/host01` directory.

      - Use the `xm create vm.cfg` command to start the **host01** VM.

```
# cd /OVS/running_pool/host01
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host01 (id=...)
```

b. Use the `cd` command to change to the `/OVS/running_pool/host02` directory.

- Use the `xm create vm.cfg` command to start the **host02** VM.

```
# cd /OVS/running_pool/host02
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host02 (id=...)
```

c. Use the `cd` command to change to the `/OVS/running_pool/host03` directory.

- Use the `xm create vm.cfg` command to start the **host03** VM.

```
# cd /OVS/running_pool/host03
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host03 (id=...)
```

d. Run the `xm list` command to verify that all three VMs are running.

```
# xm list
Name              ID    Mem    VCPUs     State    Time(s)
Domain-0           0    2048      2      r-----       ...
host01           242    1536      1      -b----       ...
host02           243    1536      1      -b----       ...
host03           244    1536      1      -b----       ...
```

- In this example, **host01**, **host02**, and **host03**, in addition to **Domain-0**, are running as expected.

  The ID values and the State values might differ from your values.

Practices for Lesson 15: Linux Containers (LXC)

Chapter 15

Practices for Lesson 15: Linux Containers

Practices Overview

In these practices, you do the following:

- Prepare your system to create a Linux Container.
- Create a Linux Container from a template.
- Explore container configuration.
- Use `lxc` commands.
- Use an Oracle VM template as a base environment.
- Create a container from an existing rootfs.

Practice 15-1: Completing Linux Container Prerequisites

Overview

In this practice, you perform the following:

- Set SELinux to Permissive mode.
- Ensure that the cgroup package is installed and start the `cgconfig` service.
- Ensure that the Btrfs package is installed.
- Ensure that the Linux Container packages are installed.
- Ensure that the `libvirt` package is installed and start the `libvirtd` service.
- Create and mount a Btrfs subvolume.

Assumptions

- You are the `root` user on **dom0**.
- The **host02** VM is running.

Tasks

1. From **dom0**, use the `ssh` command to log in to **host02**.
    - The `root` password is `oracle`.

```
[dom0]# ssh host02
root@host02's password: oracle
[root@host02 ~]#
```

2. From **host02**, use the `setenforce` command to set SELinux to Permissive mode.

```
# setenforce 0
# getenforce
Permissive
```

3. Verify that the required packages are installed.
    a. Use the `rpm` command to check whether the `libcgroup` package is installed.

```
# rpm -qa | grep libcgroup
```

- In this example, the package is not installed.

    b. Run the `yum install` command to install the `libcgroup` package.

```
# yum install libcgroup
...
Transaction Summary
============================================================
Install       1 Package(s)
Total download size: 125 k
Installed size: 321 k
Is this ok [y/N]: y
...
Complete!
```

c. Use the `chkconfig` command to configure the `cgconfig` service to start at boot time.

```
# chkconfig cgconfig on
# chkconfig cgconfig --list
cgconfig  0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

- In this example, `cgconfig` is properly configured to start at run levels 2 through 5.

d. Use the `service` command to start the `cgconfig` service.

```
# service cgconfig start
Starting cgconfig service:                        [  OK  ]
```

- Starting this service reads the `/etc/cgconfig.conf` configuration file and populates the `/cgroup` hierarchy.

e. Use the `rpm -qa` command to check whether the `btrfs-progs` package is installed.

```
# rpm -qa |grep btrfs-progs
```

- In this example, the package is not installed.

f. Run the `yum install` command to install the `btrfs-progs` package.

```
# yum install btrfs-progs
...
Transaction Summary
=============================================================
Install       1 Package(s)
Total download size: 393 k
Installed size: 2.7 M
Is this ok [y/N]: y
...
Complete!
```

g. Use the `modprobe` command to load the `btrfs` kernel module. Use the `lsmod` command to ensure that the `btrfs` kernel module is loaded.

```
# modprobe btrfs
# lsmod | grep btrfs
btrfs           817000   0
zlib_deflate     21991   1   btrfs
libcrc32c         1252   1   btrfs
```

h. Use the `rpm -qa` command to ensure that the `lxc` and `lxc-libs` packages are installed.

```
# rpm -qa |grep lxc
```

- In this example, the packages are not installed.

i. Run the `yum install` command to install the `lxc` and the `lxc-libs` packages.

- There are a number of dependency packages that are also installed.

```
# yum install lxc lxc-libs
...
Transaction Summary
=============================================================
```

```
Install       20 Package(s)
Total download size: 9.8 M
Installed size: 29 M
Is this ok [y/N]: y
...
Complete!
```

j. Use the `rpm -qa` command to ensure that the `libvirt` package is installed.

```
# rpm –qa |grep libvirt
libvirt-0.10.2-29.0.1.el6.x86_64
libvirt-client-0.10.2-29.0.1.el6.x86_64
```

- In this example, the packages are installed.

k. Use the `chkconfig` command to configure the `libvirtd` service to start at boot time.

```
# chkconfig libvirtd on
# chkconfig libvirtd --list
libvirtd  0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

- In this example, `libvirtd` is properly configured to start at run levels 2 through 5.

l. Use the `service` command to start the `libvirtd` service.

```
# service libvirtd start
Starting libvirtd daemon:
```

4. Run the `lxc-checkconfig` command to display information about your kernel configuration.

```
# lxc-checkconfig
Kernel configuration not found at /proc/config.gz; search...
Kernel configuration found at /boot/config-3.8.13-16.2.1...
--- Namespaces ---
Namespaces: enabled
Utsname namespaces: enabled
Ipc namespaces: enabled
Pid namespaces: enabled
User namespaces: missing
Network namespaces: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled
```

```
--- Misc  ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
...
```

- All namespaces, control groups, and miscellaneous kernel functionalities are enabled with the exception of user namespaces. Support for user namespaces is included with kernel version `3.13`.

5. Set up the Btrfs file system for the Linux Containers.
   a. Use the `mkfs.btrfs` command to create a Btrfs file system on `/dev/xvdd`.
      - Ensure that you create the file system on `/dev/xvdd`, not `/dev/xvdb`.

```
# mkfs.btrfs –f –L container /dev/xvdd
...
```

   b. Use the `vi` editor to add the following entry in `/etc/fstab` to mount `/dev/xvdd` on `/container`.

```
# vi /etc/fstab
...
/dev/xvdd  /container  btrfs  defaults  0  0
```

   c. Use the `mkdir` command to make `/container`.

```
# mkdir /container
```

   d. Use the `mount -a` command to mount file systems in `/etc/fstab`.

```
# mount –a
```

Practice 15-2: Creating an Oracle Linux Container

Overview

In this practice, you perform the following:

- Create an Oracle Linux Container using the `lxc-oracle` template.
- Explore the container configuration.

Assumptions

- You are the `root` user on **host02**.

Tasks

1. Create an Oracle Linux Container.
   a. Use the `lxc-create` command to display options supported by the `lxc-oracle` template.

```
# lxc-create –t oracle –h
...
Template-specific options (TEMPLATE_OPTIONS):
-a|--arch=<arch>          architecture (ie. i686, x86_64)
-R|--release=<release>    release to download for the new
container
-u|--url=<url>            replace yum repo url (ie. local yum
mirror)
-t|--templatefs=<path>    copy/clone rootfs at path instead of
downloading
-h|--help
...
```

   - Notice that you can use the `-u` option and reference a local Yum mirror, which is what you do in step 1b.

   b. Use the `lxc-create` command to create an Oracle Linux Container named `ol-test` from the `lxc-oracle` template. Use the `-u http://192.0.2.1/repo` template option to get packages from the local Yum mirror on **dom0** (`192.0.2.1`).

   - This command takes a few minutes to complete.

```
# lxc-create –n ol-test –t oracle -- -u http://192.0.2.1/repo
...
Host is OracleServer 6.5
Create configuration file /container/ol-test/config
Downloading release 6.5 for x86_64
...
Complete!
Rebuilding rpm database
Configuring container for Oracle Linux 6.5
Added container user:oracle password:oracle
Added container user:root password:root
Container : /container/ol-test/rootfs
```

```
Config     : /container/ol-test/config
Network    : eth0 () on virbr0
'oracle' template installed
'ol-test' created
```

- Notice that the `lxc-oracle` template creates an `oracle` user with the password `oracle`.
- The `lxc-oracle` template sets the `root` password to `root`.
- The `lxc-oracle` template uses the `virbr0` bridge set up by `libvirtd`.

2. Explore the container configuration.
   a. Use the `ls` command to display the contents of `/container`.

```
# ls /container
ol-test
```

- Notice that the system object directory with the name of the container, `ol-test`, exists in `/container`.

   b. Use the `cd` command to change to the `/container/ol-test` directory, and then use the `ls` command to display the contents of the directory.

```
# cd /container/ol-test
# ls
config  fstab  rootfs
```

- The `config` file contains the configuration settings for the container, which were obtained from the template script.
- The file system table, `fstab`, was also configured from the template.

   c. Use the `less` command to view the `config` file.

```
# less config
# Template used to create this container: oracle
# Parameters passed to the template: -u http://192.0.2.1/repo
# Template script checksum (SHA-1): ...

lxc.network.type = veth
lxc.network.link = virbr0
lxc.network.flags = up

# Container configuration for Oracle Linux 6.5
lxc.arch = x86_64
lxc.utsname = ol-test
lxc.devttydir = lxc
lxc.tty = 4
lxc.pts = 1024
lxc.rootfs = /container/ol-test/rootfs
lxc.mount  = /container/ol-test/fstab
...
lxc.cap.drop = mac_admin mac_override setfcap setpcap
```

```
lxc.cap.drop = sys_module sys_nice sys_pacct
lxc.cap.drop = sys_rawio sys_time
lxc.cap.drop = sys_resource
# Networking
lxc.network.name = eth0
lxc.network.mtu = 1500
lxc.network.hwaddr = ...
# Control Group devices: all denied except those whitelisted
lxc.cgroup.devices.deny = a
lxc.cgroup.devices.allow = c 1:3 rwm  # /dev/null
lxc.cgroup.devices.allow = c 1:5 rwm  # /dev/zero
lxc.cgroup.devices.allow = c 1:7 rwm  # /dev/full
lxc.cgroup.devices.allow = c 5:0 rwm  # /dev/tty
lxc.cgroup.devices.allow = c 1:8 rwm  # /dev/random
lxc.cgroup.devices.allow = c 1:9 rwm  # /dev/urandom
lxc.cgroup.devices.allow = c 136:* rwm  # /dev/tty[1-4] ptys ...
lxc.cgroup.devices.allow = c 5:2 rwm  # /dev/ptmx pty master
```

- The `lxc.arch` key specifies the architecture for the container. Without any template options, the architecture is the same as the host system, `x86_64`.
- The `lxc.utsname` key specifies the host name for the container.
- The `lxc.rootfs` key specifies the root file system, `/container/ol-test/rootfs`, for the container. When not specified, the container shares its root file system with the host system.
- The `lxc.mount` key specifies the file system table, `/container/ol-test/fstab`, which contains the file system mount information.
- Containers can have their own network interface (the `lxc-oracle` template defaults to `eth bridge`), or share with the host system (`lxc.network.type` is not configured; this is not recommended), or have only loopback (`lxc.network.type = empty`, which is the most secure).
- The `lxc.network.type` key is set to `veth`, which is a peer network device with one side assigned to the container and the other side attached to a bridge specified by `lxc.network.link`.
- Notice all the `lxc.cgroup.devices` keys, which deny cgroup tasks access to all devices except those whitelisted.

d. Use the `cat` command to view the `fstab` file.

```
# cat fstab
proc   /container/ol-test/rootfs/proc    proc   nodev,...
devpts /container/ol-test/rootfs/dev/pts devpts defaults 0 0
sysfs  /container/ol-test/rootfs/sys     sysfs  defaults 0 0
```

e. Use the `ls` command to view the contents of the `rootfs` directory.

```
# ls rootfs
bin   dev  home  lib64  mnt  proc  sbin     srv  tmp  var
boot  etc  lib   media  opt  root  selinux  sys  usr
```

f. Use the `btrfs subvolume list` command to view the subvolumes in the `/container` Btrfs file system.

```
# btrfs sub list /container
ID 256 gen 10 top level 5 path ol-test/rootfs
```

- Notice that the `rootfs` directory is a Btrfs subvolume when `/container` is Btrfs.

Practice 15-3: Using `lxc` Commands

Overview

In this practice, you perform the following:

- Use the `lxc-start` command to start the container.
- Log in to the container and explore the configuration.
- Use the `lxc-stop` command to stop the container.
- Use the `lxc-info` command to display the state of the container.
- Use the `lxc-start` command to start the container as a daemon.
- Use the `lxc-console` command to launch a console for the container.
- Use the `lxc-freeze` and `lxc-unfreeze` commands to freeze and unfreeze the container's processes.
- Use the `lxc-cgroup` command to view and modify container cgroup subsystem parameters.

Assumptions

- You are the `root` user on **host02**.

Tasks

1. Verify that a bridge is set up on the host.
   a. Use the `ifconfig –a` command to view the network interface configuration.
      - Sample output is shown. Output on your system might be different.

```
# ifconfig –a
...
virbr0      Link encap:Ethernet  HWaddr 52:54:00:94:63:0D
            inet addr: 192.168.122.1  ...
...
virbr0-nic  Link encap:Ethernet  HWaddr 52:54:00:94:63:0D
            BROADCAST MULTICAST  MTU: 1500  Metric:1
...
```

   b. If you do not see `virbr0` configuration information, use the `service` command to restart the `libvirtd` service.

```
# service libvirtd restart
Stopping libvirtd daemon:                [ OK ]
Starting libvirtd daemon:                [ OK ]
```

      - Repeat step 1a to ensure that `virbr0` configuration information exists.
2. Start a Linux Container.
   a. Use the `lxc-start` command to start the `ol-test` container.
      - Log in as the `root` user. Password is `root`.

```
# lxc-start –n ol-test
...
Oracle Linux Server release 6.5
```

```
Kernel 3.8.13-16.2.1.el6uek.x86_64 on an x86_64
ol-test login: root
Password: root
[root@ol-test ~]#
```

- When your start an Oracle Linux container without the daemon option, `-d`, you get a login prompt in your current terminal.
- This login prompt is the "console." The `lxc-oracle` template, by default, also sets up four additional terminals that you can log in on with `lxc-console`.

3. Explore the Linux Container.
   a. Use the `ps` command to view the processes within the container.

```
# ps -ef
UID      PID   PPID  ...  CMD
root       1      0  ...  /sbin/init
root     194      1  ...  /sbin/dhclient -H ol-test -1 -q
root     217      1  ...  /sbin/rsyslogd -i /var/run/syslog...
root     266      1  ...  login -- root
root     270      1  ...  /sbin/mingetty /dev/tty1
...
```

- When you use the `lxc-start` command to start a container, by default, the copy of `/sbin/init` in the container is started to spawn other processes in the container's process space.
- Any device access or system calls are handled by the kernel running on the host.
- Notice that the `/sbin/init` process within the container has a PID of `1`.
- Notice that four terminals are available.

   b. Use the `mount` command to display the mounted file systems within the container.

```
# mount
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

   c. Use the `ifconfig` command to display the network interfaces for the container.

- Sample output is shown; yours might be different.

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr ...
          inet addr:192.168.122.20  Bcast: 192.168.122.255...
          inet6 addr: ... Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          ...

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          ...
```

d. Use the `cat` command to view the `eth0` configuration file, `/etc/sysconfig/network-scripts/ifcfg-eth0`.

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
HOSTNAME=ol-test
NM_CONTROLLED=no
TYPE=Ethernet
```

- Notice that the default configuration uses DHCP.

e. Use the `cat` command to view the SELinux configuration file, `/etc/selinux/config`.

```
# cat /etc/selinux/config
SELINUX=disabled
```

- Notice that SELinux is disabled because some things in containers do not yet work with SELinux enabled.

4. Log out of the Linux Container.

a. Use the `exit` command to log off the container.

```
# exit
Oracle Linux Server release 6.5
Kernel 3.8.13-16.2.1.el6uek.x86_64 on an x86_64
ol-test login:
```

- Notice that you are still presented with a login prompt.
- The only way to get back this terminal is by stopping the container, either by issuing a `shutdown` command while logged in, or by running `lxc-stop` from another terminal.

b. Open a new terminal window from **dom0**. Become the `root` user by running the `su -` command. The `root` password on **dom0** is `oracle`.

```
[dom0]$ su -
Password: oracle
```

c. From **dom0**, use the `ssh` command to connect to **host02**. The `root` password on **host02** is `oracle`.

```
[dom0]# ssh host02
root@host02's password: oracle
```

d. Use the `lxc-info` command to display the state of the `ol-test` container.

```
# lxc-info –n ol-test
state:     RUNNING
pid:        26939
```

- Notice that the container is `RUNNING`.
- The PID in this example is `26939`; your PID might be different.

e. Run the `lxc-stop` command to get back to **host02** on the other terminal.

```
# lxc-stop -n ol-test
```

- Notice that the other terminal window now has a **host02** prompt.

f. Use the `lxc-info` command to display the state of the `ol-test` container.

```
# lxc-info -n ol-test
state:     STOPPED
pid:         -1
```

- Notice that the container is stopped.

5. Start a Linux Container as a daemon.

a. Use the `lxc-start` command with the `-d` option to start the `ol-test` container as a daemon.

```
# lxc-start -n ol-test -d
```

- Notice that you are not presented with a login prompt.

b. Use the `lxc-info` command to display the state of the `ol-test` container.

```
# lxc-info -n ol-test
state:     RUNNING
pid:        27485
```

- In this example, the PID is `27485`. Yours might be different.

c. Use the `ps` command to view the processes pertaining to the running container. Pipe the output to `grep` and search for the PID from the previous step.

```
# ps -ef | grep 27485
root  27485   27478  ...  /sbin/init
root  27740   27485  ...  /sbin/dhclient -H ol-test -1 -q ...
root  27763   27485  ...  /sbin/rsyslogd -i /var/run/syslog...
...
root  27797   27485  ...  /sbin/mingetty /dev/console
root  27801   27485  ...  /sbin/mingetty /dev/tty1
...
```

- Notice that a number of `/sbin/mingetty` processes are running.
- In this example, the parent (PPID) of the `/sbin/init` process (PID `27485`) is `27478`. Yours might be different.

d. Use the `ps` command to view the command associated with the PPID.

```
# ps -ef | grep 27478
root  27478       1  ...  lxc-start -n ol-test -d
root  27485   27478  ...  /sbin/init
...
```

- Notice that the `lxc-start` command is associated with this PID.

6. Log in to and out of the Linux Container.
   a. Use the `lxc-console` command to launch a console for the container.
      - Log in as the `root` user. Password is `root`.

```
# lxc-console -n ol-test
Oracle Linux Server release 6.5
Kernel 3.8.13-16.2.1.el6uek.x86_64 on an x86_64
ol-test login: root
Password: root
[root@ol-test ~]#
```

      - Use of `lxc-console` requires that the container be running a `/sbin/mingetty` process.
      - You can also use `ssh` to log in to the container.
      - To log in using `ssh`, SELinux must be disabled on the host system and the container must be running the SSH daemon.
   b. Press Ctrl + A followed by Q to exit the `lxc-console` session.
      - Do not hold all three keys down at once.
      - Press Ctrl + A, then release these keys, and then press Q to exit.

```
[root@ol-test ~]# hostname
ol-test
# CTRL-a q
[root@host02 ~]# hostname
host02.example.com
```

      - Ctrl + A, Q is a feature of the `lxc-console` command.
      - Ctrl + A, Q works only when you log in using `lxc-console`.
      - Notice the host name before and after exiting the `lxc-console` session.
7. Freeze and unfreeze a Linux Container.
   a. Use the `lxc-info` command to display the state of the `ol-test` container.

```
# lxc-info -n ol-test
state:     RUNNING
pid:        27485
```

      - Notice that the state of the container is `RUNNING`.
   b. Use the `lxc-freeze` command to freeze all of the container's processes.

```
# lxc-freeze -n ol-test
```

   c. Use the `lxc-info` command to display the state of the `ol-test` container.

```
# lxc-info -n ol-test
state:     FROZEN
pid:        27485
```

      - Notice that the state of the container is `FROZEN`.

d. Use the `lxc-console` command to launch a console for the container.

```
# lxc-console –n ol-test

Type <Ctrl+a q> to exit the console, ...
```

- You cannot log in while a container is in a `FROZEN` state.
- Press Ctrl + A, then release these keys, and then press Q to exit.

e. Use the `lxc-unfreeze` command to unfreeze all of the container's processes.

```
# lxc-unfreeze –n ol-test
```

f. Use the `lxc-info` command to display the state of the `ol-test` container.

```
# lxc-info –n ol-test
state:     RUNNING
pid:        27485
```

- Notice that the state of the container is `RUNNING`.
- Now you can successfully log in using the `lxc-console` command.

8. View and modify the container resources.

a. Use the `lxc-cgroup` command to view the devices (`devices.list`) allowed to be used by the container.

```
# lxc-cgroup –n ol-test devices.list
c 1:3 rwm
c 1:5 rwm
c 1:7 rwm
c 5:0 rwm
c 1:8 rwm
c 1:9 rwm
c 136:* rwm
c 5:2 rwm
```

b. Use the `grep` command to display the lines in the `/container/ol-test/config` file that contain the string "`devices`."

```
# grep devices /container/ol-test/config
lxc.cgroup.devices.deny = a
lxc.cgroup.devices.allow = c 1:3 rwm # /dev/null
lxc.cgroup.devices.allow = c 1:5 rwm # /dev/zero
lxc.cgroup.devices.allow = c 1:7 rwm # /dev/full
lxc.cgroup.devices.allow = c 5:0 rwm # /dev/tty
lxc.cgroup.devices.allow = c 1:8 rwm # /dev/random
lxc.cgroup.devices.allow = c 1:9 rwm # /dev/urandom
lxc.cgroup.devices.allow = c 136:* rwm    # /dev/tty[1-4]...
lxc.cgroup.devices.allow = c 5:2 rwm # /dev/ptmx ...
```

- Notice that these entries correspond with the output of the `lxc-cgroup` command.

c. Use the `lxc-cgroup` command to view the relative share of CPU time available (`cpu.shares`) to the tasks in the container.

```
# lxc-cgroup -n ol-test cpu.shares
1024
```

d. Use the `cat` command to view the `cpu.shares` subsystem parameter for both the host and the container.

```
# cat /cgroup/cpu/cpu.shares                 (for the host)
1024
# cat /cgroup/cpu/lxc/ol-test/cpu.shares  (for the container)
1024
```

e. Use the `lxc-cgroup` command to change the `cpu.shares` to 750 for the `ol-test` container.

```
# lxc-cgroup -n ol-test cpu.shares 750
```

f. Use the `cat` command to view the `cpu.shares` subsystem parameter for both the host and the container.

```
# cat /cgroup/cpu/cpu.shares                 (for the host)
1024
# cat /cgroup/cpu/lxc/ol-test/cpu.shares  (for the container)
750
```

g. Use the `vi` editor to add an entry to the `/container/ol-test/config` file, which permanently sets `cpu.shares` to `500` for the `ol-test` container.

```
# vi /container/ol-test/config
...
lxc.cgroup.cpu.shares = 500
```

- You need to use `lxc-stop` and `lxc-start` on the container to re-read the container's configuration file.

h. Use the `lxc-stop` command to stop the `ol-test` container.

```
# lxc-stop -n ol-test
```

i. Use the `lxc-start` command with the `-d` option to start the `ol-test` container as a daemon.

```
# lxc-start -n ol-test -d
```

j. Use the `cat` command to view the `cpu.shares` subsystem parameter for the container.

```
# cat /cgroup/cpu/lxc/ol-test/cpu.shares
500
```

- Notice that the `cpu.shares` setting is now persistent.

Practice 15-4: Creating a Container from an Oracle VM Template

Overview

In this practice, you perform the following:

- Use an Oracle VM template as a base environment.
- Create a container from an existing rootfs.

Assumptions

- You are the `root` user on **host02**.

Tasks

1. Create a Btrfs subvolume.
   a. Use the `btrfs sub create` command to create a subvolume named `ol6-template` in `/container`.

```
# btrfs sub create /container/ol6-template
Create subvolume '/container/ol6-template'
```

2. Set up a VM template as the system image.

* Perform steps a–d from **dom0**.

   a. As the `root` user on **dom0**, use the `sftp` command to connect to **host02**.
      - The `root` password is `oracle`.

```
[dom0]# sftp host02
Connecting to host02...
root@host02's password: oracle
sftp>
```

   b. Use the `lcd` command to change to the `/OVS/seed_pool` directory on **dom0**.

```
sftp> lcd /OVS/seed_pool
```

   c. Use the `put` command to copy `V42906-01.zip` from **dom0** to **host02**.

```
sftp> put V42906-01.zip
Uploading V42906-01.zip to /root/V42906-01.zip
V42906-01.zip ...
```

   d. Use the `exit` command to close the `sftp` connection.

```
sftp> exit
```

* Perform the remaining steps from **host02**.

   e. From **host02**, use the `mkdir` command to create the `/work` directory.

```
[host02]# mkdir /work
```

   f. Use the `cd` command to change to the `/work` directory.

```
# cd /work
```

g. Use the `mv` command to move `V42906-01.zip` from `/root` to the current directory.

```
# mv /root/V42906-01.zip .
```

h. Use the `unzip` command to unzip the Oracle Linux 6 Update 5 ZIP archive (file name is `V42906-01.zip`).

```
# unzip V42906-01.zip
Archive: V42906-01.zip
  inflating: OVM_OL6U5_x86_64_PVM.ova
```

i. Use the `tar` command to extract the `OVM_OL6U5_x86_64_PVM.ova` file.

```
# tar xvf OVM_OL6U5_x86_64_PVM.ova
OVM_OL6U5_x86_64_PVM.ovf
OVM_OL6U5_x86_64_PVM.mf
System.img
```

j. Use the `rm` command to remove the following files:

```
# rm V42906-01.zip
rm: remove regular file 'V42906-01.zip'? y
# rm O*
rm: remove regular file 'OVM_OL6U5_x86_64_PVM.mf? y
rm: remove regular file 'OVM_OL6U5_x86_64_PVM.ova? y
rm: remove regular file 'OVM_OL6U5_x86_64_PVM.ovf? y
```

k. Use the `mv` command to rename the Oracle VM template System image, `System.img`, to `System.img.gz`.

```
# mv System.img System.img.gz
```

l. Use the `gunzip -d` command to uncompress the `System.img.gz` file.

- Note that this command takes a few minutes to complete.

```
# gunzip -d System.img.gz
```

m. Use the `ls` command to list the contents of `/dev/mapper`.

```
# ls /dev/mapper
control  vg_host02-lv_root  vg_host02-lv_swap
```

n. Use the `kpartx -l` command to list the partitions found on the `System.img` file.

```
# kpartx -l System.img
loop0p1 : 0 1028096 /dev/loop0 2048
loop0p2 : 0 19941376 /dev/loop0 210944
loop0p3 : 0 4194304 /dev/loop0 20971520
```

- The largest of these partitions, `loop0p2`, corresponds to the `root` file system.

o. Use the `kpartx -a` command to add device mappings for the detected partitions on the `System.img` file.

```
# kpartx -a System.img
```

p. Use the `ls` command to list the contents of `/dev/mapper`.

```
# ls /dev/mapper
control  loop0p1  loop0p2  loop0p3  vg_host02-lv_root
vg_host02-lv_swap
```

- Notice the new “loop” device mappings.

q. Use the `mount` command to mount the `root` file system, `/dev/mapper/loop0p2`, on `/mnt`.

```
# mount /dev/mapper/loop0p2 /mnt
```

r. Use the `cd` command to change to the `/mnt` directory, and then use the `ls` command to display the contents of `/mnt`.

```
# cd /mnt
# ls
bin  dev home lib64      media opt  root selinux sys u01 var
boot etc lib  lost+found mnt   proc sbin srv     tmp usr
```

s. Copy the contents of the image mounted on `/mnt` to the `/container/ol6-template` Btrfs subvolume.

```
# find . -mount -depth | cpio -pdv /container/ol6-template
...
```

- This “`find ... | cpio ...`” command is just one of several ways to recursively copy the contents of `/mnt` to the Btrfs subvolume. Use any other method if desired.

t. Use the `ls` command to display the contents of `/container/ol6-template`.

```
# ls /container/ol6-template
bin  dev home lib64      media opt  root selinux sys u01 var
boot etc lib  lost+found mnt   proc sbin srv     tmp usr
```

- Notice that the contents are identical to `/mnt`.

u. Use the `cd` command to change to the `/work` directory, and then use the `umount` command to unmount `/mnt`.

```
# cd /work
# umount /mnt
```

v. Use the `kpartx -d` command to delete the device mappings for the detected partitions on the `System.img` file.

```
# kpartx -d System.img
loop deleted : /dev/loop0
```

w. Use the `ls` command to list the contents of `/dev/mapper`.

```
# ls /dev/mapper
control  vg_host02-lv_root  vg_host02-lv_swap
```

- Notice that the “loop” device mappings are gone.

x. Use the `rm` command to remove the `System.img` file from the `/work` directory.

```
# rm /work/System.img
rm: remove regular file ‘/work/System.img’? y
```

3. Create a container from an existing rootfs.
   a. Use the `lxc-create` command to create an Oracle Linux Container as follows:
      - Name the container `ol65-64`.
      - Use the `lxc-oracle` template.
      - Clone the `/container/ol6-template` rootfs.

```
# lxc-create –n ol65-64 –t oracle -- -t /container/ol6-template
lxc-create: No config file specified, using the default config
/etc/lxc/default.conf
Host is OracleServer 6.5
Create configuration file /container/ol65-64/config
Delete subvolume '/container/ol65-64/rootfs'
Create a snapshot of '/container/ol6-template/' in
'/container/ol65-64/rootfs'
Configuring container for Oracle Linux 6.5
useradd: user 'oracle' already exists
Added container user:oracle password:oracle
Added container user:root password:root
Container : /container/ol65-64/rootfs
Config    : /container/ol65-64/config
Network   : eth0 () on virbr0
'oracle' template installed
'ol65-64' created
```

      - Notice that the packages were not downloaded from the Public Yum Server.
      - Notice that a Btrfs snapshot was taken of the `/container/ol6-template` subvolume.
   b. Use the `btrfs subvolume list` command to view the subvolumes in the `/container` Btrfs file system.

```
# btrfs sub list /container
ID 256 gen 45 top level 5 path ol-test/rootfs
ID 259 gen 49 top level 5 path ol6-template
ID 262 gen 51 top level 5 path ol65-64/rootfs
```

      - Notice that the `ol65-64/rootfs` Btrfs snapshot exists.
   c. Use the `lxc-ls` command to list the containers on the system.

```
# lxc-ls
ol65-64  ol-test
```

      - Notice the new `ol65-64` container.

d. Use the `lxc-start` command to start the `ol65-64` container.

```
# lxc-start -n ol65-64
...
Oracle Linux Server release 6.5
Kernel 3.8.13-16.2.1.el6uek.x86_64 on an x86_64
ol65-64 login: root
Password: root
[root@ol65-64 ~]#
```

e. Use the `shutdown` command to shut down the `ol65-64` container.

```
[root@ol65-64 ~]# shutdown -h now
...
Halting system...
[root@host02 ~]#
```

- Notice that the container is halted and you are returned to **host02**.

f. Use the `lxc-destroy` command to destroy the `ol65-64` container.

```
# lxc-destroy -n ol65-64
Delete subvolume '/container/ol65-64/rootfs'
```

g. Use the `lxc-ls` command to list the containers on the system.

```
# lxc-ls
ol-test
```

- Notice that the `ol65-64` container no longer exists.

h. Use the `btrfs subvolume list` command to view the subvolumes in the `/container` Btrfs file system.

```
# btrfs sub list /container
ID 256 gen 52 top level 5 path ol-test/rootfs
ID 259 gen 49 top level 5 path ol6-template
```

- Notice that the `ol65-64/rootfs` Btrfs snapshot no longer exists.

4. Log off **host02**.

a. Use the `exit` command to log off **host02**.

- You are returned to **dom0**.

```
# exit
logout
Connection to host02 closed.
```

Practices for Lesson 16: Security Enhanced Linux (SELinux)

Chapter 16

Practices for Lesson 16: Security Enhanced Linux (SELinux)

Practices Overview

In these practices, you:

- Perform SELinux configuration from the GUI and the command line.
- Change the value of an SELinux Boolean to allow `ftp` to read and write files in user home directories.
- Change the SELinux context file type to enable a file system to be used for storing KVM virtual machine disk images when SELinux is in enforcing mode.

Practice 16-1: Exploring SELinux

Overview

In this practice, you explore SELinux files and directories, execute SELinux commands, install the SELinux Administration GUI package, and explore the GUI.

Assumptions

- You are the `root` user on **dom0**.

Tasks

1. From **dom0**, connect to the **host03** guest by using `vncviewer`.
   a. Determine the VNC port number for **host03** by running the `xm list -l host03 | grep location` command.

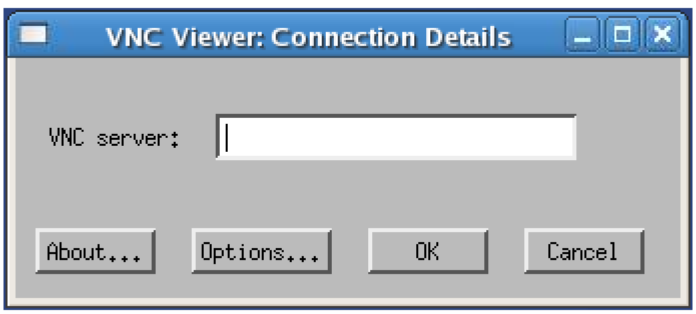
```
# xm list -l host03 | grep location
                (location 0.0.0.0:5903)
                (location 2)
```

   - The sample shown indicates that the port number is `5903`. This might not be true in your case.

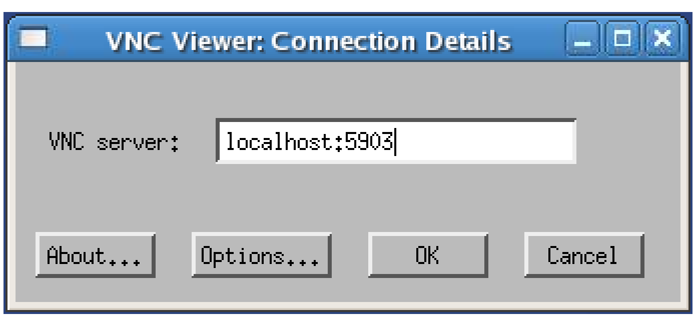
   b. Run the `vncviewer&` command:

```
# vncviewer&
```

   - The **VNC Viewer: Connection Details** dialog box is displayed:



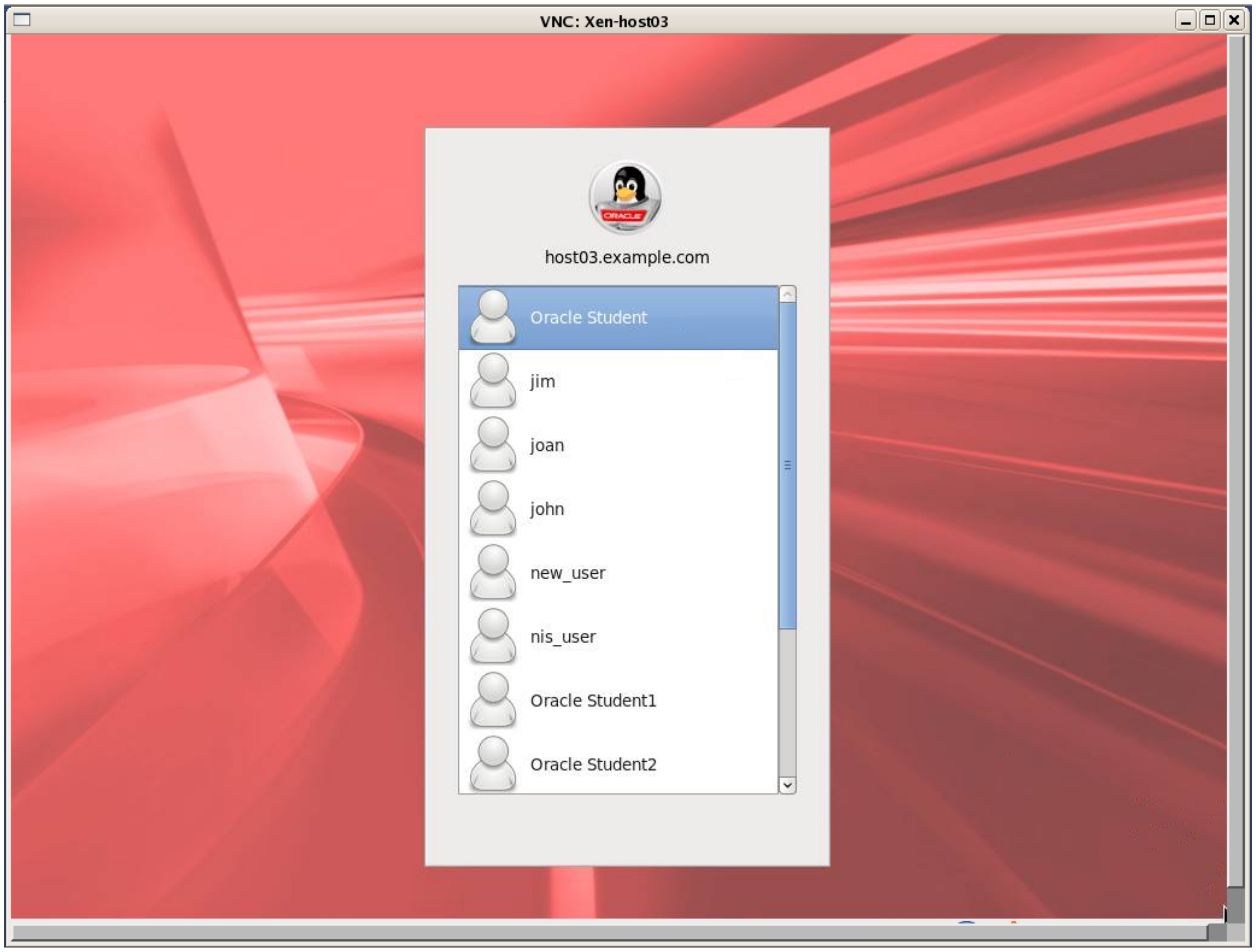
   c. Enter the `localhost:<port_number>` command, substituting the port number displayed from the previous `xm list` command. For example, if the port number is `5903`, enter `localhost:5903` and click **OK**.



   - The GNOME desktop login window appears.

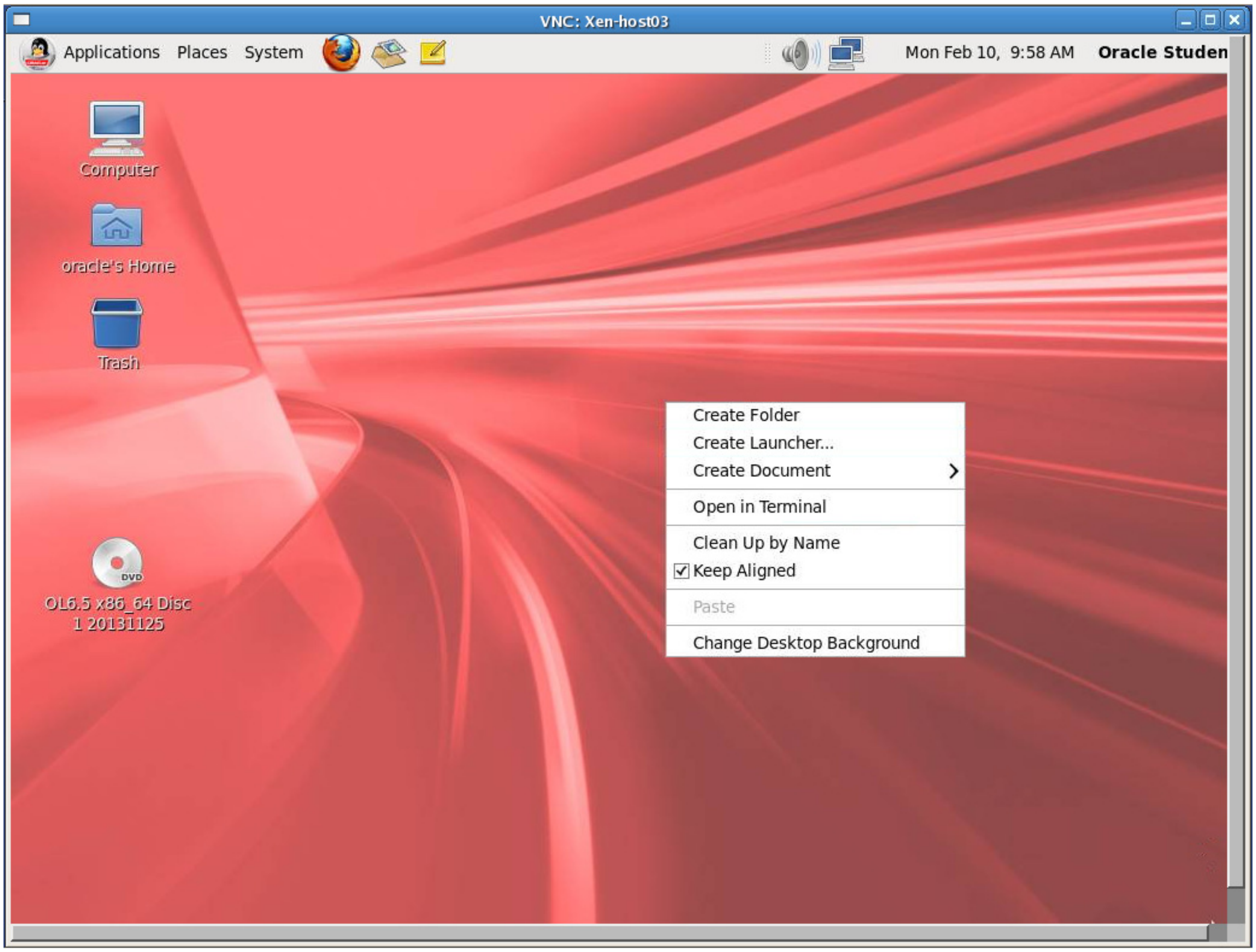
d. From the list of users, select **Oracle Student**.

- The password is `oracle`.



- The GNOME desktop appears.

e. Right-click on the desktop to display the pop-up menu.
- Select **Open in Terminal** from the pop-up menu.



f. In the terminal window, run the `su –` command to become the `root` user.
- The password is `oracle`.

```
[oracle@host03 Desktop]$ su –
Password: oracle
[root@host03 ~]#
```

2. As the `root` user on **host03**, explore the SELinux configuration.

a. Use the `cat` command to view the `/etc/selinux/config` file.

```
# cat /etc/selinux/config
...
SELINUX=enforcing
...
SELINUX=targeted
```

- Notice that SELinux security policy is set to `enforcing` mode.
- Notice that the `targeted` policy is in use, meaning that only targeted network daemons are protected.

b. Run the `sestatus` command.

```
# sestatus
SELINUX status:           enabled
...
Current mode:             enforcing
Mode from config file:    enforcing
...
Policy from config file:  targeted
```

- Notice that this output confirms the settings in the `/etc/selinux/config` file.

c. Run the `getenforce` command to display the current mode.

```
# getenforce
Enforcing
```

- Notice that this output also confirms the output from the `sestatus` command.
- If the current mode is **Permissive**, use the `setenforce 1` command to change the mode to **Enforcing**.
- Note that running `setenforce 0` would set the mode to **Permissive**.
- Using the `setenforce` command to change the mode is not persistent.

3. Perform SELinux package management.

a. Use the `rpm` command to list the installed `selinux` packages.

```
# rpm –qa |grep selinux
selinux-policy-targeted...
selinux-policy-...
libselinux-python-...
libselinux-utils-...
libselinux-...
```

b. Some SELinux tools are provided by other packages. Use the `rpm` command to list the `policycore` packages.

```
# rpm –qa |grep policycore
policycoreutils-...
```

- Notice that the `policycoreutils-gui` package, which provides the SELinux Administration GUI, is not installed by default.

c. Use the `yum` command to install the `policycoreutils-gui` package.

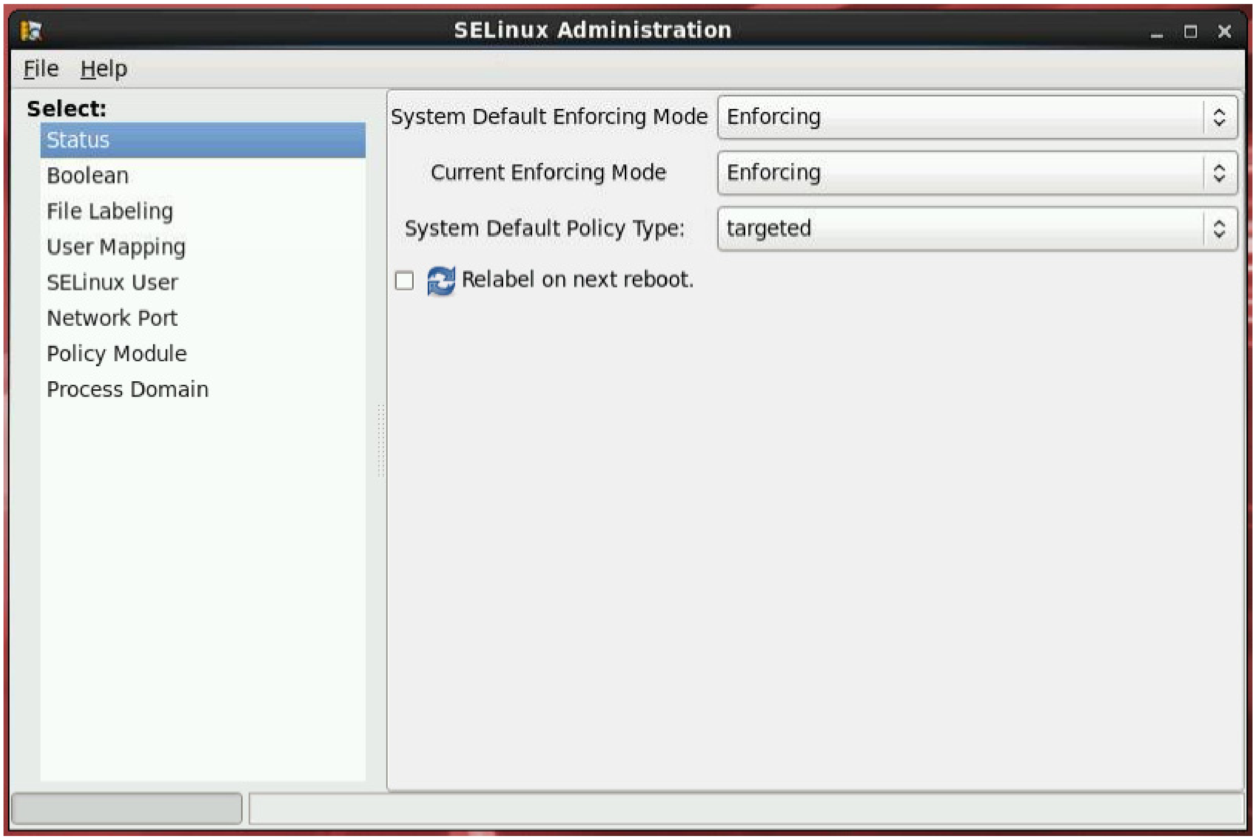
- This command also installs some dependency packages.

```
# yum install policycoreutils-gui
...
Transaction Summary
==========================================================
Install       9 Package(s)
Total download size: 1.8 M
Installed size 6.3 M
Is this ok [y/N]: y
...
Complete!
```

4. Use the SELinux Administration GUI.
   a. Use the `system-config-selinux` command to display the SELinux Administration GUI.

```
# system-config-selinux
```

   - The GUI appears.



   - Notice that policy type is **targeted** and policy mode is **Enforcing**.

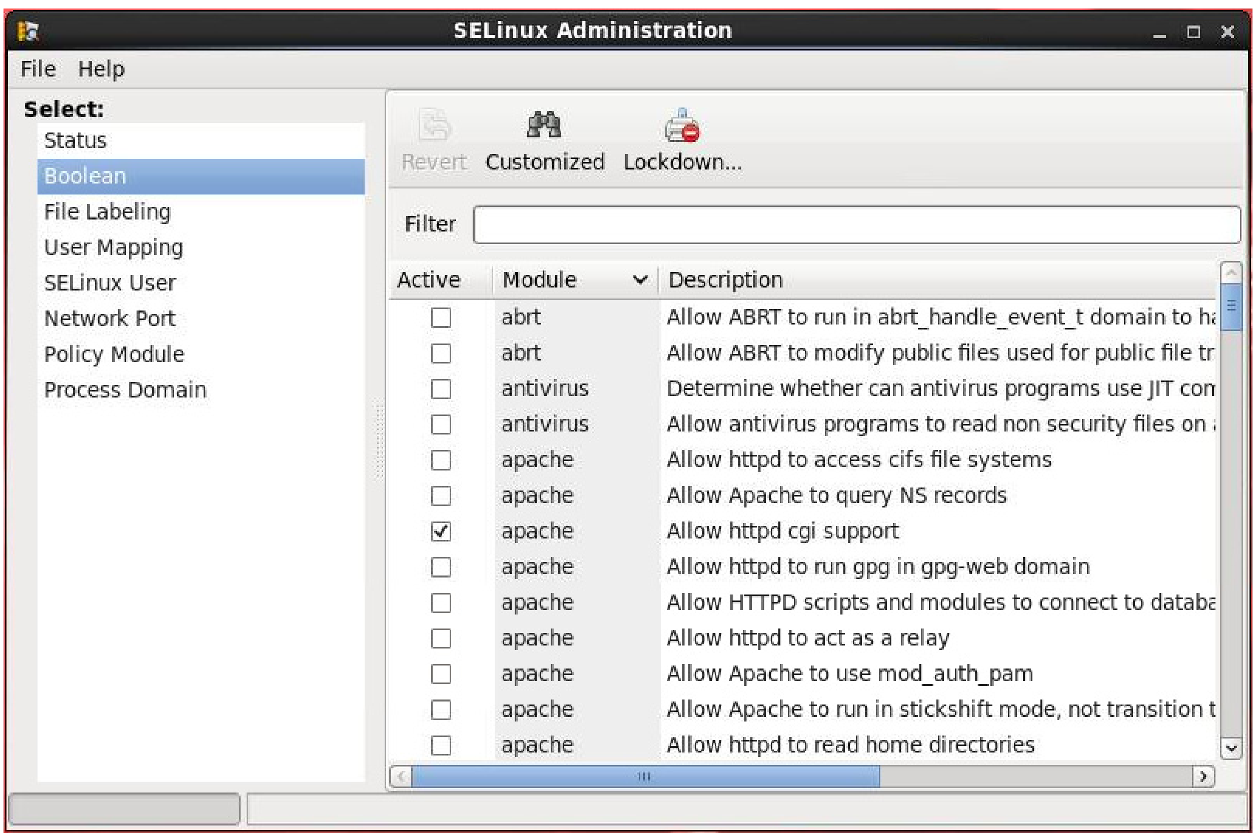
b. Display the **System Default Enforcing Mode** options by clicking the drop-down list.

- Notice the three options:
  - **Disabled**: No security policy is loaded in the kernel.
  - **Permissive**: A diagnostic state. Security policy rules are not enforced but messages are logged.
  - **Enforcing**: Access is denied to users and programs unless permitted by SELinux security policy rules.

c. Display the **System Default Policy Type** options by clicking the drop-down list.

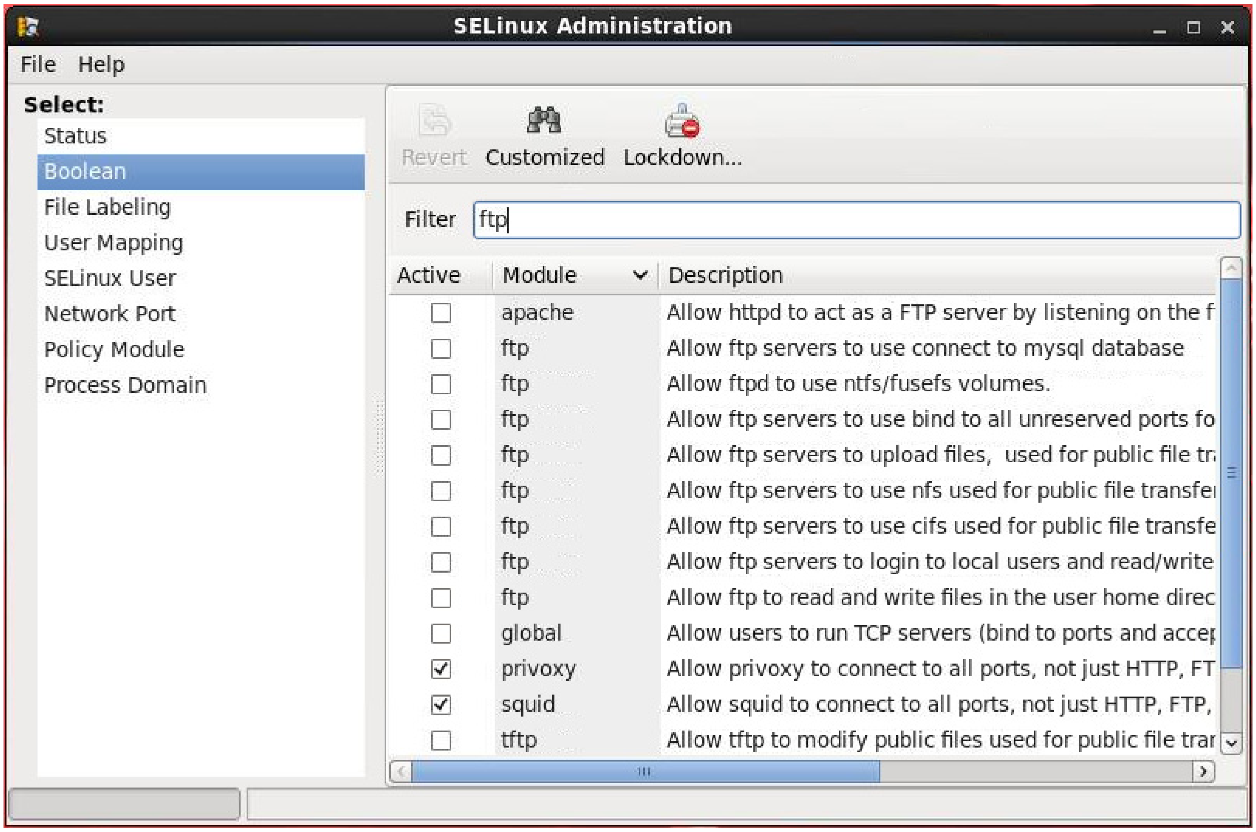
- Notice that **targeted** is the only option.
- The other possible policy type option is MLS (Multi-Level Security).
- MLS requires that the `selinux-policy-mls` package be installed (it is not installed by default).

d. Select **Boolean** in the left pane of the GUI.



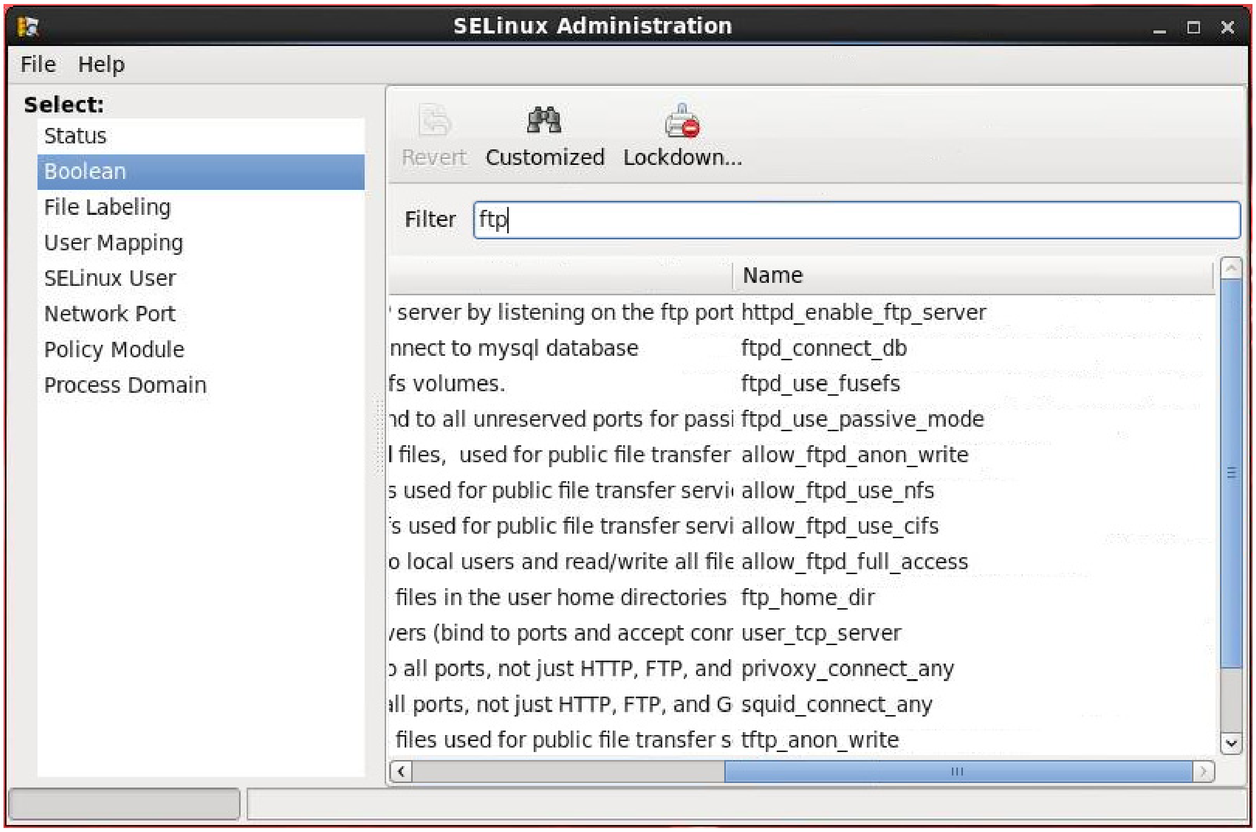
- Booleans allow parts of SELinux policy to be changed.
- The Active check box indicates whether the Boolean is on or off.

e. Enter `ftp` as a filter and press Enter to display the following screen.



- Notice that each item contains "`ftp`" in the long description.

f. Scroll to the right to display the Boolean name.



- Notice that each Boolean is referenced by a short name.
- The short name can be used as an argument to several SELinux commands.

g. Close the SELinux Administration GUI by selecting **File > Quit** from the menu bar.

5. Manage SELinux Booleans.

a. Use the `semanage` command to list the Booleans.

- If the `semanage` command is not found, use `yum` to install the `policycoreutils-python` package.

```
# semanage boolean -l
SELINUX Boolean   State Default Description
ftp_home_dir      (off ,  off)  Allow ftp to read and write...
smartmon_3ware    (off ,  off)  Enable additional permissions...
...
```

- Notice that the Boolean name is listed in the first column, the current setting (`on|off`) is given in the second column, the default setting (`on|off`) is given in the third column, and the long description is listed in the last column.

b. Use the `getsebool` command to view Booleans and their status.

- Use the `-a` option to display all Booleans and their status.

```
# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
allow_console_login --> on
```

```
...
xserver_object_manager --> off
zabbix_can_network --> off
```

- Notice that this command does not include the long description.

c. Include a Boolean name (for example, `use_nfs_home_dirs`) as an argument to display a specific Boolean status.

```
# getsebool use_nfs_home_dirs
use_nfs_home_dirs --> on
```

- **Note:** The `getsebool` command accepts multiple Boolean names as arguments.

d. You can also view Booleans in the `/selinux/booleans` directory. Use the `cat` command to view the status of both `use_nfs_home_dirs` and `abrt_anon_write` Booleans.

```
# cat /selinux/booleans/use_nfs_home_dirs
1 1
# cat /selinux/booleans/abrt_anon_write
0 0
```

- A value of `1` indicates that the Boolean is `on`, while `0` indicates `off`.
- The first number indicates the current value of the Boolean.
- The second number represents the pending value of the Boolean.

e. Use the `echo` command to change the pending value of the `abrt_anon_write` Boolean to `on`.

```
# echo 1 > /selinux/booleans/abrt_anon_write
# cat /selinux/booleans/abrt_anon_write
0 1
```

f. Use the `echo` command to commit all pending values for all Booleans.

- The `/selinux/commit_pending_bools` file is the interface for committing the pending values of all Booleans as the current values.
- Commit pending values by writing a `1` to this file.

```
# echo 1 > /selinux/commit_pending_bools
```

g. Use the `cat` command to view the `/selinux/booleans/abrt_anon_write` file and use the `getsebool` command to view the `abrt_anon_write` Boolean to confirm the commit.

```
# cat /selinux/booleans/abrt_anon_write
1 1
# getsebool abrt_anon_write
abrt_anon_write --> on
```

h. Use the `setsebool` command to change the value of the `abrt_anon_write` Boolean. Confirm the change.

```
# setsebool abrt_anon_write off
# getsebool abrt_anon_write
abrt_anon_write --> off
# cat /selinux/booleans/abrt_anon_write
0 0
```

- The `setsebool -P` command causes the change to persist across a reboot.

Practice 16-2: Configuring an SELinux Boolean

Overview

In this practice, you change the value of an SELinux Boolean to allow `ftp` to read and write files in user home directories.

Assumptions

- This practice is performed on **host01** and **host03**.
- You are currently the `root` user on **host03**.
- You are instructed to connect to **host01** from **dom0** by using `ssh`.
- The prompts include either **host01** or **host03** to indicate which system to enter the command from.

Tasks

1. Use the `yum` command to install the `vsftpd` package on **host03**.

```
[host03]# yum install vsftpd
...
Transaction Summary
============================================================
Install       1 Package(s)
Total download size: 150 k
Installed size: 331 k
Is this ok [y/N]: y
...
Complete!
```

2. Use the `service` command to start the `vsftpd` service.

```
[host03]# service vsftpd start
Starting vsftpd for vsftpd:                [  OK  ]
```

3. As the `root` user on **dom0**, connect to the **host01** guest by using `ssh`.
   a. If necessary, open a terminal window on **dom0** and use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
[dom0]$ su –
Password: oracle
#
```

   b. As the `root` user on **dom0**, use the `ssh` command to login to **host01**.
      - The password is `oracle`.

```
[dom0]# ssh host01
root@host01's password: oracle
Last login: ...
[root@host01 ~]#
```

4. Use the `yum` command to install the `ftp` (client) package on **host01**.

```
[host01]# yum install ftp
...
Transaction Summary
============================================================
Install       1 Package(s)
Total download size: 57 k
Installed size: 95 k
Is this ok [y/N]: y
...
Complete!
```

5. From **host01**, use the `ftp` utility to connect to the FTP server, **host03**, as `anonymous` user.
   - The connection fails. Use the `quit` command to exit `ftp`.

```
[host01]# ftp host03
ftp: connect: No route to host
ftp> quit
[host01]#
```

   - Notice the error message, “No route to host”.
   - The firewall running on **host03** is denying the `ftp` connection.

6. From **host03**, use the `service` command to stop `iptables`.

```
[host03]# service iptables stop
iptables: Setting chains to policy ACCEPT: filter  [  OK  ]
iptables: Flushing firewall rules:                 [  OK  ]
iptables: Unloading modules:                       [  OK  ]
```

7. From **host01**, use the `ftp` utility to connect to the FTP server, **host03**, as `anonymous` user.
   - Press **ENTER** when prompted for a password.
   - Notice that you can successfully connect to **host03** when the firewall is disabled.
   - Use the `quit` command to exit `ftp`.

```
[host01]# ftp host03
Connected to host03 (192.0.2.103).
220 (vsFTPd 2.2.2)
Name (host03:root): anonymous
331 Please specify the password.
Password: ENTER
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
[host01]#
```

8. From **host01**, use the `ftp` utility to connect to the FTP server, **host03**, as `oracle` user.
   - The password is `oracle`.
   - Notice that you are unable to change to the `/home/oracle` directory and the login fails.
   - Use the `quit` command to exit `ftp`.

```
[host01]# ftp host03
Connected to host03 (192.0.2.103).
220 (vsFTPd 2.2.2)
Name (host03:root): oracle
331 Please specify the password.
Password: oracle
500 OOPS: cannot change directory:/home/oracle
Login failed.
ftp> quit
[host01]#
```

9. Enable SELinux Boolean on **host03**.
   a. Use the `getsebool –a` command to list all the SELinux Booleans and their statuses. Pipe the output to `grep` and search for `ftp`.

```
[host03]# getsebool –a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
...
```

   - Notice that the `ftp_home_dir` Boolean is set to `off`.

   b. Use the `semanage boolean -l` command to list the SELinux Booleans and their long descriptions. Pipe the output to `grep` and search for `ftp_home_dir`.

```
[host03]# semanage boolean –l | grep ftp_home_dir
ftp_home_dir (off  ,  off)  Allow ftp to read and write files in
the user home directories
```

   - Based on the long description, this Boolean appears to be the reason you cannot `ftp` to **host03** as the `oracle` user.

   c. Use the `setsebool` command to enable the `ftp_home_dir` Boolean.
   - Use the `-P` option to make the change persistent across reboots. There is a slight delay when using the `-P` option.
   - Use the `getsebool` command to confirm that the Boolean is `on`.

```
[host03]# setsebool –P ftp_home_dir on
[host03]# getsebool ftp_home_dir
ftp_home_dir --> on
```

10. From **host01**, use the `ftp` utility to connect to the FTP server, **host03**, as `oracle` user.
    - The password is `oracle`.
    - Notice that the login is successful when the `ftp_home_dir` Boolean is enabled on the FTP server.
    - Use the `quit` command to exit `ftp`.

```
[host01]# ftp host03
Connected to host03 (192.0.2.103).
220 (vsFTPd 2.2.2)
Name (host03:root): oracle
331 Please specify the password.
Password: oracle
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
[host01]#
```

11. Log off **host01** and **host03**.
    a. From **host01**, enter `exit` to log off.

```
[host01]# exit
logout
Connection to host01 closed.
[dom0]#
```

    b. From **host03**, select **System > Log Out oracle…** from the GNOME menu bar.
       - Click **Log Out** when prompted.
       - Click the `X` in the upper-right corner of the GNOME login window to close the window.

Practice 16-3: Configuring SELinux Context

Overview

In this practice, you change the SELinux context file type to enable a file system to be used for storing KVM virtual machine disk images when SELinux is in enforcing mode.

Assumptions

- You are the `root` user on **dom0**.

Tasks

1. Shut down the **host01**, **host02**, and **host03** VMs.
   a. From **dom0**, run `xm destroy hostN` command to shut down the three VMs.

```
# xm destroy host01
# xm destroy host02
# xm destroy host03
```

   b. Run the `xm list` command to display the running VMs.

```
# xm list
Name                ID     Mem    VCPUs      State    Time(s)
Domain-0             0    2048        2     r-----        ...
```

   - In this example, only **Domain-0** is running.

2. Start the **host05** virtual machine.
   - Before starting **host05**, you create a second disk and update the `vm.cfg` file.

   a. From **dom0**, use the `cd` command to change to the `/OVS/running_pool/host05` directory.

```
# cd /OVS/running_pool/host05
```

   b. Use the `dd` command to create a 10 GB utility disk image.
   - This command takes a few seconds to complete.

```
# dd if=/dev/zero of=u01.img bs=1M count=10240
10240+0 records in
10240+0 records out
10737418240 bytes (11 GB) copied...
```

   c. Use the `vi` editor to add the following **bold** line to the `vm.cfg` file.
   - This line allows the **host05** VM to recognize the new `u01.img` disk image.

```
# vi vm.cfg
...
disk = [ 'file:/OVS/running_pool/host05/system.img,hda,w',
         'file:/OVS/running_pool/host05/u01.img,hdb,w',
         'file:/OVS/seed_pool/OracleLinux-R6-U5-Server-x86_64-
dvd.iso,hdc:cdrom,r']
...
```

d. Run the `xm create` command to start the **host05** virtual machine.

```
# xm create vm.cfg
Started domain host05 (id=…
```

3. Log in to **host05**.

a. Use the `ssh` command to log in to `192.0.2.105`. The **host05** VM is not resolved on **dom0** so you must use the IP address rather than the host name.

- The `root` password is `oracle` (all lowercase).
- Enter the `hostname` command to verify you connected to **host05**.

```
# ssh 192.0.2.105
The authenticity of host '192.0.2.105 (192.0.2.105)' can't be established. RSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.0,2,105' (RSA) to the list of known hosts.
root@192.0.2.105's password: oracle
[root@192.0.2.105 ~]# hostname
host05.example.com
```

4. Configure **host05** to access the local Yum repository on **dom0**.

a. From **host05**, use the `cd` command to change to the `/etc/yum.repos.d` directory.

```
# cd /etc/yum.repos.d
```

b. Use the `vi` command to create the `vm.repo` file as follows.

```
# vi vm.repo
[OL6U5Dom0]
Name="Oracle Linux 6 U5 Dom0 Repo"
baseurl=http://192.0.2.1/repo/OracleLinux/OL6/5/base/x86_64
enabled=1
gpgkey=http://192.0.2.1/repo/OracleLinux/OL6/5/base/x86_64/RPM-GPG-KEY-oracle
gpgcheck=1
```

- Notice that the `baseurl` references the local Yum repository on **dom0** (`192.0.2.1`).

c. Use the `vi` command to edit the `public-yum-ol6.repo` file and disable all repositories.

- Set `enabled=0` for all repositories.

```
# vi public-yum-ol6.repo
[public_ol6_latest]
...
enabled=0
...
[public_ol6_UEK_latest]
...
enabled=0
```

```
...
[public_ol6_UEKR3_latest]
...
enabled=0
...
```

d. Use the `grep` command to view enabled repositories in both files.

```
# grep enabled *
public-yum-ol6.repo:enabled=0
...
vm.repo:enabled=1
```

- Only the `vm.repo` file contains an enabled repository (`enabled=1`).
- If this is not the case, repeat task 4 to ensure that only the `vm.repo` is enabled.

e. Run the `yum clean all` command to clean the Yum cache.

```
# yum clean all
Loaded plugins: ...
Cleaning repos: ...
Cleaning up Everything
```

5. Use the `yum` command to install the `policycoreutils-python` package.
   - This package contains the `semanage` utility which is needed in the next task.
   - Answer "`y`" when prompted "**Is this ok [y/N]:**".
   - You are asked about the GPG key only the first time you use the `yum install` command.

```
# yum install policycoreutils-python
...
Transaction Summary
============================================================
Install       5 package(s)
Total download size: 1.1 M
Installed size: 4.1 M
Is this ok [y/N]: y
...
Importing GPG key ...
...
Is this ok [y/N]: y
...
Complete!
```

6. Prepare the new disk for use.
   a. Use the `fdisk` command to partition the `/dev/xvdb` device.
      - Create a single, primary partition for the entire disk.
      - Press ENTER when prompted for the first cylinder and last cylinder.

```
# fdisk /dev/xvdb
...
Command (m for help): n
Command action
   e   extended
   P   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1): ENTER
Using default value 1
Last cylinder, +cylinders or +size{K,M,G}... ENTER
Using default value 1305
Command (m for help): w
The partition table has been altered!
...
Syncing disks.
```

   b. Use the `pvcreate` command to create a physical volume on `/dev/xvdb1`.

```
# pvcreate /dev/xvdb1
  Physical volume "/dev/xvdb1" successfully created
```

   c. Use the `vgcreate` command to create a volume group, named `myvolg`, on `/dev/xvdb1`.

```
# vgcreate myvolg /dev/xvdb1
  Volume group "myvolg" successfully created
```

   d. Use the `lvcreate` command to create a 5 GB logical volume group, named `myvol`, from the `myvolg` volume group.

```
# lvcreate –L 5G –n myvol myvolg
  Logical Volume "myvol" created
```

   e. Use the `mkfs.ext4` command to create an ext4 file system on the `myvol` logical volume.

```
# mkfs.ext4 /dev/mapper/myvolg-myvol
...
Writing superblocks and filesystem accounting information: done
```

   f. Use the `mkdir` command to create a new directory, `/virtstore`, for mounting the new logical volume.

```
# mkdir /virtstore
```

   g. Use the `mount` command to mount the logical volume on `/virtstore`.

```
# mount /dev/mapper/myvolg-myvol /virtstore
```

7. Change the SELinux context file type on `/virtstore`.
   a. Use the `ls -dZ` command to view context information for the `/var/lib/libvirt/images` directory.
      - The `/var/lib/libvirt/images` directory is the default directory for KVM virtual machine disk images.

```
# ls -dZ /var/lib/libvirt/images
drwx...  root root system_u:object_r:virt_image_t:s0  /var/...
```

      - Notice that this directory has the `virt_image_t` label applied to it.

   b. Use the `ls -dZ` command to view context information for the `/virtstore` directory.

```
# ls -dZ /virtstore
drwx...  root root system_u:object_r:file_t:s0  /virtstore/
```

      - Notice that this directory has the `file_t` label applied to it.

   c. Use the `semanage` command to set the SELinux file context type to `virt_image_t` for the `/virtstore` directory.

```
# semanage fcontext -a -t virt_image_t "/virtstore(/.* )?"
```

   d. Use the `cat` command to view the context information for the `targeted` SELinux policy.

```
# cat /etc/selinux/targeted/contexts/files/file_contexts.local
...
/virtstore(/.* )?   system _u:object_r:virt_image_t:s0
```

   e. Use the `ls -dZ` command to view context information for the `/virtstore` directory.

```
# ls -dZ /virtstore
drwx...  root root system_u:object_r:file_t:s0  /virtstore/
```

      - Notice that this directory still has the `file_t` label applied to it.

   f. Use the `restorecon` command to change the type on `/virtstore` as defined in the `/etc/selinux/targeted/contexts/files/file_contexts.local` file.
      - The `-R` option recursively changes file and directory file labels.
      - The `-v` option shows the changes in file labels.

```
# restorecon -R -v /virtstore
restorecon reset /virtstore context system_u:object_r...
...
```

   g. Use the `ls -dZ` command to view context information for the `/virtstore` directory.

```
# ls -dZ /virtstore
drwx...  root root system_u:object_r:file_t:s0  /virtstore/
```

      - Notice that the directory no has the `virt_image_t` label applied to it.
      - The `/virtstore` file system is now prepared to store KVM virtual machine disk images when SELinux is in enforcing mode.

8. Clean up.
   - In this task, you shut down **host05** and re-start the **host03** VM.

   a. Run the `shutdown -h now` command to shut down **host05**.

```
[host05]# shutdown -h now
...
The system is going down for halt NOW!
Connection to 192.0.2.105 closed by remote host.
Connection to 192.0.2.105 closed.
```

   b. From **dom0**, use the `cd` command to change to the `/OVS/running_pool/host03` directory.

```
[dom0]# cd /OVS/running_pool/host03
```

   c. Run the `xm create vm.cfg` command to start the **host03** VM.

```
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host03 (id=...)
```

   d. Run the `xm list` command to display the running VMs.

```
# xm list
Name               ID     Mem    VCPUs     State   Time(s)
Domain-0            0    2048        2    r-----       ...
host03            ...
```

   - In this example, the **host03** VM is running.
   - It is not necessary to start the **host01** and **host02** VMs because these VMs are not used in the remaining practices.

Practices for Lesson 17: Core Dump Analysis

Chapter 17

Practices for Lesson 17: Core Dump Analysis

Practices Overview

In these practices, you:

- Configure Kdump
- Create a core dump file
- Install the kernel-debuginfo RPMs
- Use the `crash` utility to analyze the state of a vmcore
- Use the `crash` utility to analyze the state of a running system

Practice 17-1: Configuring Kdump

Overview

In this practice, you:

- Configure Kdump by using the Kernel Dump Configuration GUI
- View the Kdump configuration, which is updated as you apply changes in the GUI

Assumptions

- You are the `root` user on **dom0**.
- The Kdump mechanism is not supported on your Xen domU guest systems.
- However, you proceed with the practice as if Kdump is supported.

Tasks

1. Connect to **host03** using `vncviewer`. If necessary, refer to Practice 3-1 for instructions.
   - From the GNOME login window on **host03**, log in as the **Oracle Student** user. The password is `oracle`.
   - From the GNOME desktop on **host03**, open a terminal window.
   - Use the `su –` command to become the `root` user. The password is `oracle`.
2. As the `root` user on **host03**, use the `rpm` command to verify that the `kexec-tools` package is installed.

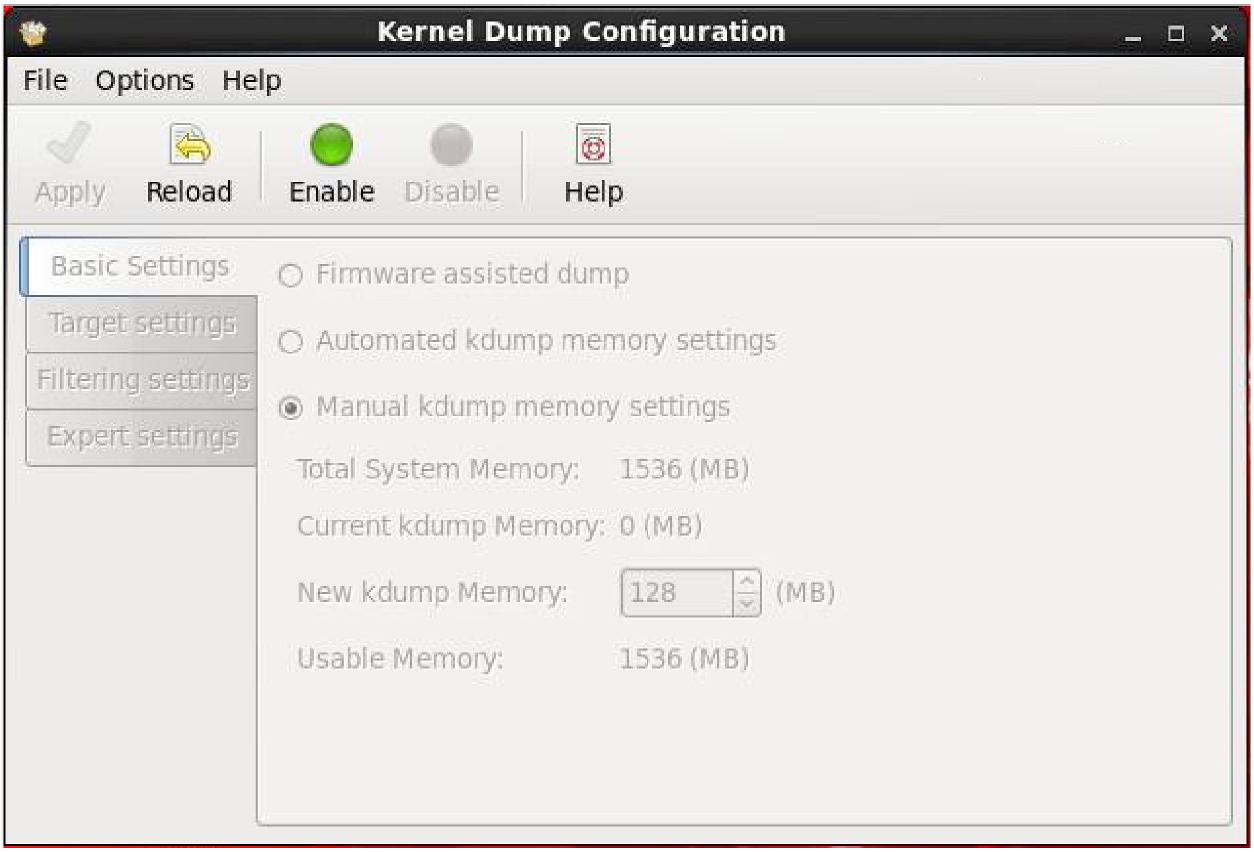
```
# rpm -q kexec-tools
kexec-tools-2.0.3-3.0.10.el6.x86_64
```

   - In this example, the package is installed.

3. Configure Kdump by using the Kernel Dump Configuration GUI.
   a. Use the `system-config-kdump` command to display the Kernel Dump Configuration GUI.

```
# system-config-kdump
```

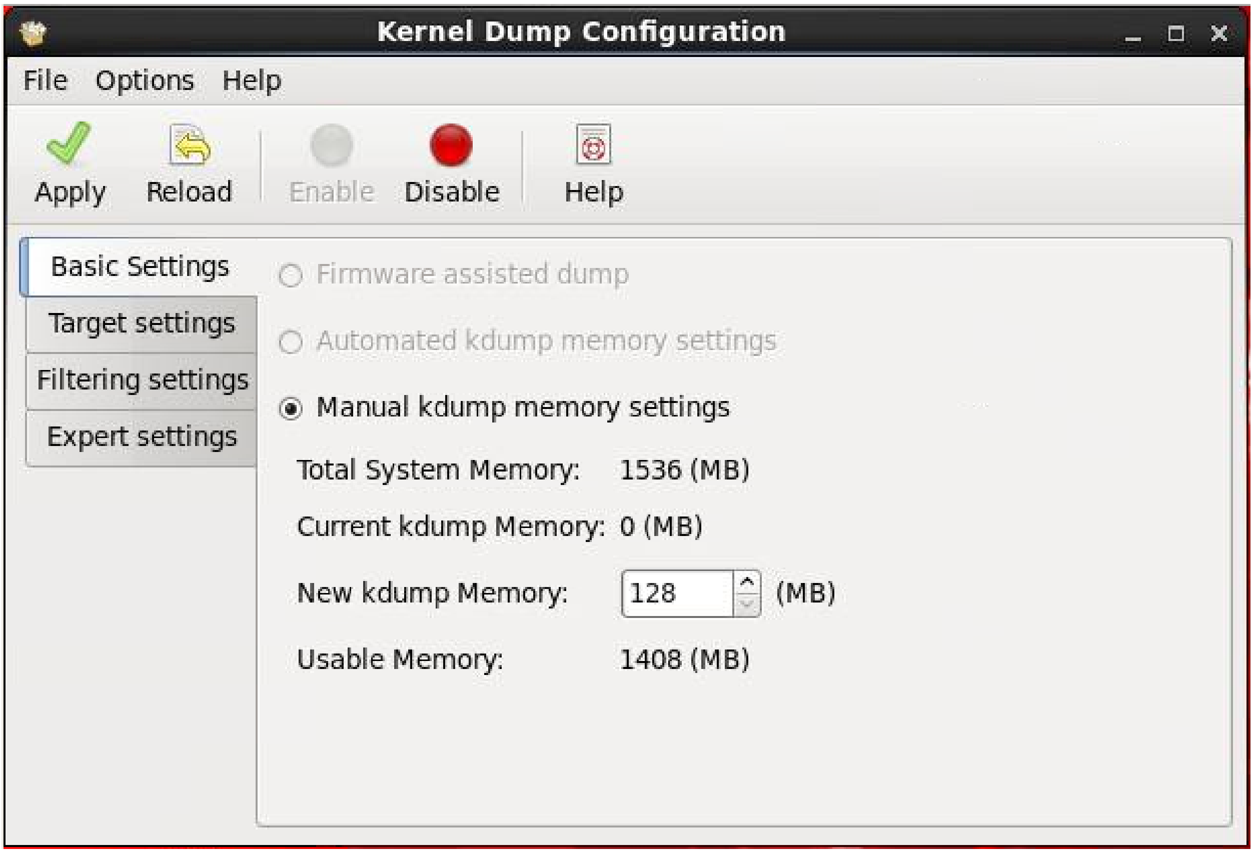
- The GUI appears.



- In this example, the kernel crash dump mechanism is disabled.

b. Click the **Enable** button.

- This configures the `kdump` service to start at boot time for run levels 2, 3, 4, and 5.

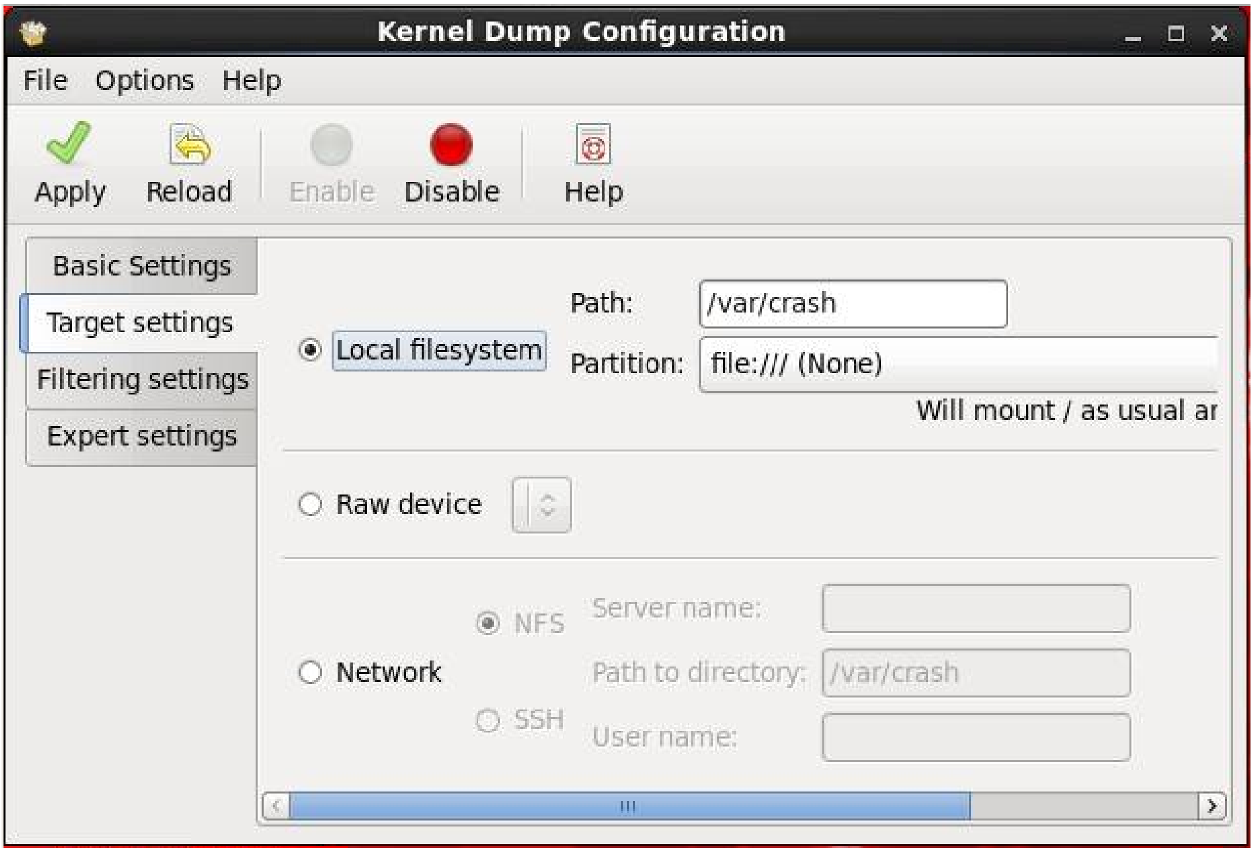


- Notice that the amount of memory reserved for Kdump is 128 MB.
- This amount of memory is represented by the `crashkernel` option and is appended to the kernel line in the GRUB configuration file, `/boot/grub/grub.conf`, as follows:

```
kernel /vmlinuz-3.8.13-26.1.1.el6uek ... crashkernel=128M
```

- You must reboot if the amount of memory reserved for Kdump is changed or if Kdump is enabled from a disabled state.

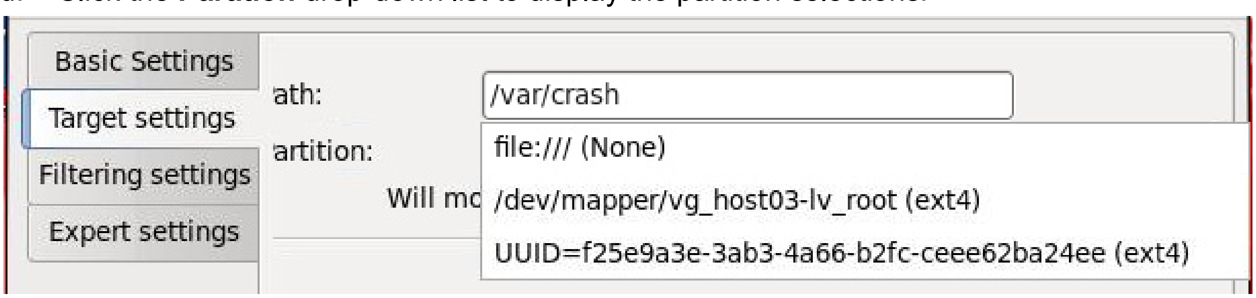
c. Click the **Target settings** tab in the GUI.



- This page is used to specify the target location for the `vmcore` dump.
- Notice that the default target location is the `/var/crash` directory on the local file system.
- This setting is represented as follows in the `/etc/kdump.conf` configuration file:

  `path /var/crash`

d. Click the **Partition** drop-down list to display the partition selections.



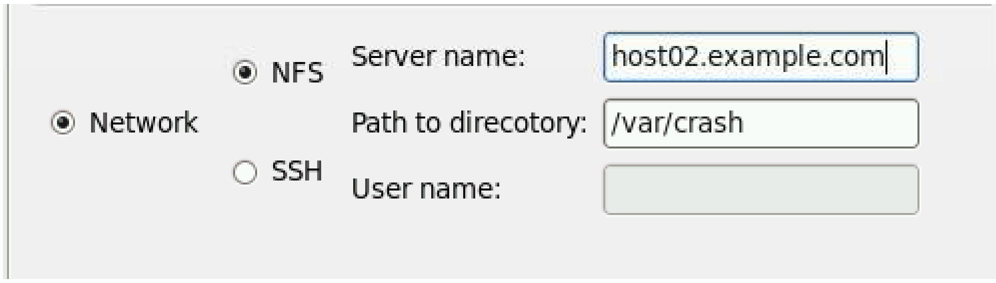
- Notice that the available devices on the local system are displayed (sample display only).

e. Select the `UUID=` entry and set the **Path** at `/var/crash`.

- These settings are represented as follows in the `/etc/kdump.conf` configuration file:

```
path /var/crash
ext4 UUID=<your selection>
```

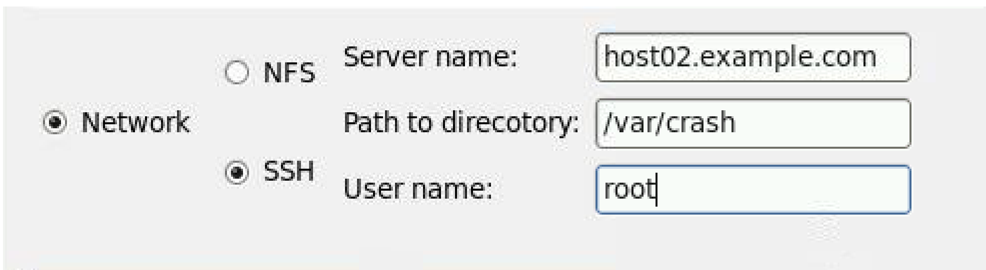
f. Click the **Network** and **NFS** buttons and enter `host02.example.com` for **Server name**.



- These settings are represented as follows in the `/etc/kdump.conf` configuration file:

```
path /var/crash
net host02.example.com
```

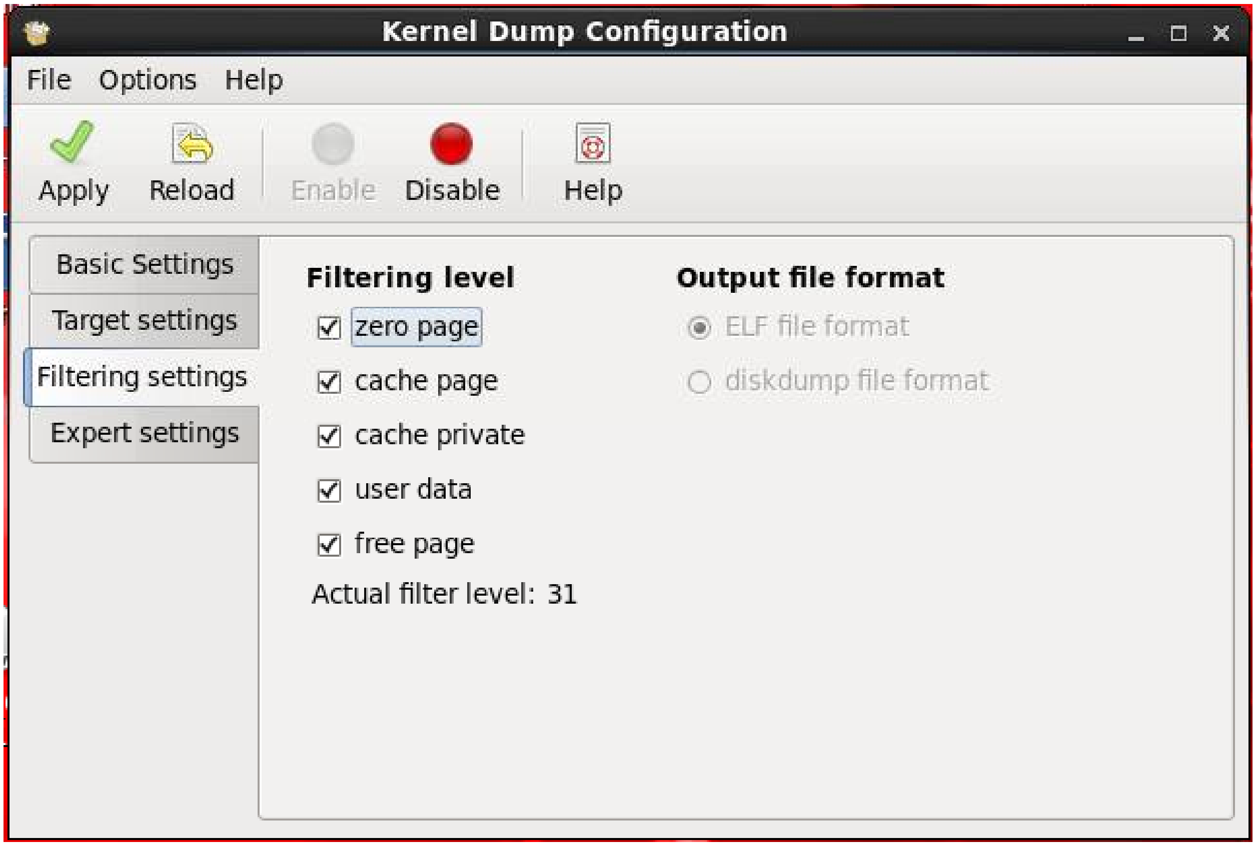
g. Click the **Network** and **SSH** buttons and enter `root` as the **User name**.



- These settings are represented as follows in the `/etc/kdump.conf` configuration file:

```
path /var/crash
net root@host02.example.com
```

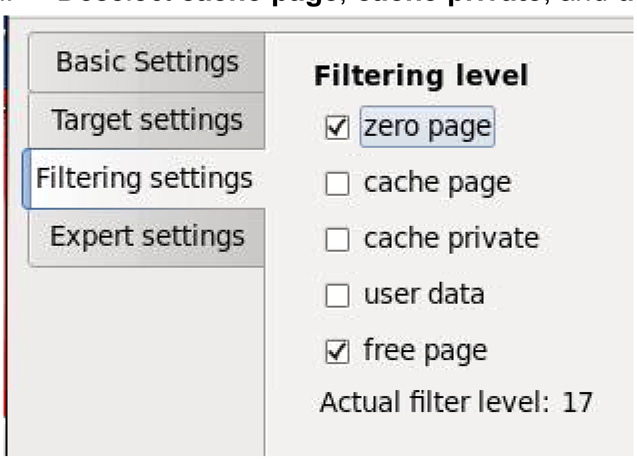
h. Click the **Filtering settings** tab in the GUI.



- Check the pages that you do not want to include in the core dump.
- By default, all page types are excluded. The default settings are represented as follows in the `/etc/kdump.conf` configuration file:

```
core_collector makedumpfile -p --message-level 1 -d 31
```

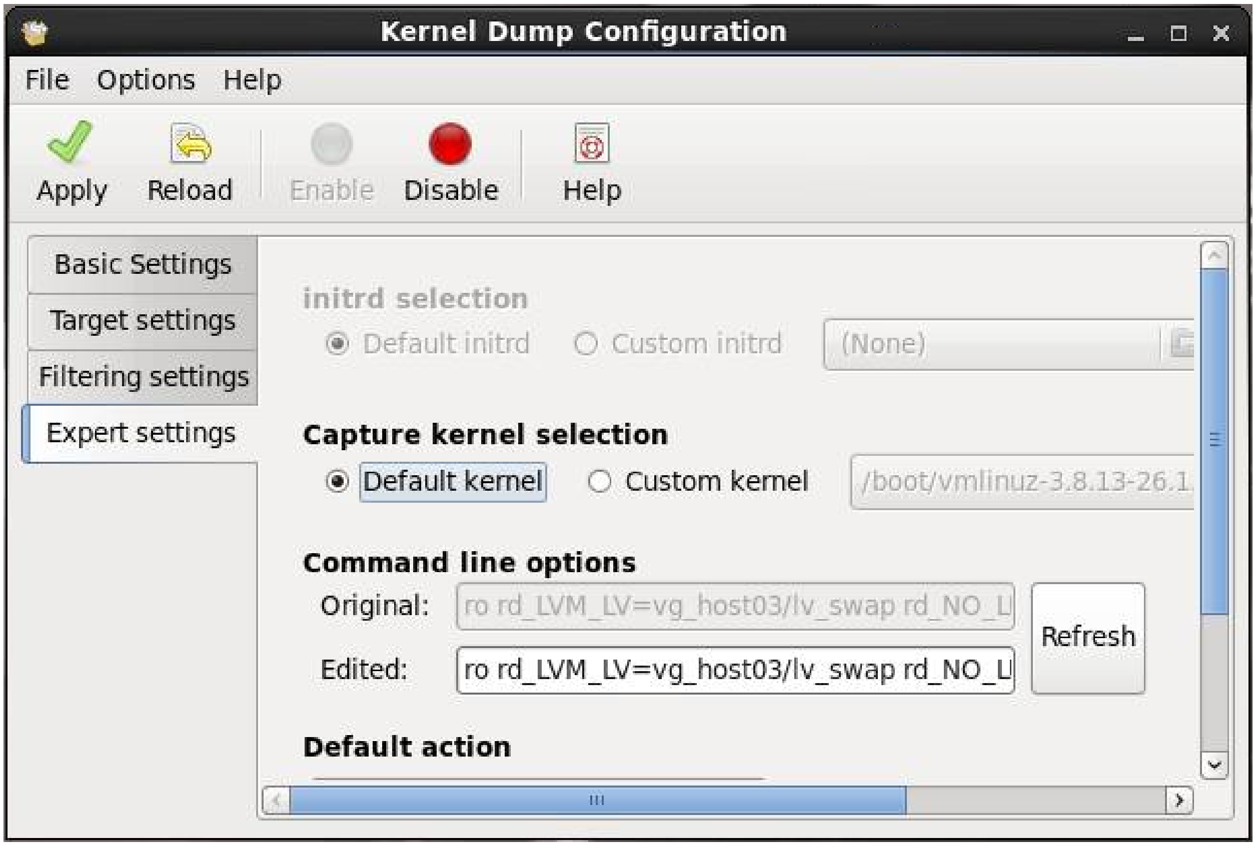
i. Deselect **cache page**, **cache private**, and **user data**.



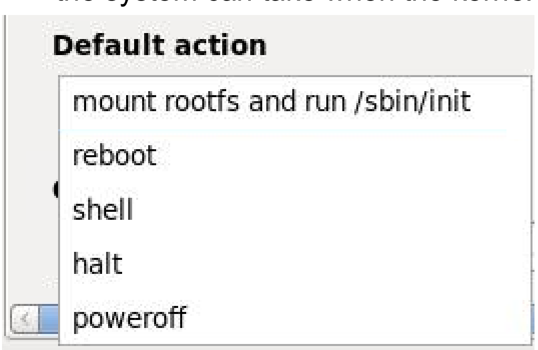
- These settings exclude only zero page and free page information. This setting is represented as follows in the `/etc/kdump.conf` configuration file:

```
core_collector makedumpfile -p -message-level 1 -d 17
```

j. Click the **Expert settings** tab in the GUI.



k. Scroll down and click the **Default action** drop-down menu to display the actions that the system can take when the kernel crash is captured.

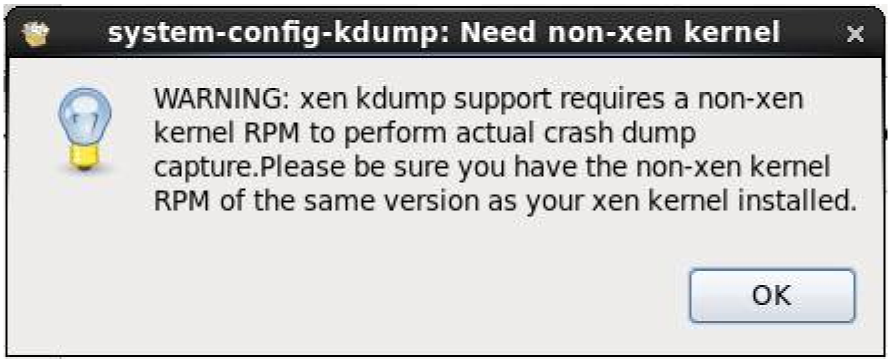


l. Click the **halt** option.

- This selection is represented as follows in the `/etc/kdump.conf` configuration file:
  `default halt`

m. Click **Apply**.

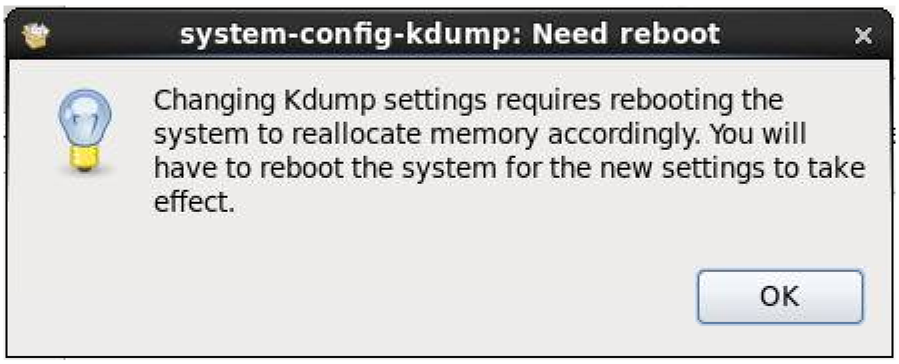
- The following message appears:



- The Kdump mechanism is not supported on Xen domU guest systems.

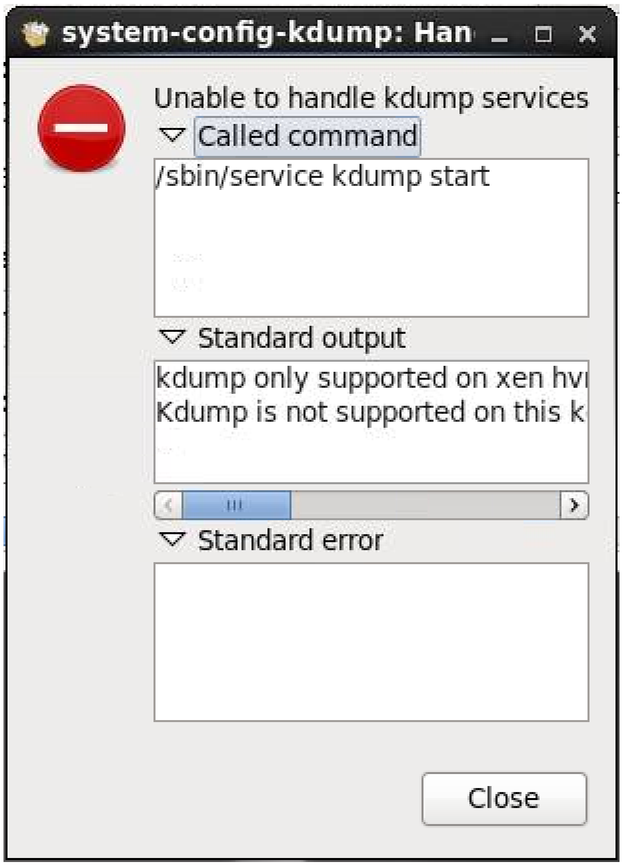
n. Click **OK**.

- The following message appears if you change memory settings and if you enable Kdump from a disabled state.



- Again, this message does not apply, because Kdump is not supported on the VM guests.

o. Click **OK.** The following message appears:



- The configuration is saved, but the service does not start on VM guests.

p. Click **Close**, and then select **File > Quit** from the Kernel Dump Configuration menu bar.

4. View the configuration changes made by using the Kdump GUI.

   a. From a terminal window on **host03**, run the `chkconfig kdump --list` command.

```
# chkconfig kdump --list
kdump    0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

   - By clicking **Enable** on the Kdump GUI, the `kdump` service is configured to start at boot time for run levels 2, 3, 4, and 5.

   b. Use the `tail -4 /etc/kdump.conf` command to display the last four lines in the Kdump configuration file.

```
# tail -4 /etc/kdump.conf
net root@host02.example.com
path /var/crash
core_collector makedumpfile –p –message-level 1 -d 17
default halt
```

   - Notice that the settings made from the GUI were written to the Kdump configuration file.

   c. Use the `makedumpfile --help | less` command to view usage for the `makedumpfile` command.

   - The `makedumpfile` command is designated as the `core_collector` in the `/etc/kdump.conf` file.

- Pipe the output of the `makedumpfile --help` command to `less` to view the help pages one screen at a time.

```
# makedumpfile --help | less
...
```

- Scroll down and notice that the `-c` option compresses the dump data.
- Notice that the default dump_level, `-d 31`, excludes zero page, cache with and without private, user data, and free page, and only captures the essential information.
- Specifying `-d 17` excludes zero page and free page information.
- Notice that the default `--message-level` of `1` prints a progress indicator. A message level of `31` displays progress indicator, common messages, error messages, debug messages, and report messages.
- Enter `q` to quit the `makedumpfile ... | less` command.

d. Run the `uname -r` command to display the running kernel version.

```
# uname -r
3.8.13-26.1.1.el6uek.x86_64
```

e. Use the `grep` command to search for the string "`kernel`" in the `/boot/grub/grub.conf` file.

```
# grep kernel /boot/grub/grub.conf
...
kernel /vmlinuz-3.8.13-26.1.1.el6uek...    crashkernel=128M
...
```

- Notice that `crashkernel=128M` is appended to the appropriate kernel line in the GRUB configuration file.

5. View the memory statistics

a. Use the `free –m > ~/before` command to write memory statistics to a file.

```
# free –m > ~/before
```

- This task saves current memory usage to a file.
- The contents of this file are compared to memory usage after configuring Kdump to reserve 128 MB of system memory for the capture kernel.

b. Use the `reboot` command to reboot **host03**.

```
# reboot
...
```

c. From **dom0**, use the `ssh` command to connect to **host03**.

- The `root` password is `oracle`.
- You need to wait a few seconds for the reboot to complete.

```
[dom0]# ssh host03
root@host03's password: oracle
```

d. From **host03**, compare the `free –m` output to the contents of `~/before`.

```
# cat ~/before
             total ...
Mem:          1498 ...
...
# free –m
             total ...
Mem:          1370 ...
...
```

- Notice that there is 128 MB less total memory (1498-1370=128) after reserving this amount for the capture kernel.

Practice 17-2: Creating a Core Dump File

Overview

In this practice, you perform the following:

- Because Kdump is not supported on your Xen domU guest systems, use the `xm dump-core` command to create a vmcore file for **host03**.
- Use the `sftp` command to upload the vmcore file from **dom0** to **host03**.

Assumptions

- You are the `root` user on **host03**.

Tasks

1. If necessary, open a new terminal window from **dom0**.
   a. From this window, use the `su -` command to become the `root` user on **dom0**.
      - The root password is `oracle`.

```
[dom0]$ su -
Password: oracle
```

2. List domain IDs.
   a. From **dom0**, use the `xm list` command to print information about domains. The output shown is a sample and might not represent your system exactly.

```
# xm list
Name                    ID    Mem  VCPUs       State      Time(s)
Domain-0                 0   2048      2      r-----       281.1
host03                 249   1536      1      -b----        13.2
```

   - In this example, there is one guest running, **host03**.
   - The **domain-id** for **host03** is `249`. This might differ on your system.

3. Dump **host03's** virtual memory.
   a. From **dom0**, use the `xm dump-core` command to dump **host03's** virtual memory.
      - Use the **domain-id** for **host03** that was obtained from the previous task.
      - The example command uses **domain-id** `249`. Your **domain-id** might be different.
      - This command takes a few seconds to complete.

```
# xm dump-core 249
Dumping core of domain: 249 ...
```

   b. Use the `cd` command to change to the `/var/xen/dump` directory on **dom0**. Use the `ls` command to display the contents of the directory.

```
# cd /var/xen/dump
$ ls
2014-0220-0525.00-host03.249.core
```

   - The vmcore file name is only a sample. Your vmcore file name is different.

4. Upload the vmcore file from **dom0** to **host03**.
   a. Use the `sftp` command to connect to **host03**.
      - The `root` password is `oracle`.

```
[dom0]# sftp host03
Connecting to host03
root@host03's password: oracle
sftp>
```

   b. Use the `put *` command to upload the vmcore file in the **dom0** directory to **host03**.

```
sftp> put *
Uploading 2014-0220-0525.00-host03.249.core to /root/2014...
sftp>
```

   c. Use the `quit` command to exit `sftp`.

```
sftp> quit
```

Practice 17-3: Preparing Your System to Analyze the vmcore

Overview

In this practice, you perform the following:

- Determine the kernel version and the architecture of the vmcore.
- Review the procedures to download the kernel-debuginfo packages from oss.oracle.com.
- Transfer the kernel-debuginfo packages from **dom0** to **host03**.
- Install the kernel-debuginfo packages on **host03**.
- Ensure that you have the latest version of `crash` installed.

Assumptions

- You are logged in as `root` on **dom0.**
- You are logged in as `root` on **host03.**

Tasks

1. Determine the kernel version and architecture of the vmcore.
   a. If you have access to the system that produced the vmcore, use the `uname -r` command to determine the kernel version and architecture.

```
[host03]# uname -r
3.8.13-26.1.1.el6uek.x86_64
```

   - In this example, the kernel version is `3.8.13-26.1.1.el6uek`.
   - The architecture is `x86_64`.

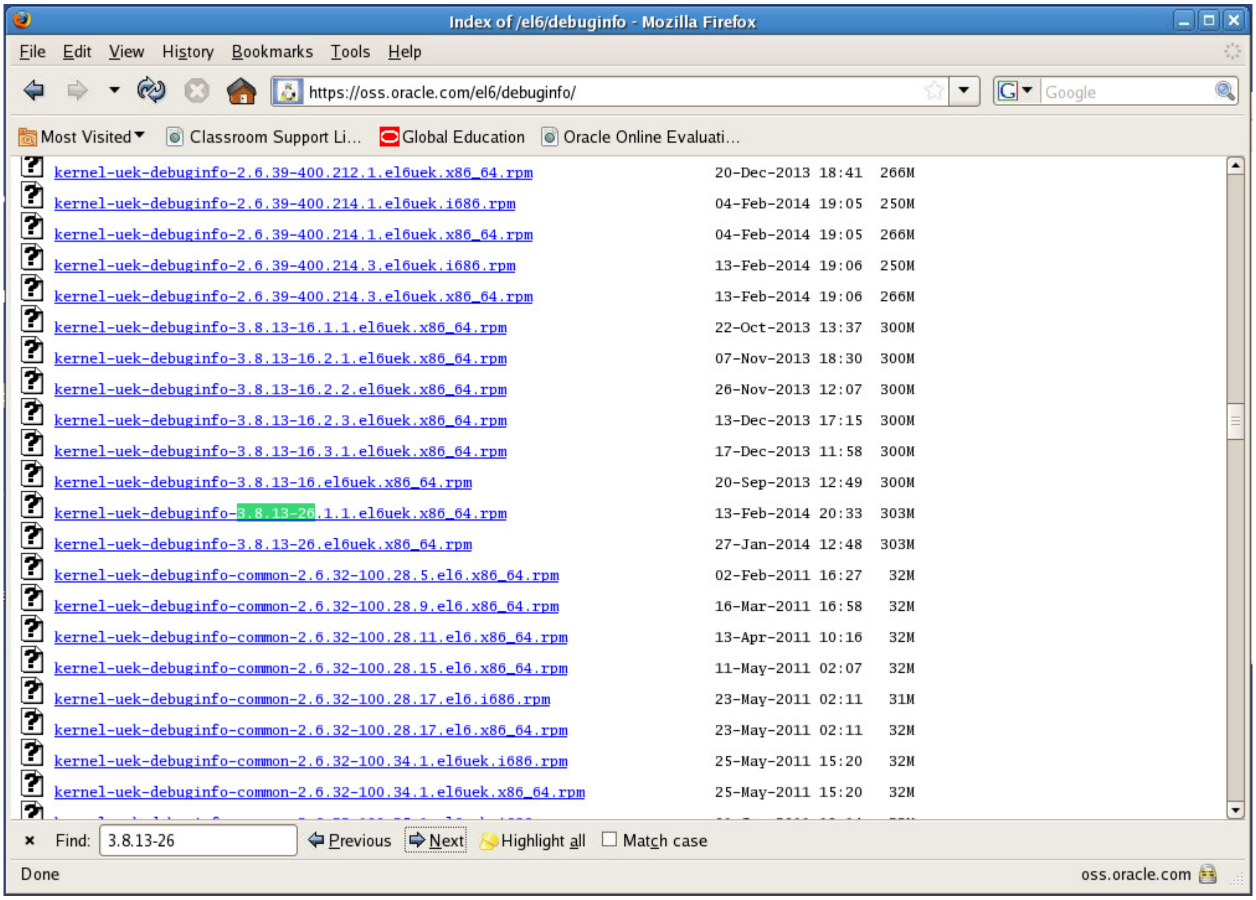
   b. If you have access only to the vmcore file, use the `strings vmcore` command.
   - In this example, the vmcore file name is `2014-0220-0525.00-host03.249.core`.
   - This file is located in the `root` user's home directory on **host03**.

```
[host03]# strings 2014-0220-0525.00-host03.249.core | less
gcc version 4.1.2 ...
...
/vmlinuz-3.8.13-26.1.1.el6uek.x86_64 ro root=/dev/mapper...
...
```

   - In this example, the kernel version is `3.8.13-26.1.1.el6uek`.
   - The architecture is `x86_64`.
   - Enter `q` to quit the `strings ... | less` command.

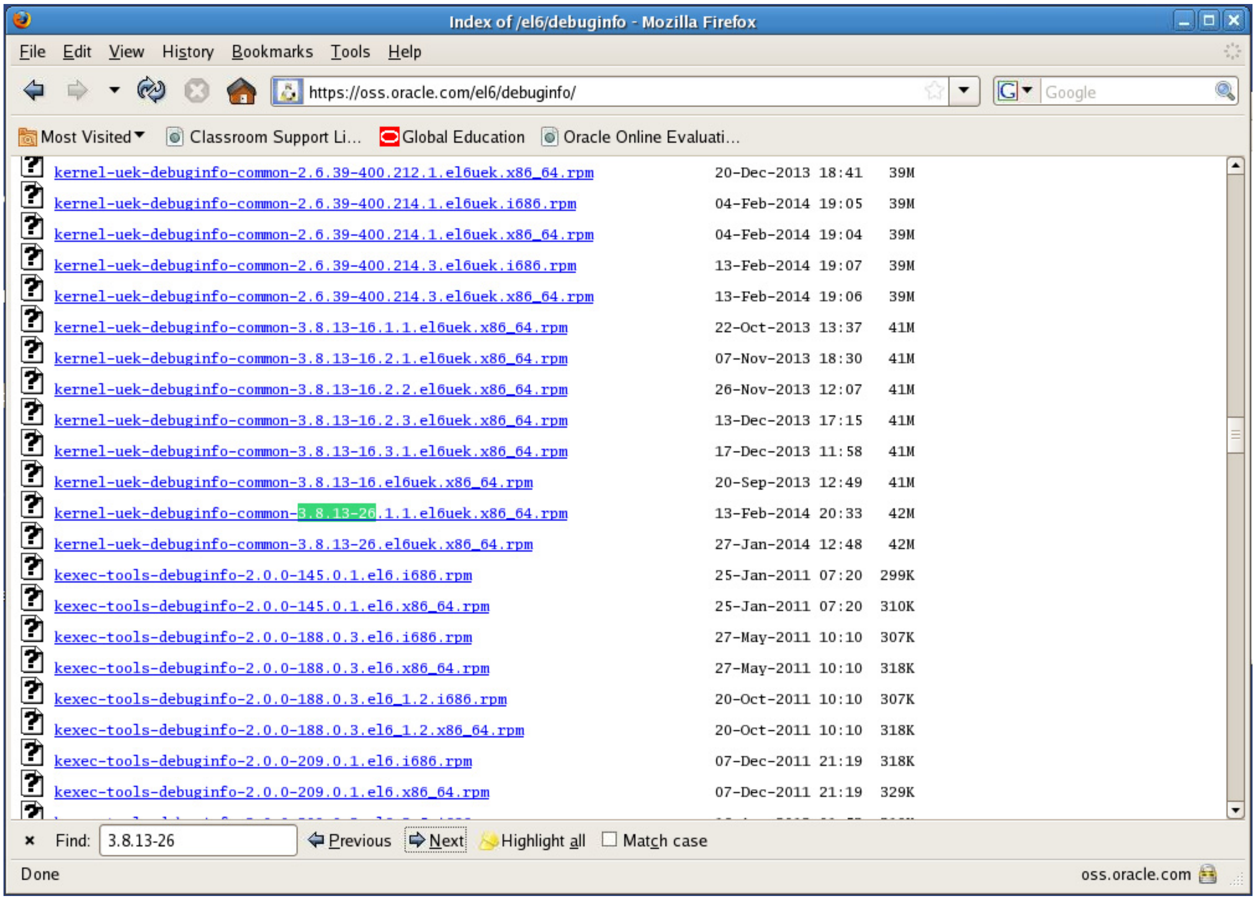
2. Download the matching kernel-debuginfo RPMs for the kernel version and architecture.
   - You do not actually download the files.

   a. Open the **Firefox Web Browser** from the **dom0** GNOME desktop.
   b. Browse to https://oss.oracle.com/el6/debuginfo/.
   c. Use the **Find** facility to search for `3.8.13-26`.
   d. Press **Next** until the `kernel-uek-debuginfo-3.8.13-26.1.1.el6uek.x86_64.rpm` file is highlighted:



- Do not download the file. It takes several minutes to download.

e. Press **Next** until the `kernel-uek-debuginfo-common-3.8.13-26.1.1.el6uek.x86_64.rpm` file is highlighted:



- Do not download the file. It takes several minutes to download.

f. Click **File > Quit** to close the **Firefox Web Browser**.

3. Use the `sftp` command to transfer the `kernel-debuginfo` packages from **dom0** to **host03**.

- The two required `kernel-debuginfo` RPMs have been downloaded to the `/OVS/seed_pool/debug` directory on **dom0**.

a. From **dom0**, use the `cd` command to change to the `/OVS/seed_pool/debug` directory.

```
[dom0]# cd /OVS/seed_pool/debug
```

b. Use the `ls` command to view the contents of the directory.

```
[dom0]# ls
kernel-uek-debuginfo-3.8.13-26.1.1.el6uek.x86_64.rpm
kernel-uek-debuginfo-common-3.8.13-26.1.1.el6uek.x86_64.rpm
```

c. Use the `sftp` command to connect to **host03**.
   - The `root` password is `oracle`.

```
[dom0]# sftp host03
Connecting to host03...
root@host03's password: oracle
sftp>
```

d. Use the `mput *` command to upload all files in the **dom0** directory to **host03**.

```
sftp> mput *
Uploading kernel-uek-debuginfo-3.8.13-26.1.1.el6uek.x86_64.rpm
to /root/...
...
sftp>
```

e. Use the `quit` command to exit `sftp`.

```
sftp> quit
```

4. From **host03**, install the `kernel-debuginfo` RPMs.

a. Use the `cd` command to ensure that you are in the `root` user's home directory. Use the `ls` command to display the `kernel-debuginfo` RPMs.

```
[host03]# cd
# ls *debuginfo*
kernel-uek-debuginfo-3.8.13-26.1.1.el6uek.x86_64.rpm
kernel-uek-debuginfo-common-3.8.13-26.1.1.el6uek.x86_64.rpm
```

b. Use the `rpm –Uvh` command to install each of the `kernel-debuginfo` RPMs.

```
# rpm –Uvh kernel-uek-debuginfo-common-3.8.13-
26.1.1.el6uek.x86_64.rpm
...
# rpm –Uvh kernel-uek-debuginfo-3.8.13-26.1.1.el6uek.x86_64.rpm
...
```

5. From **host03**, use the `yum update` command to ensure that you have the latest version of `crash` installed.

```
# yum update crash
...
```

- In this example, the latest version is already installed.

Practice 17-4: Using the `crash` Utility

Overview

In this practice, you run the following:

- The `crash` utility on a vmcore
- Various `crash` commands to analyze the state of the vmcore
- The `crash` utility on a running system
- Various `crash` commands to analyze the state of the running system

Assumptions

- You are logged in as `root` on **host03.**

Tasks

1. Determine the location of the `vmlinux` file in the `kernel-uek-debuginfo` RPM.
   a. Use the `rpm –ql` command to list the files in an RPM package. Pipe the output to the `grep` command to search for `vmlinux`.

```
# rpm –ql kernel-uek-debuginfo | grep vmlinux
/usr/lib/debug/lib/modules/3.8.13-26.1.1.el6uek.x86_64/vmlinux
```

   - This file is provided as a command-line argument to the `crash` command.

2. Use the `ls` command to list the vmcore file name. Ensure that you are in the `root` user's home directory.

```
# cd
# ls *core
2014-0220-0525.00-host03.249.core
```

   - The file name for your vmcore is different.

3. Use the `crash` command to initialize the core dump analysis session.
   - In this example, the core dump file name is `2014-0220-0525.00-host03.249.core`. Substitute the correct vmcore file name as determined in the previous task.
   - A sample output is displayed:

```
# crash 2014-0220-0525.00-host03.249.core
/usr/lib/debug/lib/modules/3.8.13-26.1.1.el6uek.x86_64/vmlinux
crash 6.1.0-5.0.1.el6
...
      KERNEL: /usr/lib/debug/lib/modules/3.8.13-
26.1.1.el6uek.x86_64/vmlinux
    DUMPFILE: 2014-0220-0525.00-host03.249.core
        CPUS: 1
        DATE: Thu Feb 20 ...
      UPTIME: 11:19:18
LOAD AVERAGE: 0.00, 0.01, 0.05
       TASKS: 215
    NODENAME: host03.example.com
```

```
      RELEASE: 3.8.13-26.1.1.el6uek.x86_64
      VERSION: #2 SMP Thu Feb 13 ...
      MACHINE: x86_64 (2992 Mhz)
       MEMORY: 1.5 GB
        PANIC: ""
          PID: 0
      COMMAND: "swapper"
         TASK: ffffffff818ae420 [THREAD_INFO: ffffffff8189a000]
          CPU: 0
        STATE: TASK_RUNNING (ACTIVE)
      WARNING: panic task not found
crash>
```

4. From the `crash>` prompt, use the following `crash` commands to analyze the Linux crash data. Use `help <command>` for each `crash` command to display usage, options, and examples.
   - No sample output is provided in this task because output varies.
   a. `?` – Display the `crash` commands.
   b. `bt` – Display kernel stack backtrace.
   c. `bt -l` – Display the source code line numbers.
   d. `dev` – Display character and block device data.
   e. `dev -d` – Display disk I/O statistics.
   f. `kmem -i` – Display general memory usage information.
   g. `log` – Display the kernel message buffer.
      - This is the same data displayed with the `dmesg` command.
   h. `mach` – Display machine-specific data.
   i. `mach -c` – Display the cpuinfo structure.
   j. `mod` – Display the currently installed kernel modules.
   k. `mount` – Display information about currently mounted file systems.
   l. `net` – Display network-related data.
   m. `ps` – Display process status information.
   n. `ps -u` – Display only user tasks.
   o. `history` – Display a history of executed commands.
      - There is no help available for the `history` command.
   p. `quit` – Exit the `crash` utility.
5. Use the `crash` command with no arguments to analyze the state of the running system.
   - Only a partial sample output is shown:

```
# crash
crash 6.1.0-5.0.1.el6
...
crash: trying /proc/kcore as an alternative to /dev/mem
```

```
      KERNEL: /usr/lib/debug/lib/modules/3.8.13-
26.1.1.el6uek.x86_64/vmlinux
    DUMPFILE: /proc/kcore
        CPUS: 1
...
         PID: 4615
     COMMAND: "crash"
...
```

- Notice that the `DUMPFILE` is different and the `COMMAND` is "`crash`".

6. From the `crash>` prompt, use the following `crash` commands to analyze the running Linux system. Use `help <`*`command`*`>` for each `crash` command to display usage, options, and examples.
   - No sample output is provided in this task because the output varies.
   a. `files` – Display information about open files.
   b. `foreach files -R /dev/pts/1` – Search for references to `/dev/pts/1`.
      - If none found, replace `/dev/pts/1` with `/dev/pts/0`.
   c. `fuser /dev/pts/1` – Display tasks using `/dev/pts/1`.
      - If none found, replace `/dev/pts/1` with `/dev/pts/0`.
   d. `sig` – Display a task's signal-handling data.
   e. `sys` – Display system information.
      - This is the same information displayed during `crash` initialization.
   f. `sys -c` – Display the system call table entries.
   g. `task` – Display the contents of `task_struct`.
      - Each `task_struct` data structure describes a process or task in the system.
   h. `vm` – Display a task's virtual memory data.
   i. `quit` – Exit the `crash` utility.

Practices for Lesson 18: Dynamic Tracing with DTrace

Chapter 18

Practices for Lesson 18: Dynamic Tracing with DTrace

Practices Overview

In these practices, you perform the following:

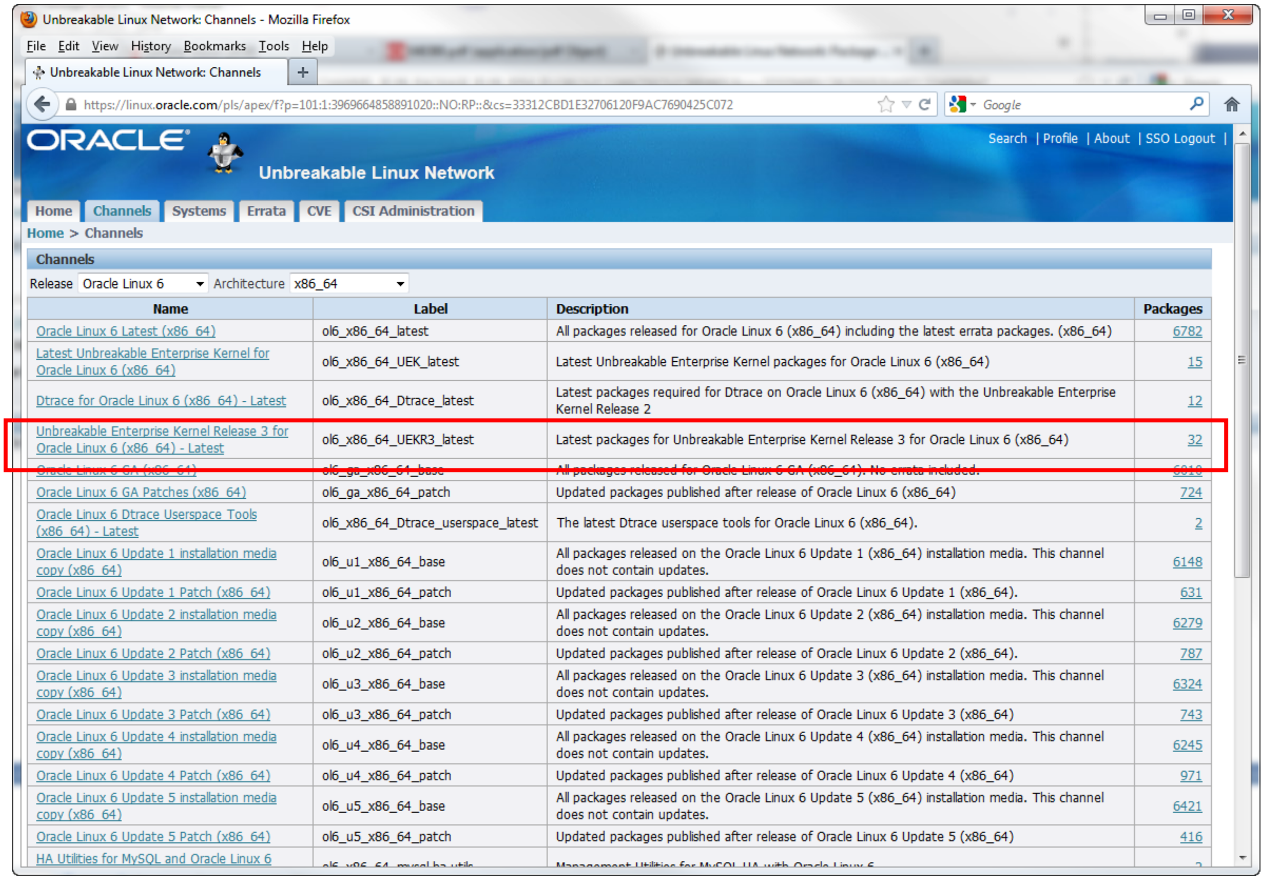
- Enable DTrace.
- Use DTrace from the command line.
- Use DTrace D scripts.

Practice 18-1: Using `sftp` to Upload DTrace Packages

Overview

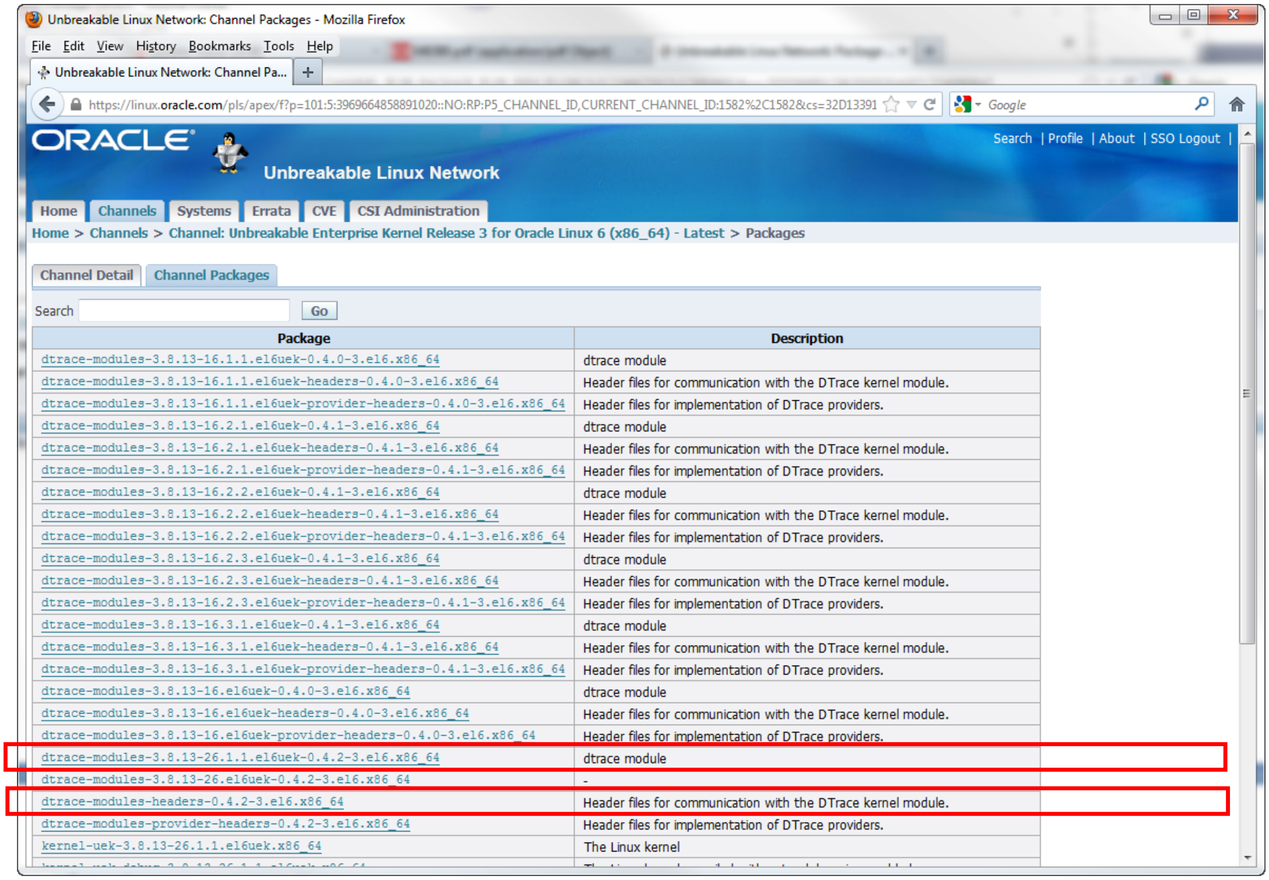
In this practice, you perform the following:

- Use `sftp` to upload the DTrace packages from **dom0** to the **host03** VM.
- These packages were downloaded to **dom0** from two different ULN channels.
- The first ULN channel, `ol6_x86_64_UEKR3_latest`, is highlighted as follows:

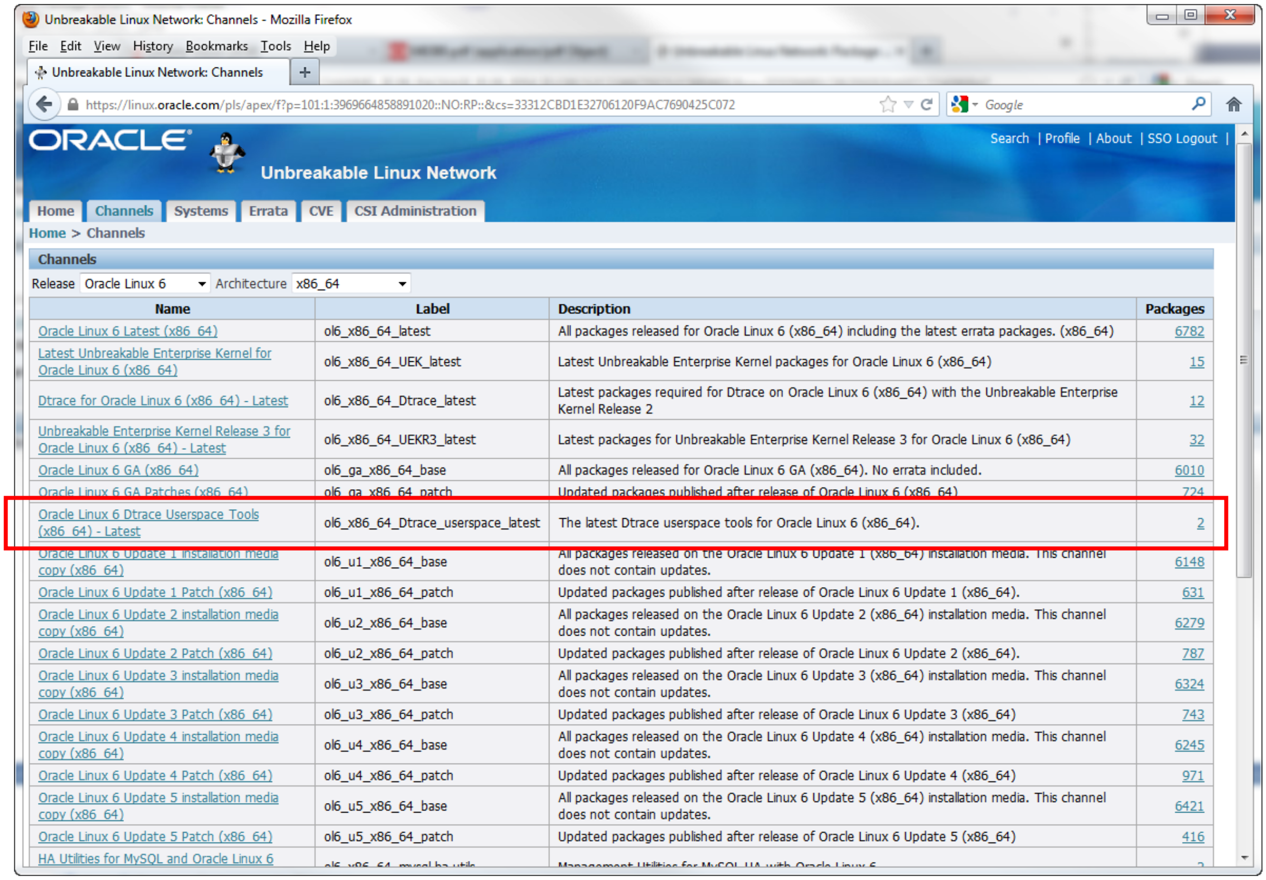


Name	Label	Description	Packages
Oracle Linux 6 Latest (x86_64)	ol6_x86_64_latest	All packages released for Oracle Linux 6 (x86_64) including the latest errata packages. (x86_64)	6782
Latest Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)	ol6_x86_64_UEK_latest	Latest Unbreakable Enterprise Kernel packages for Oracle Linux 6 (x86_64)	15
Dtrace for Oracle Linux 6 (x86_64) - Latest	ol6_x86_64_Dtrace_latest	Latest packages required for Dtrace on Oracle Linux 6 (x86_64) with the Unbreakable Enterprise Kernel Release 2	12
Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64) - Latest	ol6_x86_64_UEKR3_latest	Latest packages for Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64)	32
Oracle Linux 6 GA (x86_64)	ol6_ga_x86_64_base	All packages released for Oracle Linux 6 GA (x86_64). No errata included.	6018
Oracle Linux 6 GA Patches (x86_64)	ol6_ga_x86_64_patch	Updated packages published after release of Oracle Linux 6 (x86_64)	724
Oracle Linux 6 Dtrace Userspace Tools (x86_64) - Latest	ol6_x86_64_Dtrace_userspace_latest	The latest Dtrace userspace tools for Oracle Linux 6 (x86_64).	2
Oracle Linux 6 Update 1 installation media copy (x86_64)	ol6_u1_x86_64_base	All packages released on the Oracle Linux 6 Update 1 (x86_64) installation media. This channel does not contain updates.	6148
Oracle Linux 6 Update 1 Patch (x86_64)	ol6_u1_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 1 (x86_64).	631
Oracle Linux 6 Update 2 installation media copy (x86_64)	ol6_u2_x86_64_base	All packages released on the Oracle Linux 6 Update 2 (x86_64) installation media. This channel does not contain updates.	6279
Oracle Linux 6 Update 2 Patch (x86_64)	ol6_u2_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 2 (x86_64).	787
Oracle Linux 6 Update 3 installation media copy (x86_64)	ol6_u3_x86_64_base	All packages released on the Oracle Linux 6 Update 3 (x86_64) installation media. This channel does not contain updates.	6324
Oracle Linux 6 Update 3 Patch (x86_64)	ol6_u3_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 3 (x86_64)	743
Oracle Linux 6 Update 4 installation media copy (x86_64)	ol6_u4_x86_64_base	All packages released on the Oracle Linux 6 Update 4 (x86_64) installation media. This channel does not contain updates.	6245
Oracle Linux 6 Update 4 Patch (x86_64)	ol6_u4_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 4 (x86_64)	971
Oracle Linux 6 Update 5 installation media copy (x86_64)	ol6_u5_x86_64_base	All packages released on the Oracle Linux 6 Update 5 (x86_64) installation media. This channel does not contain updates.	6421
Oracle Linux 6 Update 5 Patch (x86_64)	ol6_u5_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 5 (x86_64)	416
HA Utilities for MySQL and Oracle Linux 6	ol6_x86_64_mysql_ha_utils	Management Utilities for MySQL HA with Oracle Linux 6	2

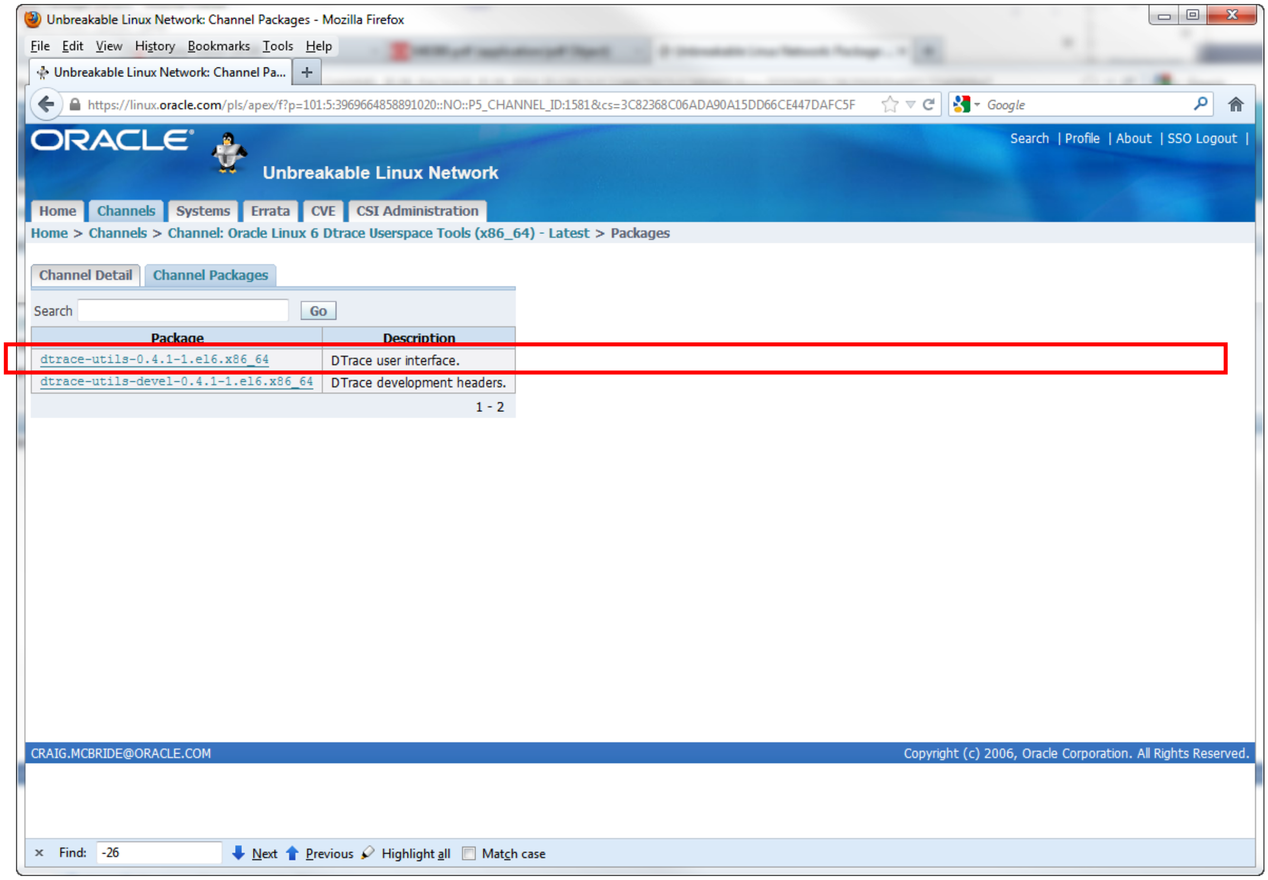
- The 32 packages from the ULN channel `ol6_x86_64_UEKR3_latest` are shown as follows:



- The first highlighted package from this channel, `dtrace-modules-3.8.13-26.1.1.el6uek-0.4.2-3.el6.x86_64.rpm`, has been downloaded to **dom0**.
- The second highlighted package from this channel, `dtrace-modules-headers-0.4.2-3.el6.x86_64.rpm`, is a required dependency package which has also been downloaded to **dom0**.
- The second ULN channel, `ol6_x86_64_Dtrace_userspace_latest`, is highlighted as follows:



- The two packages from the ULN channel `ol6_x86_64_Dtrace_userspace_latest` are shown as follows:



- The highlighted package from this channel, `dtrace-utils-0.4.1-1.el6.x86_64.rpm`, has been downloaded to **dom0**.

Assumptions

- You are logged in as `root` on **dom0.**
- You are logged in as `root` on **host03.**
- The DTrace packages from the ULN have been downloaded to the `root` user's home directory on **dom0**.

Tasks

1. Use the `sftp` command to transfer the DTrace packages from **dom0** to **host03**.
   a. From **dom0**, use the `cd` command to change to the `/OVS/seed_pool/dtrace_rpms` directory.

```
[dom0]# cd /OVS/seed_pool/dtrace_rpms
```

   b. Use the `ls` command to view the contents of the directory.

```
[dom0]# ls
dtrace-modules-3.8.13-26.1.1.el6uek-0.4.2-3.el6.x86_64.rpm
dtrace-modules-headers-0.4.2-3.el6.x86_64.rpm
dtrace-utils-0.4.1-1.el6.x86_64.rpm
```

c. Use the `sftp` command to connect to **host03** as `root`.
   - The password is `oracle`.

```
[dom0]# sftp host03
Connecting to host03...
root@host03's password: oracle
sftp>
```

d. Use the `mput *` command to upload all files on **dom0** to **host03**.

```
sftp> mput *
Uploading dtrace-modules-3.8.13-26.1.1.el6uek-0.4.2-3.el6.x86_64.rpm to /root/...
Uploading dtrace-modules-headers-0.4.2-3.el6.x86_64.rpm to /root/...
Uploading dtrace-utils-0.4.1-1.el6.x86_64.rpm to /root/...
...
sftp>
```

e. Use the `quit` command to exit `sftp`.

```
sftp> quit
```

Practice 18-2: Installing the DTrace Packages

Overview

In this practice, you perform the following:

- Install the DTrace packages.
- Load the DTrace modules into the kernel.
- View the DTrace release notes.

Assumptions

- You are the `root` user on **host03**.

Tasks

1. Install the DTrace packages and dependency packages.
   a. As the `root` user on **host03**, use the `ls` command to view the contents of the `root` user's home directory.
      - Use the `cd` command to ensure that you are in the root user's home directory.

```
# cd
# ls
...
dtrace-modules-3.8.13-26.1.1.el6uek-0.4.2-3.el6.x86_64.rpm
dtrace-modules-headers-0.4.2-3.el6.x86_64.rpm
dtrace-utils-0.4.1-1.el6.x86_64.rpm
...
```

      - The directory contains the DTrace packages along with some files that were created during the OS installation and other files that were used in previous practices.

   b. Use the `rpm -Uvh` command to install each of the DTrace packages.
      - If your system was registered with the following ULN channels, you could use the `yum` command to update to the UEK R3 and to install the DTrace packages.
      - `ol6_x86_64_latest`
      - `ol6_x86_64_UEKR3_latest`
      - `ol6_x86_64_Dtrace_userspace_latest`

```
# rpm –Uvh dtrace-modules-headers-0.4.2-3.el6.x86_64.rpm
...
# rpm –Uvh dtrace-utils-0.4.1-1.el6.x86_64.rpm
...
# rpm –Uvh dtrace-modules-3.8.13-26.1.1.el6uek-0.4.2-
3.el6.x86_64.rpm
...
```

2. Load the DTrace modules into the running kernel.
   a. Use the `cd` command to change to the `/lib/modules/3.8.13-26.1.1.el6uek.x86_64/kernel/drivers/dtrace` directory.

```
# cd /lib/modules/3.8.13-
26.1.1.el6uek.x86_64/kernel/drivers/dtrace
```

b. Use the `ls -l` command to view the contents of the directory.

```
# ls -l
-rw-r--r--.  ...  dt_perf.ko
-rw-r--r--.  ...  dtrace.ko
-rw-r--r--.  ...  dt_test.ko
-rw-r--r--.  ...  fasttrap.ko
-rw-r--r--.  ...  profile.ko
-rw-r--r--.  ...  sdt.ko
-rw-r--r--.  ...  systrace.ko
```

c. Use the `modprobe` command to load the necessary modules.

- Do not load the `dt_*` modules; they are for development purposes only.
- It is not necessary to load the `dtrace` module. Loading any of the other modules automatically loads the `dtrace` module.
- The following set of commands shows that the `dtrace` module is loaded by loading the `fasttrap` module.
- The last `lsmod` command shows that the `dtrace` module is used by the `systrace`, `sdt`, `profile`, and `fasttrap` modules.

```
# modprobe fasttrap
# lsmod | grep dtrace
dtrace   136502  1  fasttrap
ctf         941  1  dtrace
# modprobe profile
# modprobe sdt
# modprobe systrace
# lsmod | grep systrace
systrace   4288  0
dtrace   136502  5  systrace,sdt,profile,fasttrap
```

3. View the DTrace release notes.

a. Use the `cd` command to change to the `/usr/share/doc/dtrace-0.4.1` directory.

```
# cd /usr/share/doc/dtrace-0.4.1
```

b. Use the `ls` command to display the contents of the directory.

```
# ls
INCOMPATIBILITIES  NEWS  README  showUSDT
```

c. Use the `less` command to view the `INCOMPATIBILITIES` file in the `/usr/share/doc/dtrace-0.4.1` directory.

```
# less INCOMPATIBILITIES
This file documents known incompatibilities between Linux and
Solaris dtrace, together with the difficulty of overcoming them,
and the likelihood that they will be overcome.


Missing providers
```

```
-----------------
Difficulty: Medium
Likelihood: High

A number of providers are missing, including pid, fbt, and net.
...
```

d. Use the `less` command to view the `NEWS` file in the `/usr/share/doc/dtrace-0.4.1` directory.

```
# less NEWS
0.4.1
-----

New features:

 - New development tools showUSDT (for dumping of DOF sections)
and ctf_module_dump (for dumping of CTF in kernel modules).
...
```

e. Use the `less` command to view the `README` file in the `/usr/share/doc/dtrace-0.4.1` directory.

```
# less README
Linux DTrace v0.3 for x86_64 Oracle Unbreakable Enterprise
Kernel 2.6.39
...
```

- In this example, the `README` file has not been updated for DTrace v0.4 and the UEK R3 (`3.8.13`).

f. Use the `cd` command to change to the `/usr/share/doc/dtrace-modules-3.8.13-26.1.1.el6uek` directory.

```
# cd /usr/share/doc/dtrace-modules-3.8.13-26.1.1.el6uek
```

g. Use the `ls` command to display the contents of the directory.

```
# ls
NEWS
```

h. Use the `less` command to view the `NEWS` file.

```
# less NEWS
DTrace Kernel Modules News
==========================

Release 0.4.2 (Dec 20th, 2013)
------------------------------
Kernel release: 3.8.13-22.el6uek

Changes:
```

```
 - SDT probe points in kernel modules are now supported.
...
Release 0.4.1 (Nov 6th, 2013)
----------------------------
Kernel release: 3.8.13-16.2.1.el6uek

Changes:
...
```

- Notice that this file contains new features, changes, and bug fixes for each release.

Practice 18-3: Using DTrace from the Command Line

Overview

In this practice, you perform the following:

- List DTrace probes for selected providers.
- Enable DTrace probes.
- Enable probes and specify actions.
- Execute specific `dtrace` commands and observe the output.

Assumptions

- You are the `root` on **host03.**

Tasks

1. List DTrace probes.
   a. Use the `dtrace -l` command to list DTrace probes for all providers.

```
# dtrace -l
ID  PROVIDER   MODULE     FUNCTION NAME
1   dtrace                         BEGIN
2   dtrace                         END
3   dtrace                         ERROR
4   syscall    vmlinux        read entry
5   syscall    vmlinux        read return
...
639 profile                        tick-1000
640 profile                        tick-5000
```

   b. Use the `dtrace -l -P dtrace` command to list the probes for the `dtrace` provider.

```
# dtrace -l -P dtrace
ID   PROVIDER   MODULE     FUNCTION NAME
1      dtrace                       BEGIN
2      dtrace                       END
3      dtrace                       ERROR
```

   c. Use the `dtrace -l -P syscall` command to list the probes for the `syscall` provider.

```
# dtrace -l -P syscall
ID   PROVIDER   MODULE     FUNCTION NAME
4     syscall   vmlinux        read entry
5     syscall   vmlinux        read return
6     syscall   vmlinux       write entry
7     syscall   vmlinux       write return
8     syscall   vmlinux        open entry
...
602   syscall   vmlinux      waitfd entry
603   syscall   vmlinux      waitfd return
```

d. Use the `dtrace -l -P sched` command to list the probes for the `sched` provider.

```
# dtrace -l -P sched
 ID   PROVIDER    MODULE                   FUNCTION NAME
604      sched   vmlinux                 __schedule remain-cpu
605      sched   vmlinux                 __schedule sleep
606      sched   vmlinux                 __schedule preempt
607      sched   vmlinux                 __schedule off-cpu
612      sched   vmlinux               dequeue_task dequeue
623      sched   vmlinux               enqueue_task enqueue
624      sched   vmlinux         finish_task_switch on-cpu
629      sched   vmlinux              set_user_nice change-pri
631      sched   vmlinux            sys_sched_yield surrender
632      sched   vmlinux             try_to_wake_up wakeup
633      sched   vmlinux       update_process_times tick
634      sched   vmlinux                   yield_to surrender
```

e. Use the `dtrace -l -P sdt` command to list the probes for the `sdt` provider.

- The SDT probes that are defined for Oracle Linux kernel are likely to change over time.

```
# dtrace -l -P sdt
 ID PROVIDER      MODULE                   FUNCTION NAME
621       sdt    vmlinux            dtrace_sdt_perf measure
626       sdt    vmlinux             psinfo_cleaner test
```

f. Use the `dtrace -l -P io` command to list the probes for the `io` provider.

```
# dtrace -l -P io
 ID PROVIDER      MODULE                   FUNCTION NAME
610        io    vmlinux             __wait_on_buffer wait-done
611        io    vmlinux             __wait_on_buffer wait-start
622        io    vmlinux          end_bio_bh_io_sync done
630        io    vmlinux                   submit_bh start
```

g. Use the `dtrace -l -P proc` command to list the probes for the `proc` provider.

```
# dtrace -l -P proc
 ID PROVIDER    MODULE                 FUNCTION NAME
608      proc  vmlinux            __send_signal signal-discard
609      proc  vmlinux            __send_signal signal-send
613      proc  vmlinux        do_execve_common exec-success
614      proc  vmlinux        do_execve_common exec
615      proc  vmlinux        do_execve_common exec-failure
616      proc  vmlinux                 do_exit exit
617      proc  vmlinux                 do_exit lwp-exit
618      proc  vmlinux                 do_fork create
619      proc  vmlinux                 do_fork lwp-create
```

```
620     proc  vmlinux       do_sigtimedwait signal-clear
625     proc  vmlinux get_signal_to_deliver signal-handle
627     proc  vmlinux         schedule_tail lwp-start
628     proc  vmlinux         schedule_tail start
```

h. Use the `dtrace -l -P profile` command to list the probes for the `profile` provider.

```
# dtrace -l -P profile
 ID PROVIDER   MODULE              FUNCTION NAME
635  profile                                tick-1
636  profile                                tick-10
637  profile                                tick-100
638  profile                                tick-500
639  profile                                tick-1000
640  profile                                tick-5000
```

2. Enable DTrace probes.
   - Enable probes by specifying them without the `-l` (list) option.

a. Use the `dtrace -P dtrace` command to enable all probes for the `dtrace` provider.
   - Press Ctrl + C to return to the prompt.

```
# dtrace -P dtrace
dtrace: description 'dtrace' matched 3 probes
CPU       ID         FUNCTION:NAME
0          1                :BEGIN
^C
0          2                :END
```

   - Notice that the `dtrace:::BEGIN` probe is enabled and displayed.
   - The `dtrace:::END` probe was not displayed until after you pressed Ctrl + C.
   - No action was specified; therefore, the default action was taken, which is to only indicate that the probe fired.

b. Use the `dtrace -n dtrace:::BEGIN` command to enable only the `dtrace:::BEGIN` probe for the `dtrace` provider.
   - The `-n` option specifies to enable the probe by name.
   - Press Ctrl + C to return to the prompt.

```
# dtrace -n dtrace:::BEGIN
dtrace: description 'dtrace:::BEGIN' matched 1 probe
CPU       ID         FUNCTION:NAME
0          1                :BEGIN
^C
```

   - Notice that only one probe matched—the probe that was specified by name.

c. Use the `dtrace -n syscall::open:` command to enable all `open` function probes for the `syscall` provider.
   - Wait for a few probes to fire before pressing Ctrl + C to return to the prompt.

```
# dtrace -n syscall::open:
dtrace: description 'syscall::open:' matched 2 probes
CPU       ID         FUNCTION:NAME
0          8             open:entry
0          9             open:return
0          8             open:entry
0          9             open:return
^C
```

   - Notice that two probes matched—the entry to and return from the `open` system call.

3. Enable probes and specify actions.

a. Use the `dtrace -n 'dtrace:::BEGIN {trace("hello, world");}'` command to enable the `dtrace:::BEGIN` probe with the action, `trace("hello, world")`.
   - Notice that the action is surrounded by `{}`.
   - Notice that the entire string, beginning with the probe to be enabled and the associated action to take, is surrounded by `''`.
   - Press Ctrl + C to return to the prompt.

```
# dtrace -n 'dtrace:::BEGIN {trace("hello, world");}'
dtrace: description 'dtrace:::BEGIN' matched 1 probe
CPU       ID         FUNCTION:NAME
0          1                :BEGIN  hello, world
^C
```

   - Notice the "`hello,world`" output, which is the result of the specified `trace()` action.

b. Add a second action, `exit(0)`, to the previous command.

```
# dtrace -n 'dtrace:::BEGIN {trace("hello, world");exit(0);}'
dtrace: description 'dtrace:::BEGIN' matched 1 probe
CPU       ID         FUNCTION:NAME
0          1                :BEGIN  hello, world
```

   - Notice that you did not need to press Ctrl + C to return to the prompt.
   - The action `exit(0)` caused the command to exit.
   - Notice that each action is terminated with a semicolon (`;`) character.

c. Add the `-q` (quiet) option to the previous command.

```
# dtrace –q -n 'dtrace:::BEGIN {trace("hello, world");exit(0);}'
hello, world
```

   - Notice that the quiet option suppresses the default output of displaying the CPU where the probe fired, the unique probe ID, the function where the probe fired (if any), and the probe name.

4. Open a second window from **dom0** and log in to **host03** as the `root` user.
   a. From the second window, use the `su -` command to become the `root` user on **dom0** and then `ssh` to **host03**.
      - The `root` password is `oracle` on **dom0** and on **host03**.

```
[dom0]$ su -
Password: oracle
# ssh host03
root@host03's password: oracle
```

5. Execute specific `dtrace` commands.
   - No sample output is provided for the following `dtrace` commands.
   - Enter each of the examples from the command line and observe the output.

   a. Trace the time of entry for each system call.
      - Press Ctrl + C to exit the `dtrace` command.

```
# dtrace -n 'syscall:::entry {trace(timestamp)}'
...
^C
```

   b. Display commands that are executing on your system.
      - Do not immediately press Ctrl + C to exit the `dtrace` command.
      - You run a command in the second window to generate a `dtrace` output.

```
# dtrace -n 'proc:::exec {trace(stringof(arg0));}'
...
```

      - From the second window on **host03**, run the `man ls` command and observe the output from the `dtrace` command in the first window.

```
# man ls
...
```

      - After viewing the output from the `dtrace` command in the first window, press Ctrl + C to kill the `dtrace` command.

```
# dtrace -n 'proc:::exec {trace(stringof(arg0));}'
...
^C
```

      - Enter `q` to terminate the `man ls` command in the second window.

```
# man ls
...
q
```

c. Display new processes with time stamps and arguments.
   - Do not immediately press Ctrl + C to exit the `dtrace` command.
   - You run a command in the second window to generate a `dtrace` output.

```
# dtrace -qn 'proc:::exec-success {printf("%d %s\n", timestamp,
curpsinfo->pr_psargs);}'
...
```

   - From the second window on **host03**, run the `man ls` command and observe the output from the `dtrace` command in the first window.

```
# man ls
...
```

   - After viewing the output from the `dtrace` command in the first window, press Ctrl + C to kill the `dtrace` command.

```
# dtrace -qn 'proc:::exec-success {printf("%d %s\n", timestamp,
curpsinfo->pr_psargs);}'
...
^C
```

   - Enter `q` to terminate the `man ls` command in the second window.

```
# man ls
...
q
```

d. Display files opened by process.
   - Do not immediately press Ctrl + C to exit the `dtrace` command.
   - You run a command in the second window to generate a `dtrace` output.

```
# dtrace -qn 'syscall::open*:entry {printf("%s %s\n",
execname,copyinstr(arg0));}'
...
```

   - From the second window on **host03**, run the `man ls` command and observe the output from the `dtrace` command in the first window.

```
# man ls
...
```

   - After viewing the output from the `dtrace` command in the first window, press Ctrl + C to kill the `dtrace` command.

```
# dtrace -qn 'syscall::open*:entry {printf("%s %s\n",
execname,copyinstr(arg0));}'
...
^C
```

- Enter `q` to terminate the `man ls` command in the second window.

```
# man ls
...
q
```

e. Display the number of system calls by system call using aggregations.
   - When using aggregations (`@`), no output is displayed until after you press Ctrl + C.
   - Do not immediately press Ctrl + C to exit the `dtrace` command.
   - You run a command in the second window to generate a `dtrace` output.

```
# dtrace -qn 'syscall:::entry {@num[probefunc] = count();}'
```

   - From the second window on **host03**, run the `man ls` command and observe the output from the `dtrace` command in the first window.

```
# man ls
...
```

   - Press Ctrl + C to kill the `dtrace` command and display the aggregated totals.

```
# dtrace -qn 'syscall:::entry {@num[probefunc] = count();}'
^C
...
```

   - Enter `q` to terminate the `man ls` command in the second window.

```
# man ls
...
q
```

f. Count the number of `write()` system calls and the number of `read()` system calls invoked by processes until you press Ctrl + C.
   - Do not immediately press Ctrl + C to exit the `dtrace` command.
   - You run a command in the second window to generate a `dtrace` output.

```
# dtrace -qn 'syscall::write:entry,syscall::read:entry
{@[strjoin(probefunc,"() calls")] = count();}'
```

   - From the second window on **host03**, run the `man ls` command and observe the output from the `dtrace` command in the first window.

```
# man ls
...
```

   - Press Ctrl + C to kill the `dtrace` command and display the aggregated totals.

```
# dtrace -qn 'syscall::write:entry,syscall::read:entry
{@[strjoin(probefunc,"() calls")] = count();}'
^C
...
```

- Enter `q` to terminate the `man ls` command in the second window.

```
# man ls
...
q
```

g. Aggregate the number of `read()` system calls but use a predicate to exclude the `read()` system calls initiated by the `dtrace` process.

- Do not immediately press Ctrl + C to exit the `dtrace` command.
- You run a command in the second window to generate a `dtrace` output.

```
# dtrace -qn 'syscall::read:entry /execname != "dtrace"/ {
@reads[execname, fds[arg0].fi_pathname] = count();}'
```

- The predicate in this example is `/execname != "dtrace"/`.
- Predicates are always enclosed in `//`.
- From the second window on **host03**, run the `man ls` command and observe the output from the `dtrace` command in the first window.

```
# man ls
...
```

- Press Ctrl + C to kill the `dtrace` command and display the aggregated totals.

```
# dtrace -qn 'syscall::read:entry /execname != "dtrace"/ {
@reads[execname, fds[arg0].fi_pathname] = count();}'
^C
...
```

- Enter `q` to terminate the `man ls` command in the second window.

```
# man ls
...
q
```

Practice 18-4: Creating and Running D Scripts

Overview

In this practice, you create and run D scripts.

Assumptions

- You are the `root` on **host03**.

Tasks

1. Create and run a D script.
   a. Use the `vi` editor to create the following `rdbufsize.d` file.
      - This shows quantized results for the buffer sizes used in `read()` syscalls.
      - It gives a statistical breakdown of the sizes passed in the `read()` syscall.
      - This can be useful to see what buffer sizes are commonly used.

```
# vi rdbufsize.d
syscall::read:entry
{
    @[“read”] = quantize(arg2);
}
```

   b. Use the `dtrace -s` command to execute the `rdbufsize.d` D script.

```
# dtrace –s rdbufsize.d
^C
...
```

2. Create and run a D script.
   a. Use the `vi` editor to create the following `syscall.d` file:
      - This script counts system calls over a 10-second period.

```
# vi syscall.d
syscall:::
{
    @[probefunc] = count();
}
tick-1s
/i++ >= 10/
{
    exit(0);
}
```

b. Use the `dtrace -s` command to execute the `syscall.d` D script.

- This script takes a few seconds to complete. Do not press Ctrl + C.

```
# dtrace –s syscall.d
...
```

3. Create and run a D script.

a. Use the `vi` editor to create the following `rtime.d` file.

- This script records the times that processes spend invoking `read()` calls.

```
# vi rtime.d
syscall::read:entry
{
    self->t = timestamp;
}
syscall::read:return
/self->t != 0/
{
    printf("%s spent %d nsecs in read()\n",execname,timestamp –
self->t);
    self->t = 0;
}
```

b. Use the `dtrace -s` command to execute the `rtime.d` D script.

- Press Ctrl + C to exit the script after a few lines of output.

```
# dtrace –s rtime.d
...
^C
```

4. Create and run a D script.

a. Use the `vi` editor to create the following `io1.d` file.

- This script displays the details of block I/O events.

```
# vi io1.d
io:::done,
io:::start,
io:::wait-done,
io:::wait-start
{
    printf("%8s %10s: %d %16s (%s size %d @ sect %d)\n",
           args[1]->dev_statname,
           probename,
           timestamp & 1000000000,
           execname,
           args[0]->b_flags & B_READ ? "R" :
           args[0]->b_flags & B_WRITE ? "W" : "?",
           args[0]->b_bcount,
           args[0]->b_blkno);
}
```

b. Use the `dtrace -s` command to execute the `io1.d` D script.

- Press Ctrl + C to exit the script.

```
# dtrace –s io1.d
...
^C
```

5. Log off from **host03**.

a. Use the `exit` command to log off from **host03**.

```
# exit
logout
Connection to host03 closed.
```

b. Repeat step 5a to log off from **host03** from the second terminal window.

Appendixes: Remote Access Options

Chapter 19

Appendix

Appendixes Overview

The three appendixes show various options for accessing your student PC remotely:

- Appendix A: Using NX Client to Connect to **dom0**
- Appendix B: Using NX Player to Connect to **dom0**
- Appendix C: Using VNC (TightVNC) to Connect Directly to VM Guests
- Appendix D: Using NoMachine Version 4 to Connect to **dom0**

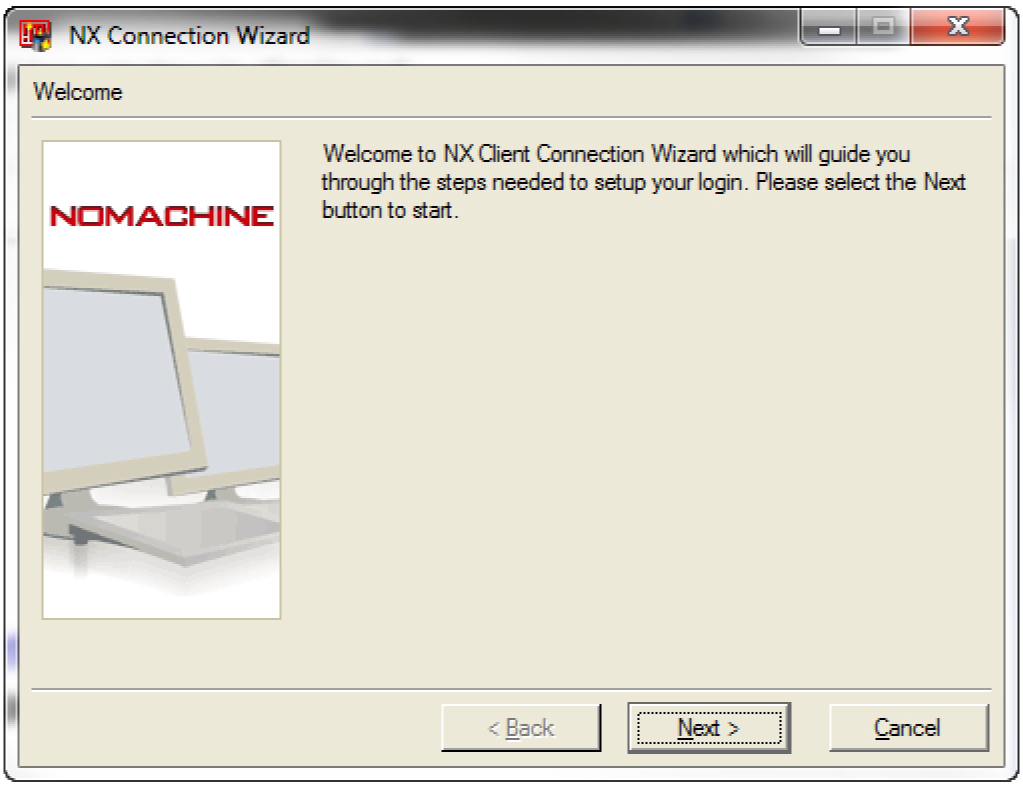
Appendix A: Using NX Client to Connect to dom0

Overview

This appendix discusses accessing your student PC (**dom0**) remotely by using NX Client. The NX Client in this appendix is NX Client for Windows, Version 3.5.0-9.

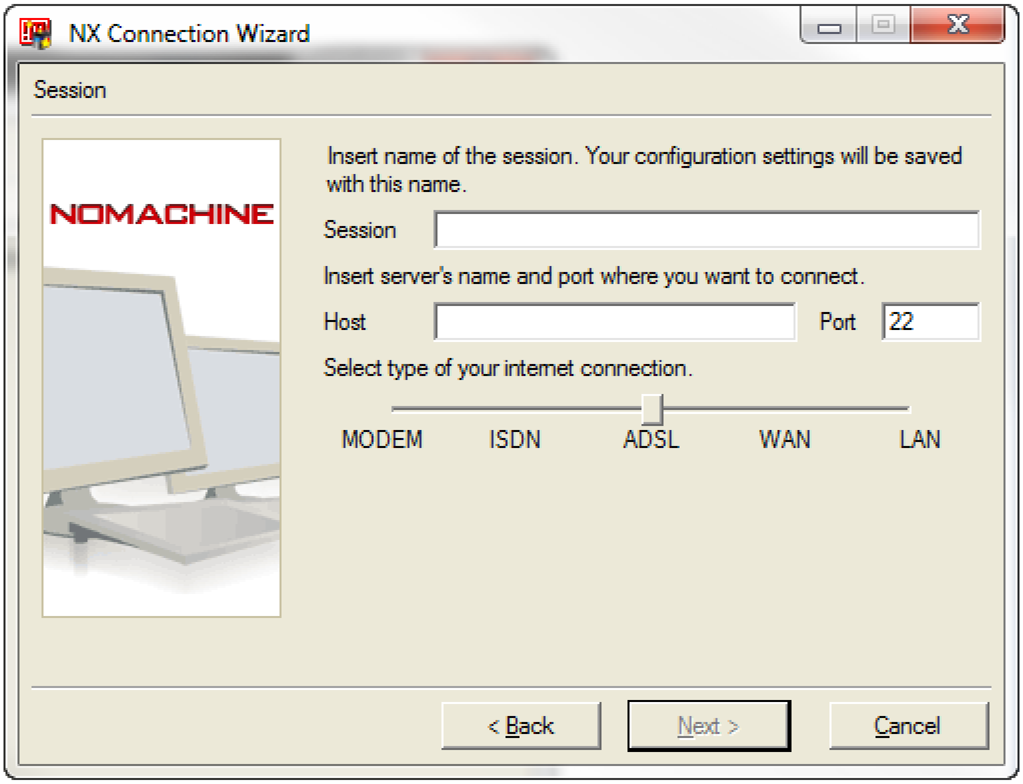
Steps

1. Install **NX Client** (if necessary) from http://www.nomachine.com/download.php.
2. Run **NX Client** (for example, select **NX Client for Windows** from the Windows Start menu).
   - An NX Connection Wizard steps you through creating the initial session.
   - The following Welcome window appears.



   a. Click **Next**.

- The following Session window appears.



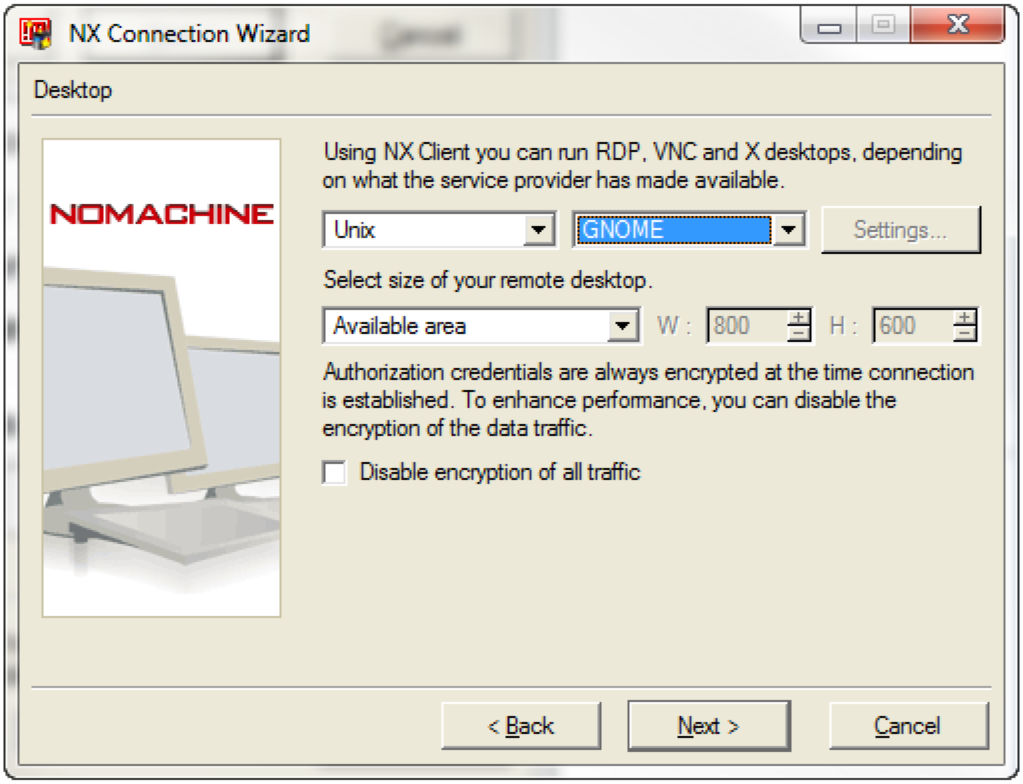
b. Enter anything you like for **Session** (for example, `OracleLinux`).

c. Enter the IP address (provided by your instructor) for **Host**.

d. Accept the remaining defaults and click **Next**.

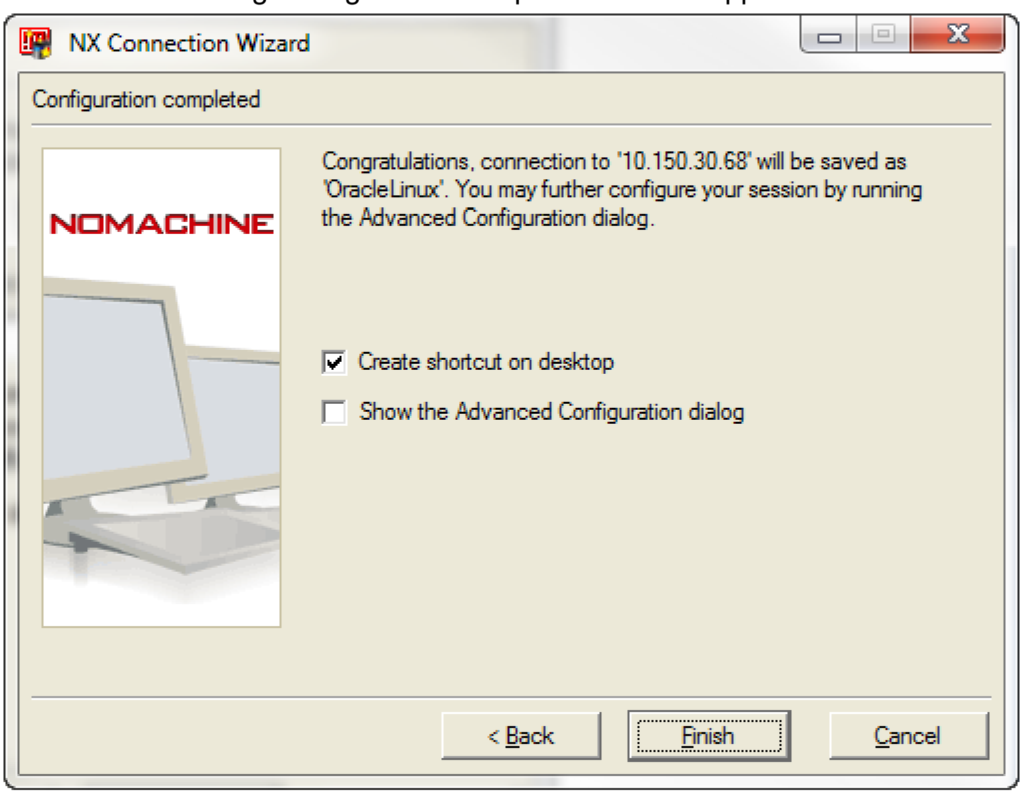
e. The Desktop window appears. Change KDE to **GNOME** by selecting from the drop-down list.

- Your window should look like the following:



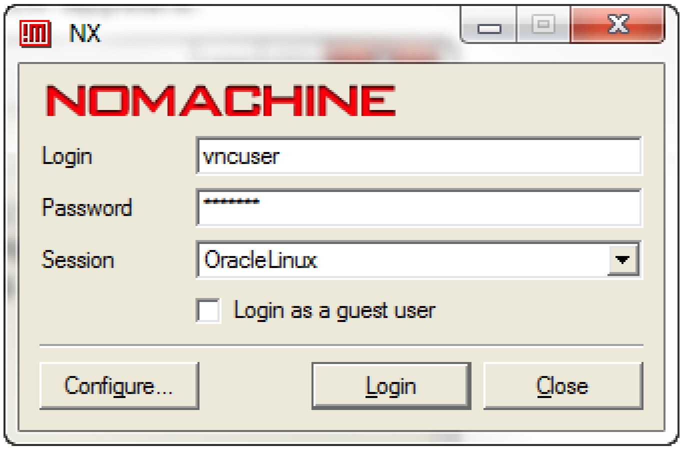
f. Accept all other defaults and click **Next**.

- The following Configuration completed window appears.



g. Click **Finish**.

- The NX Login window appears.



h. For **Login**, enter `vncuser`.

i. For **Password**, enter `vnctech`.

j. Your **Session** defaults to the session that you just created. In this example, the **Session** is OracleLinux. Your session name may be different.

k. Click **Login**.

- The **dom0** GNOME virtual desktop window appears.
- Future connections will bypass the configuration wizard and only bring up the NX Login window.

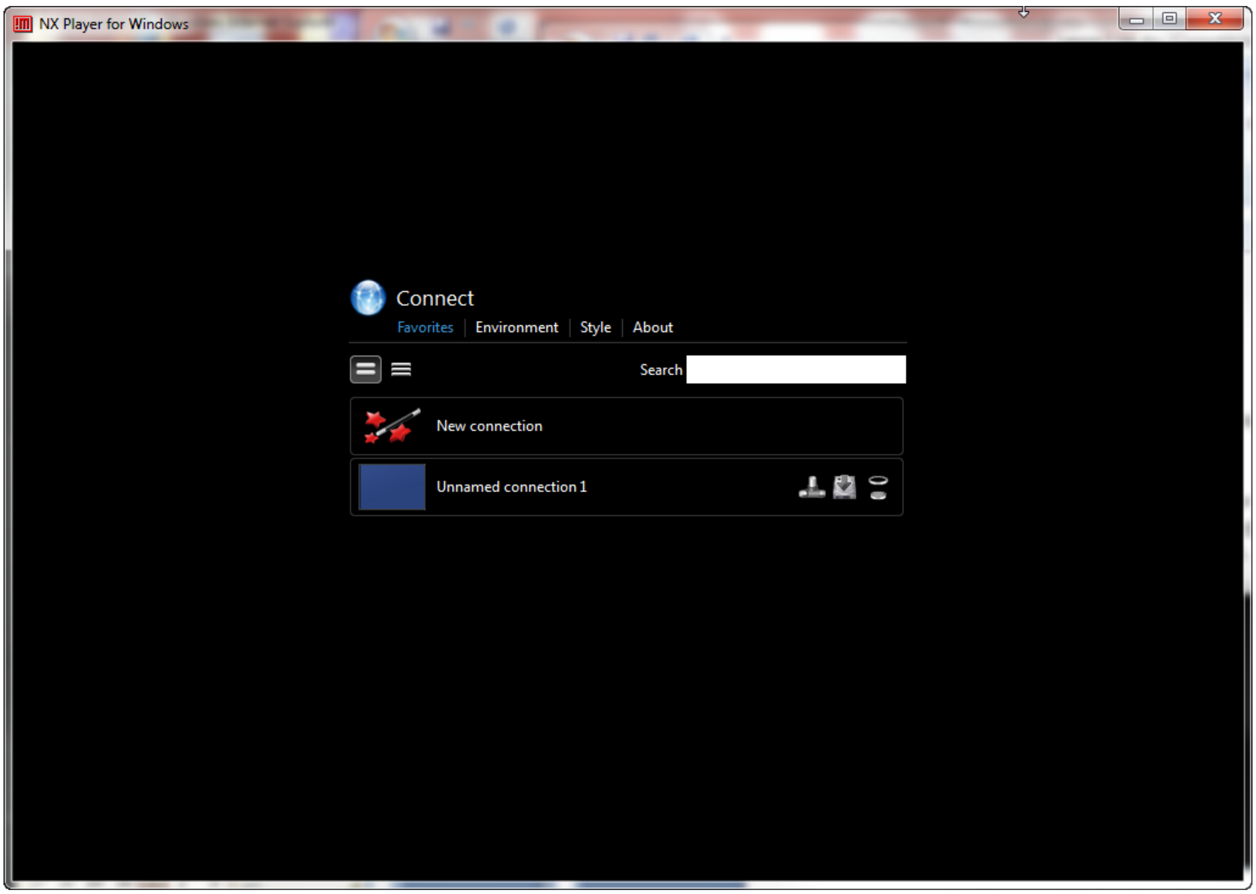
Appendix B: Using NX Player to Connect to dom0

Overview

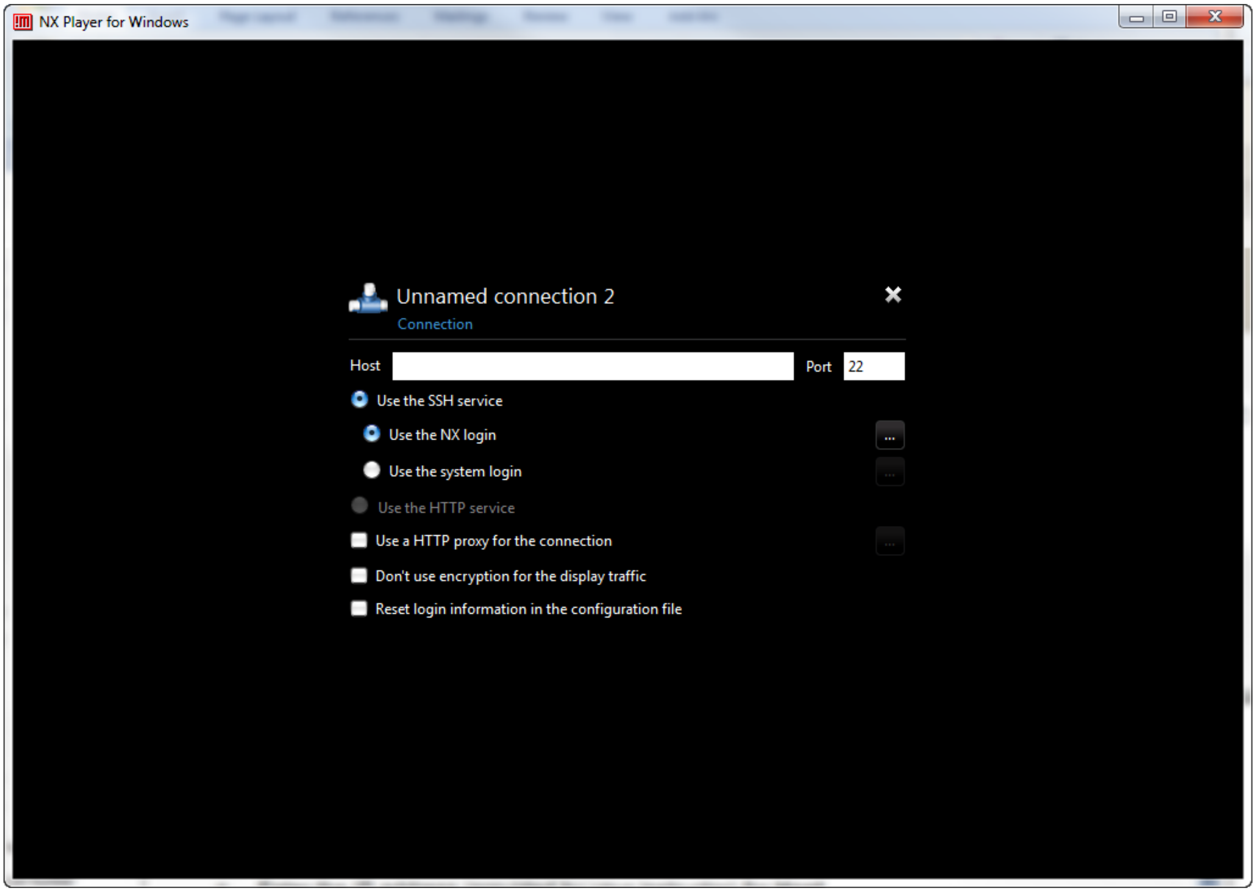
This appendix discusses accessing your student PC (**dom0**) remotely using NX Player. The NX Player in this appendix is NX Player for Windows, Preview 5, version 4.0.132.

Steps

1. Install **NX Player** (if necessary) from http://www.nomachine.com/download.php.
2. Run **NX Player** (for example, select **NX Player for Windows** from the Windows Start menu).
   a. Ensure that the **Favorites** tab is selected.
      - The Connect window appears.



b. Click **New connection** to display the following window.



c. Enter the IP address (provided by your instructor) for **Host**.

d. Accept the defaults:

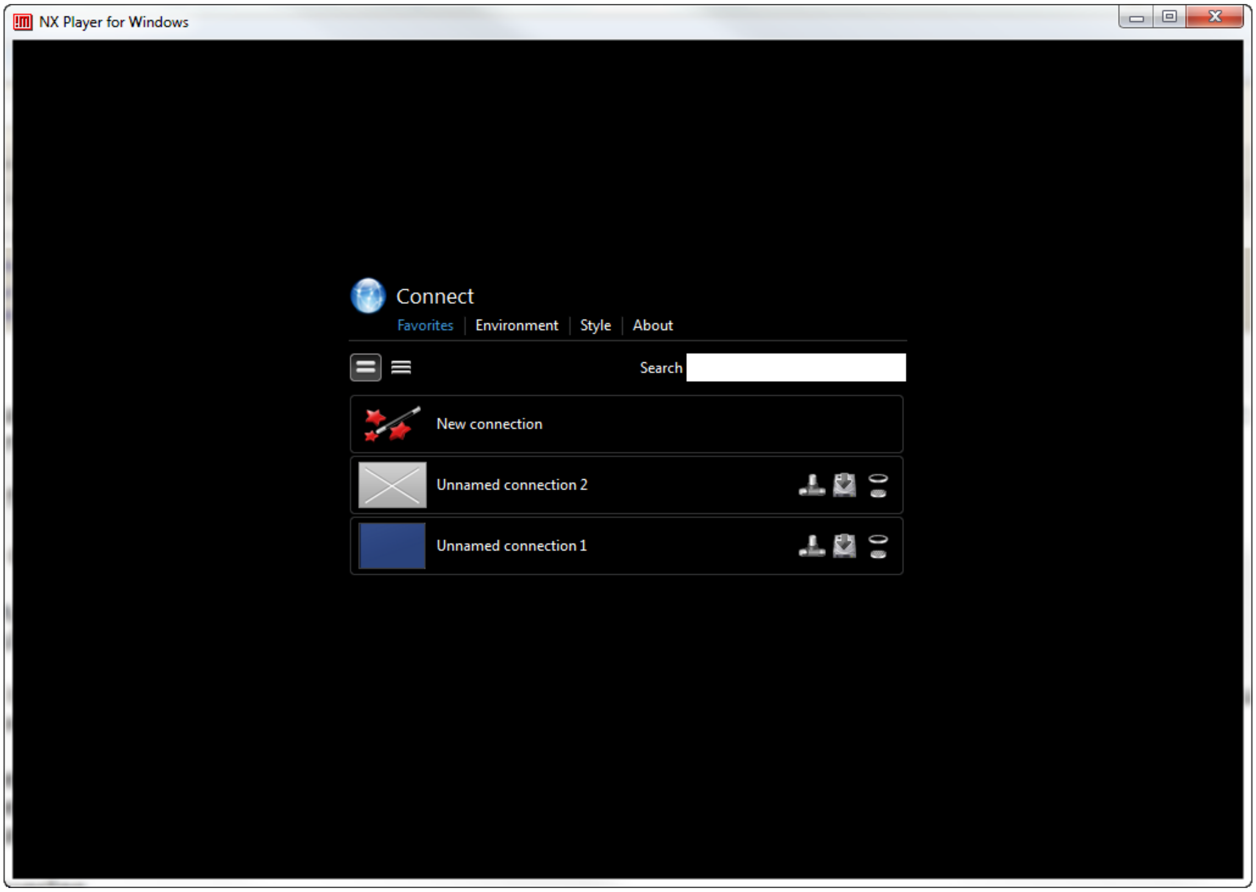
1) Port 22

2) Use the SSH service

3) Use the NX login

e. Note the connection name. In this example it is **Unnamed connection 2**. Yours is most likely **Unnamed connect 1**.

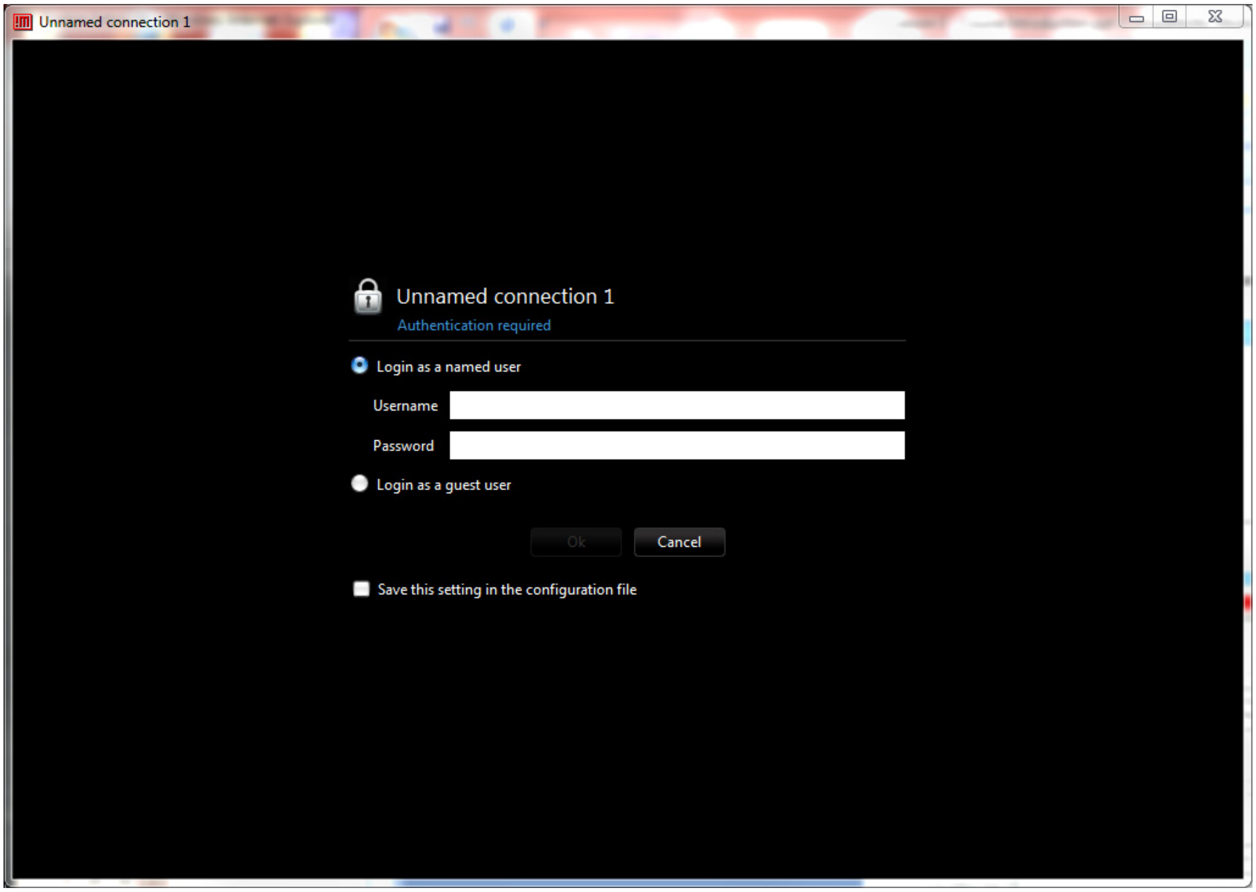
f. Press **Enter**. The following window appears.



g. Click the connection that you just created (**Unnamed connection 1**, for example).

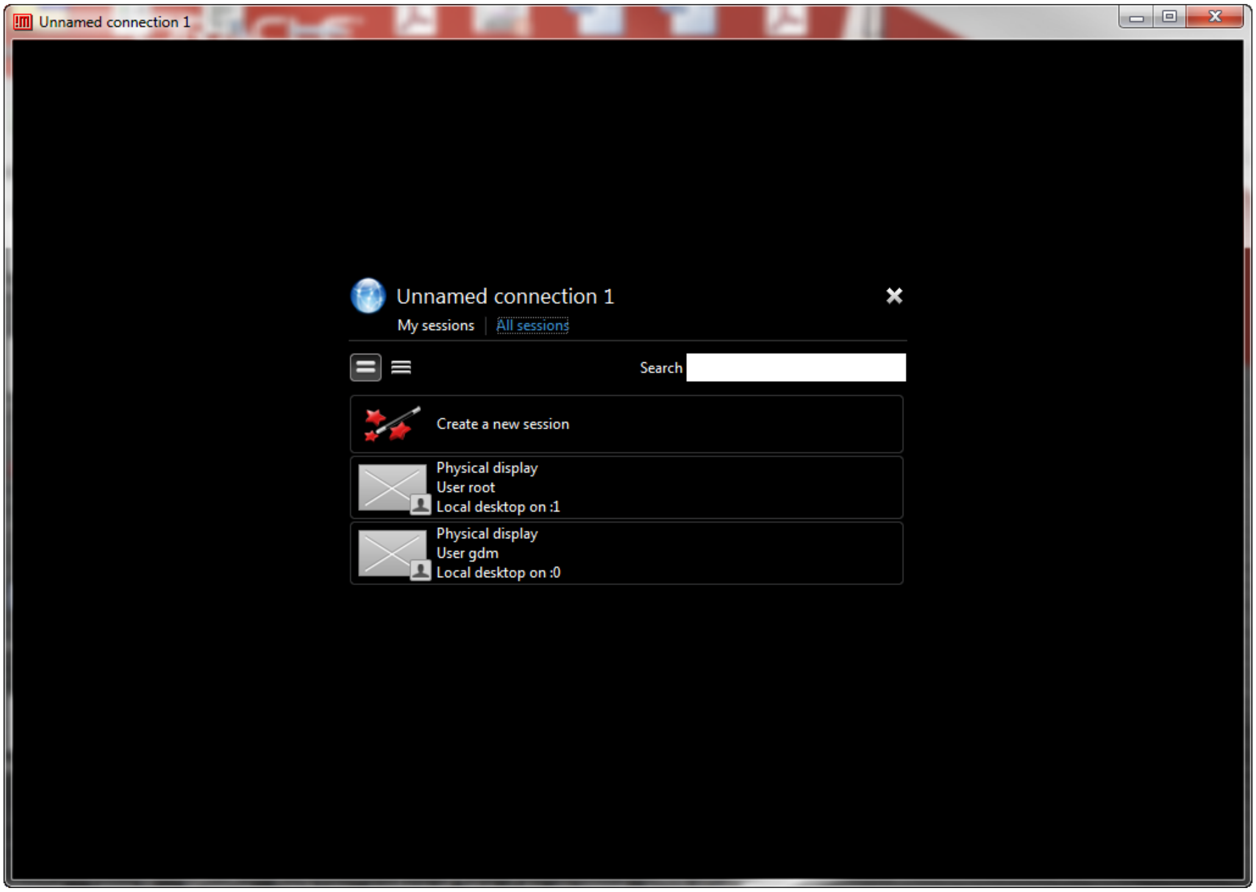
- The Login window appears.

3. Log in.
   - The window shown in the following screenshot appears, prompting for login authentication.



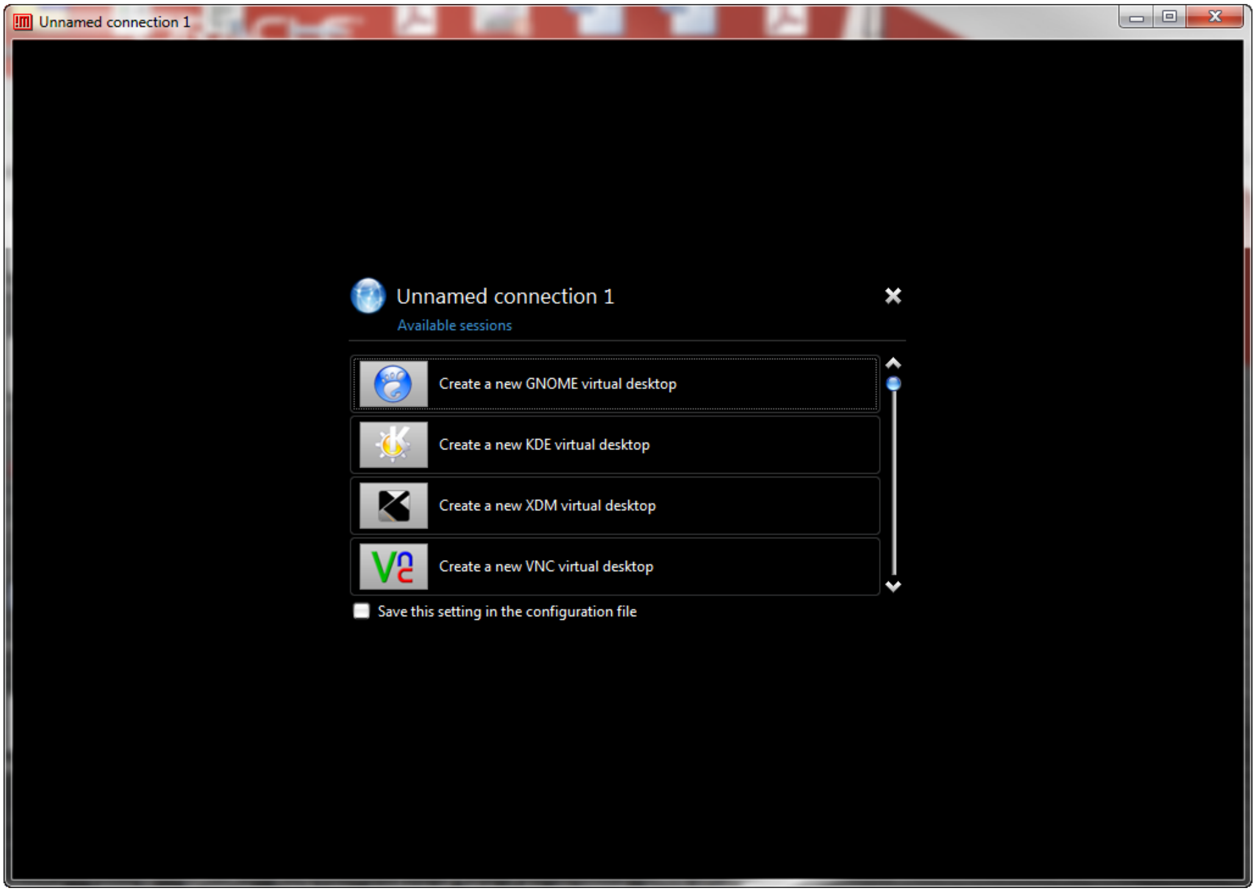
   a. Ensure that **Login as a named user** is selected.
   b. For **Username**, enter `vncuser`.
   c. For **Password**, enter `vnctech`.
   d. Click **Ok**.

4. Create a new session.
   - The window shown in the following screenshot appears.

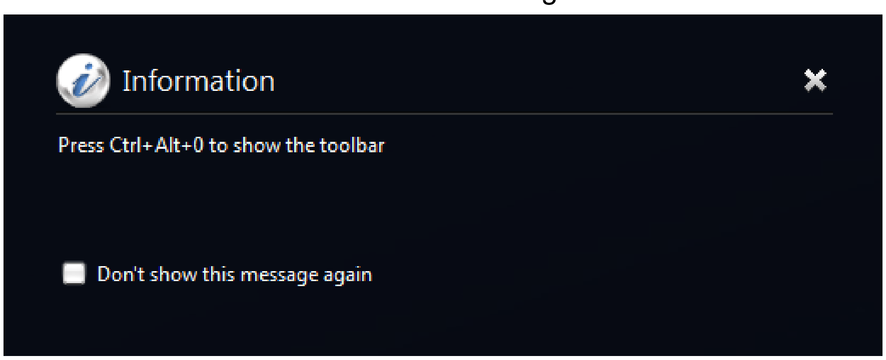


   a. Click **Create a new session**.

5. Create a new GNOME virtual desktop.
   - The window shown in the following screenshot appears.



   a. Click **Create a new GNOME virtual desktop**.
   b. Click the **X** in the **Information** message box to close the box.



   - The **dom0** GNOME virtual desktop window appears.

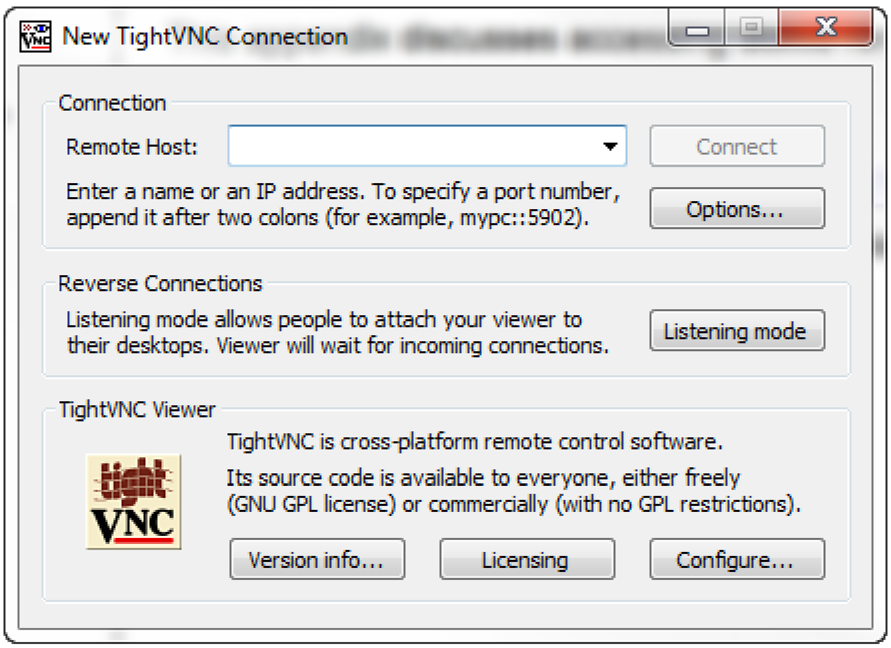
Appendix C: Using VNC (TightVNC) to Connect Directly to VM Guests

Overview

This appendix discusses accessing the VM guest systems that directly uses VNC (TightVNC). It is not recommended to connect to **dom0** or to the **host03** VM by using VNC. Both **dom0** and **host03** have the GNOME user interface, which causes various problems when connecting using VNC.

Steps

1. Install **tightvnc** (if necessary) from http://www.tightvnc.com/.
2. Run **TightVNC Viewer** (for example, select **TightVNC Viewer** from the Windows Start menu).
   - The following New TightVNC Connection window appears.



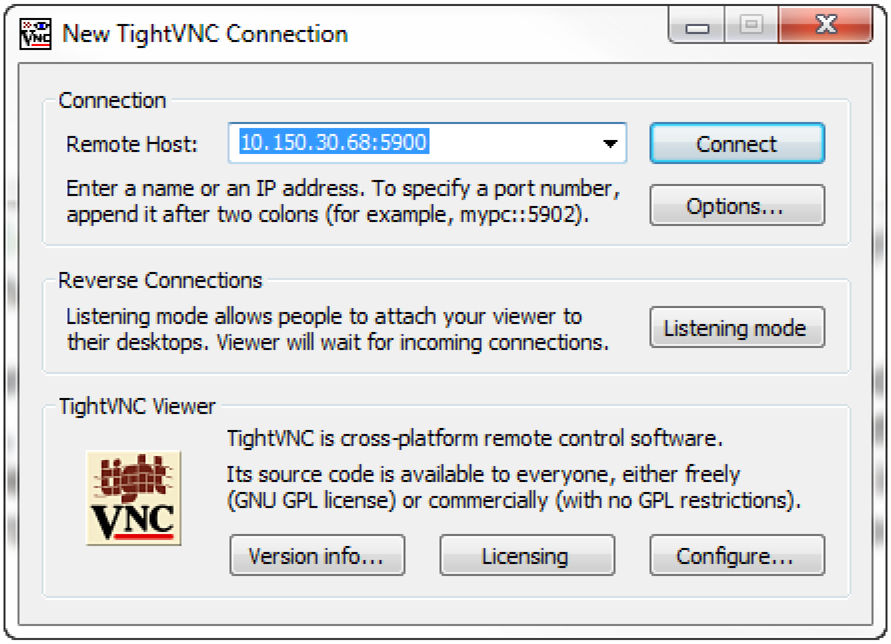
3. Connect directly to your **host01** virtual machine.
   - The following assumptions are made:
     - The **host01** VM was created first (has a port number of `5900`).
     - The **host02** VM was created next (has a port number of `5902`).
     - The **host03** VM was created last (has a port number of `5903`).
   - The output of the following commands (from **dom0** as `root`) indicates that this assumption is true.

```
# xm list -l host01 | grep location
          (location 0.0.0.0:5900)
# xm list -l host02 | grep location
          (location 0.0.0.0:5902)
# xm list -l host03 | grep location
          (location 0.0.0.0:5903)
```

- Enter the IP address (provided by your instructor), followed by the port number to connect directly a VM guest.

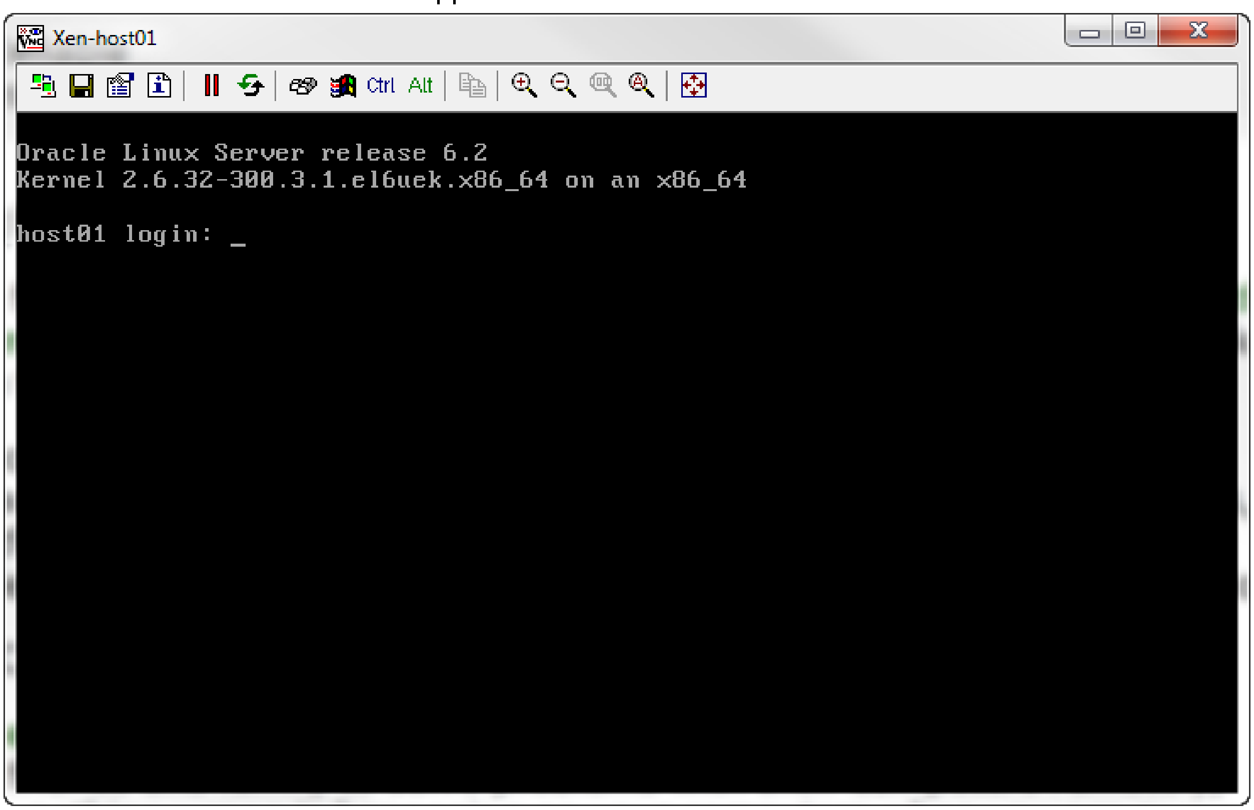
a. To connect directly to the **host01** VM, enter the following.

  - In this example, the IP address of your student PC is 10.150.30.68. Your IP address is different.



1) Click **Connect**.

   - A terminal window appears.



2) Log in as `root` with password `0racle` (leading zero, not letter O).
3) Enter the `hostname` command to confirm that you are logged in to **host01**.

```
# hostname
host01.example.com
```

4) Log off by entering the `exit` command.
5) Close the VNC window by clicking the **X** in the top-right corner of the window.

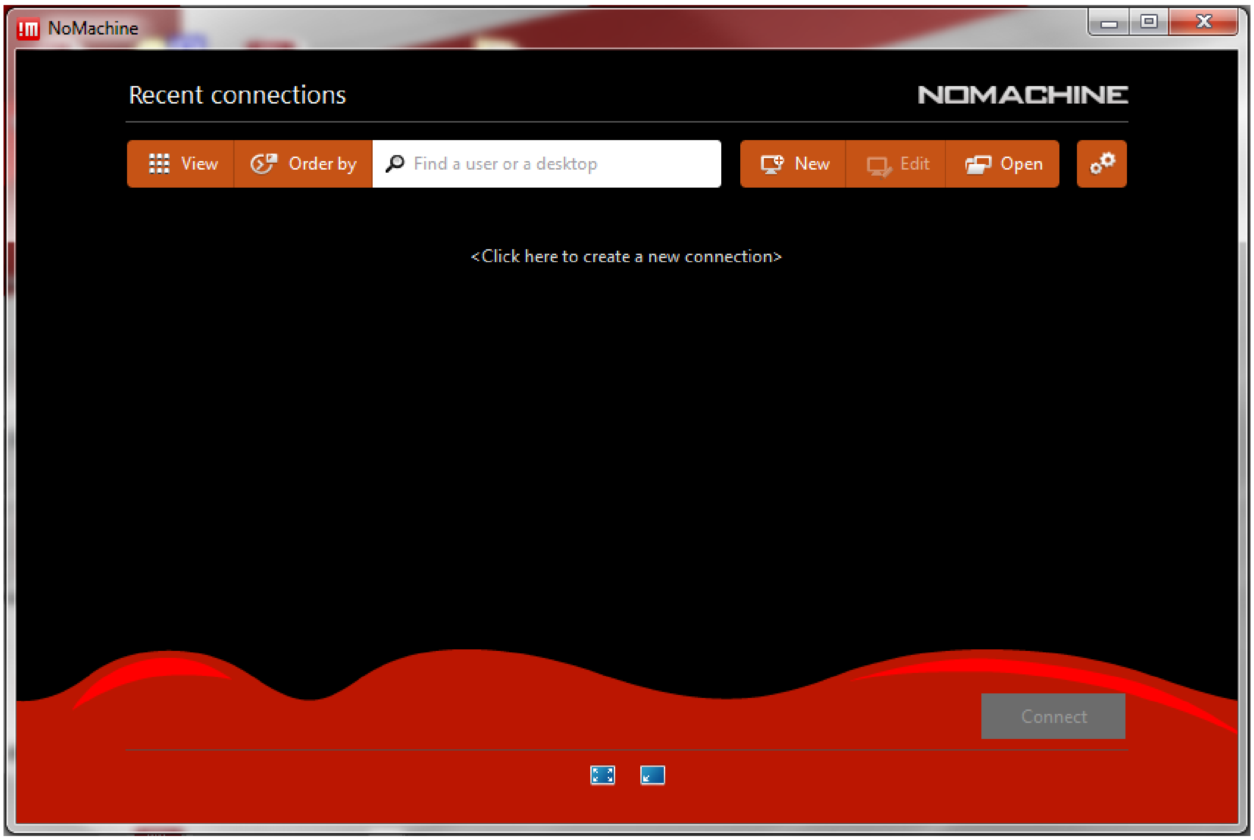
Appendix D: Using NoMachine Version 4 to Connect to dom0

Overview

This appendix discusses accessing your student PC (**dom0**) remotely by using NoMachine for Windows, Version 4.1.29.

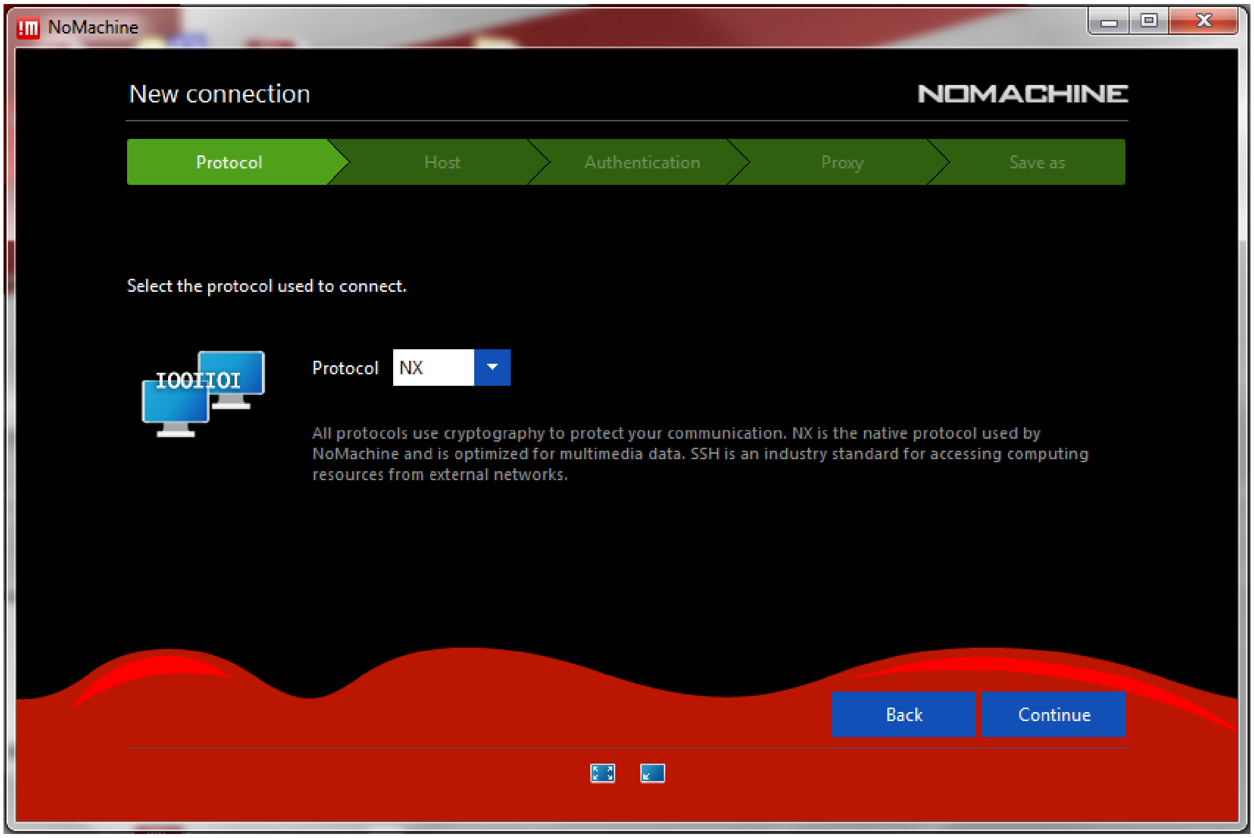
Steps

1. Install **NoMachine for Windows** (if necessary) from https://www.nomachine.com/.
2. Run **NoMachine** (for example, select **NoMachine** from the Windows Start menu).
   - The following Recent connections window appears.



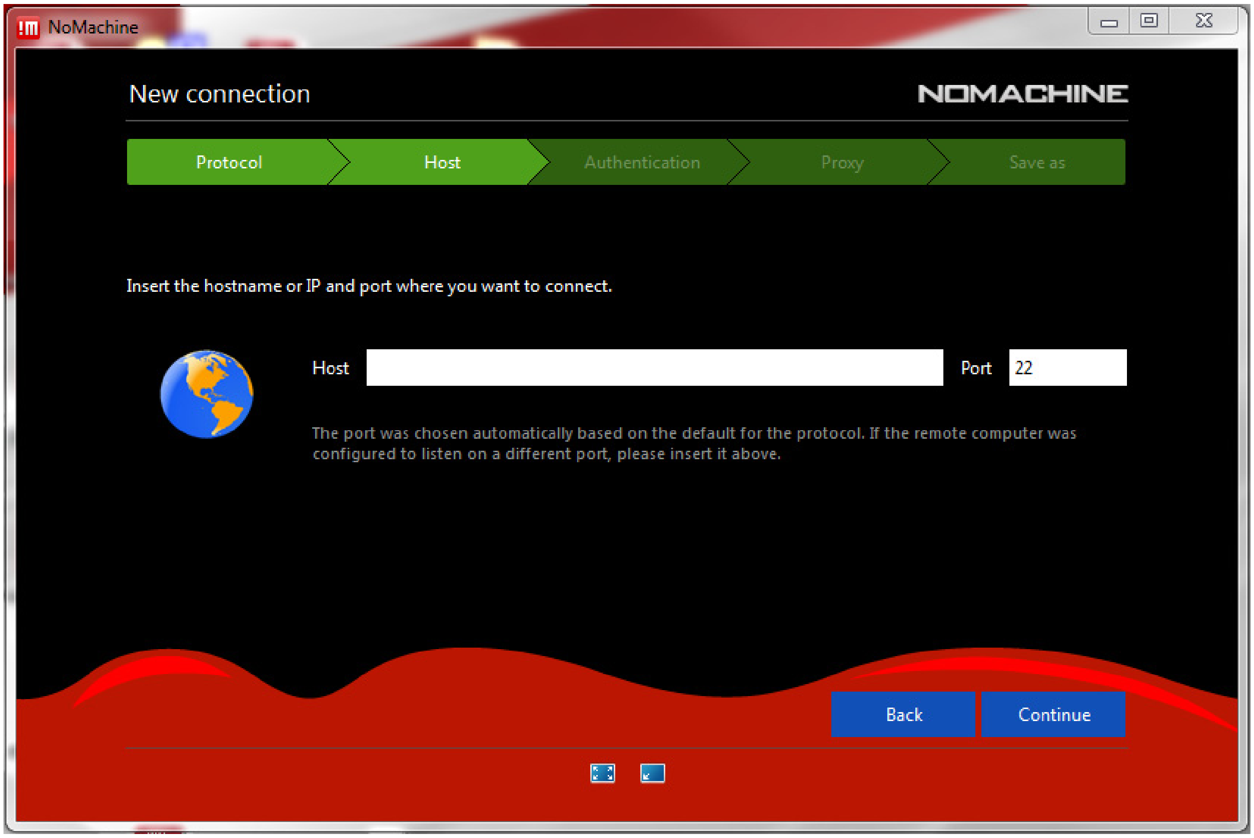
   a. Click **New** from the tool bar or click the **<Click here to create a new connection>** link on this screen.

- The following Session window appears.



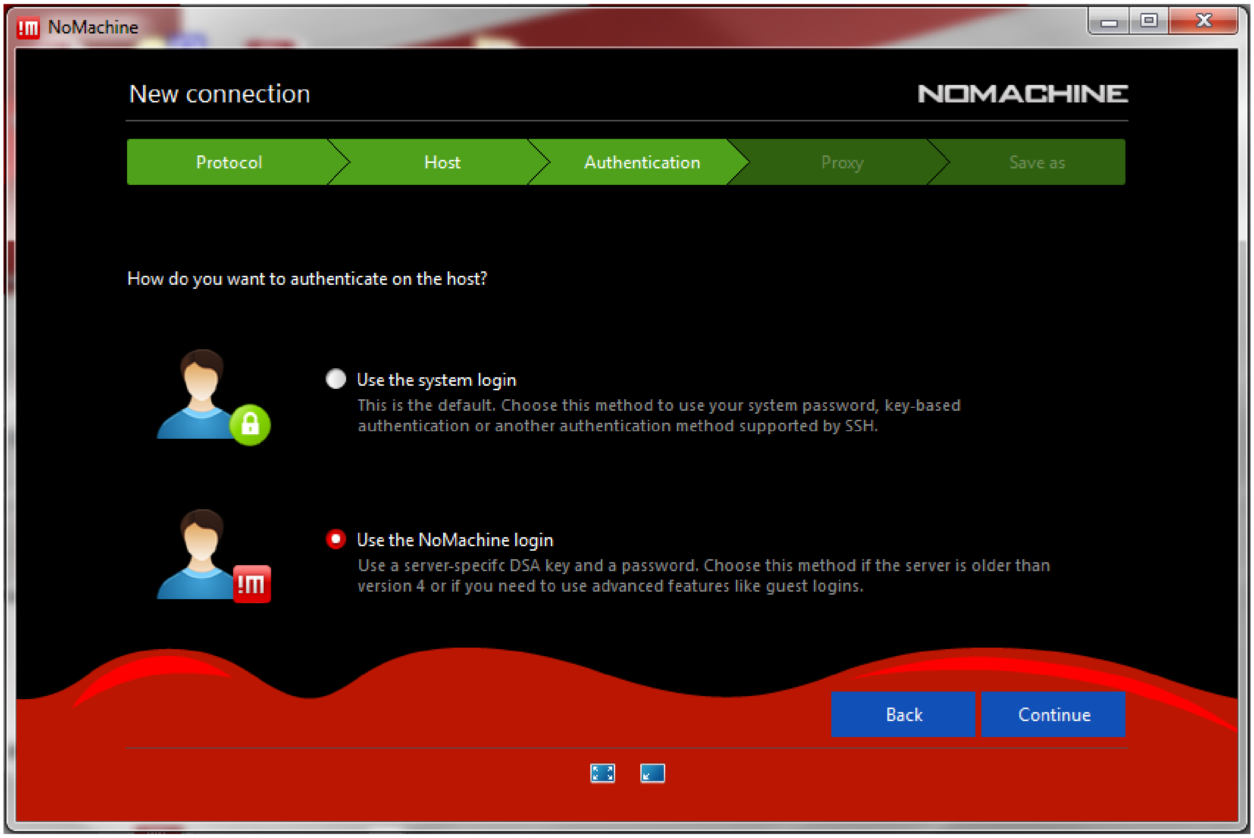
b. From the **Protocol** drop-down menu, select **SSH**.

- Click **Continue**.
- The following window appears:



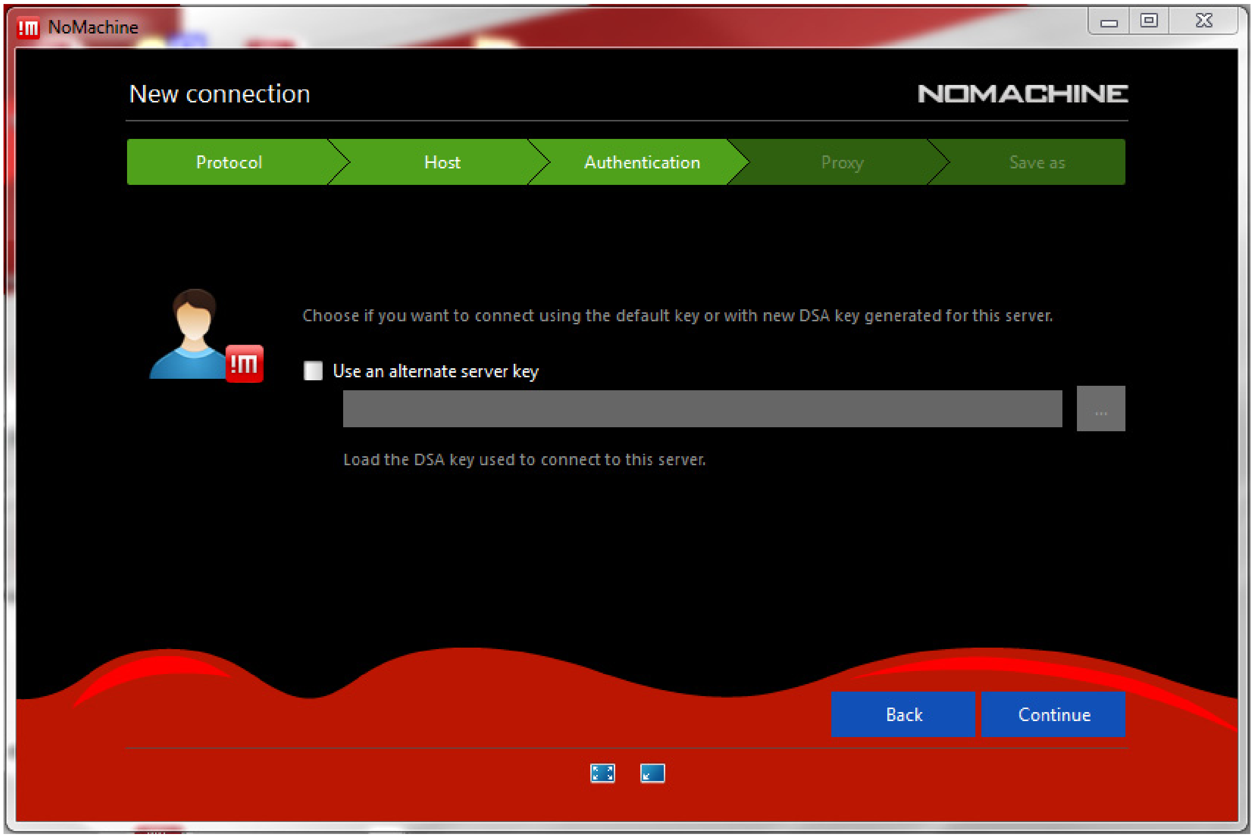
c. For **Host**, enter the **IP address** of the Oracle University machine that has been assigned to you.

- Click **Continue**.
- The following window appears:



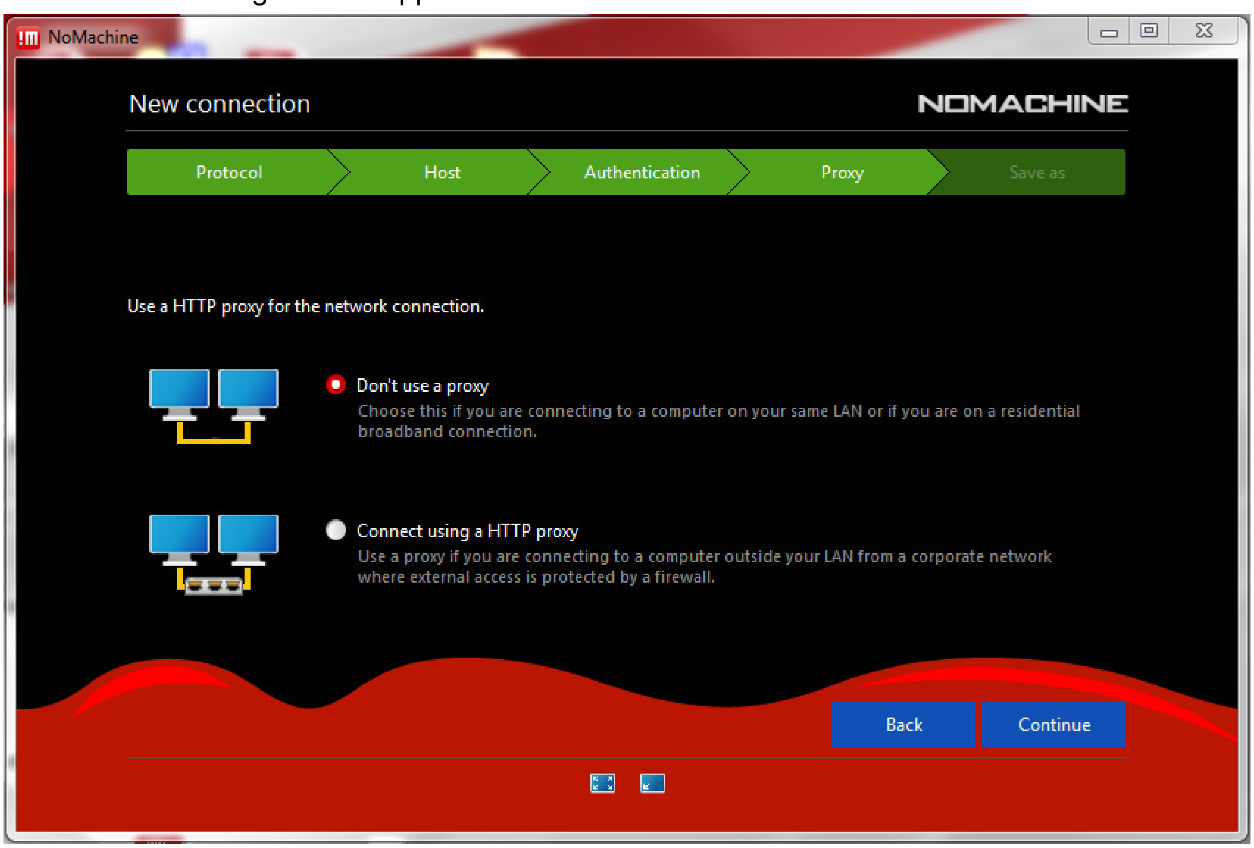
d. In the **Authentication** window, select **Use the NoMachine login** option.

- Click **Continue**.
- The following window appears:



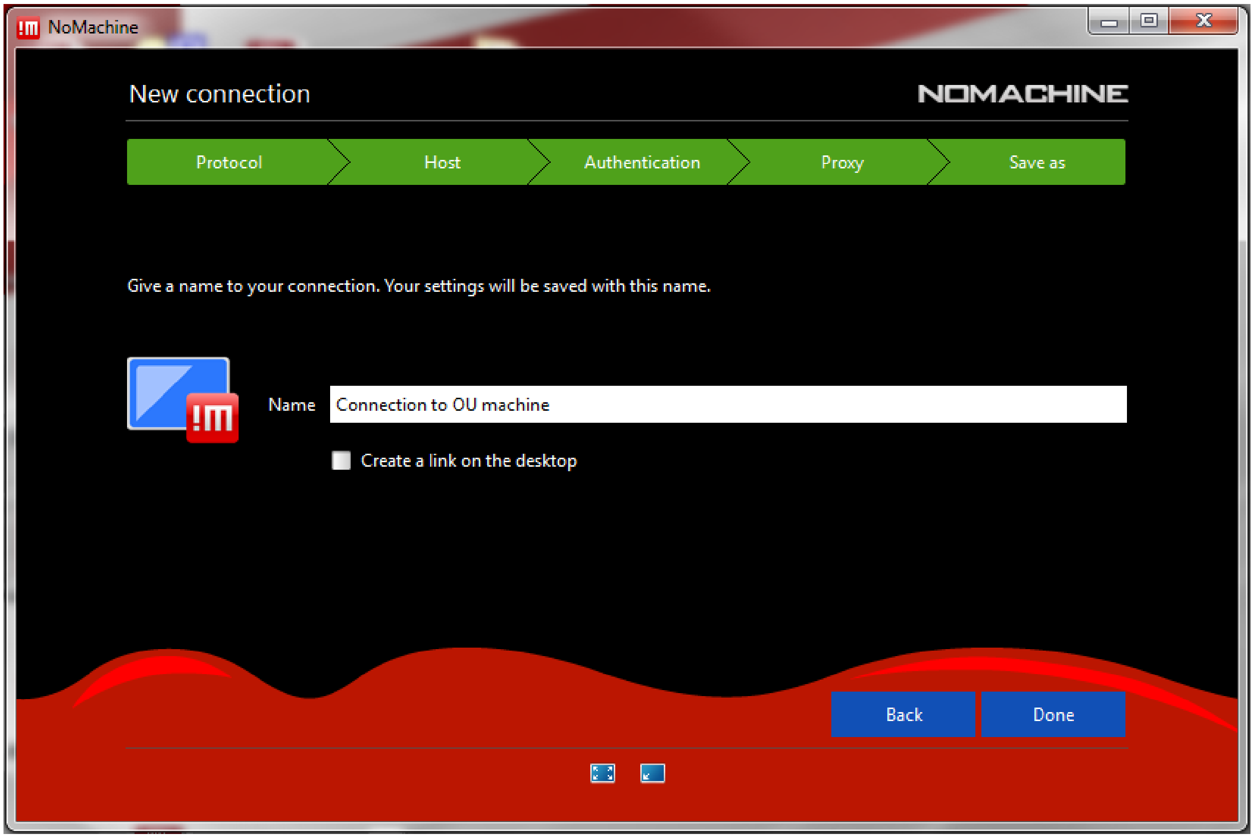
e. Click **Continue**. Do not enter anything on this screen.

- The following window appears:



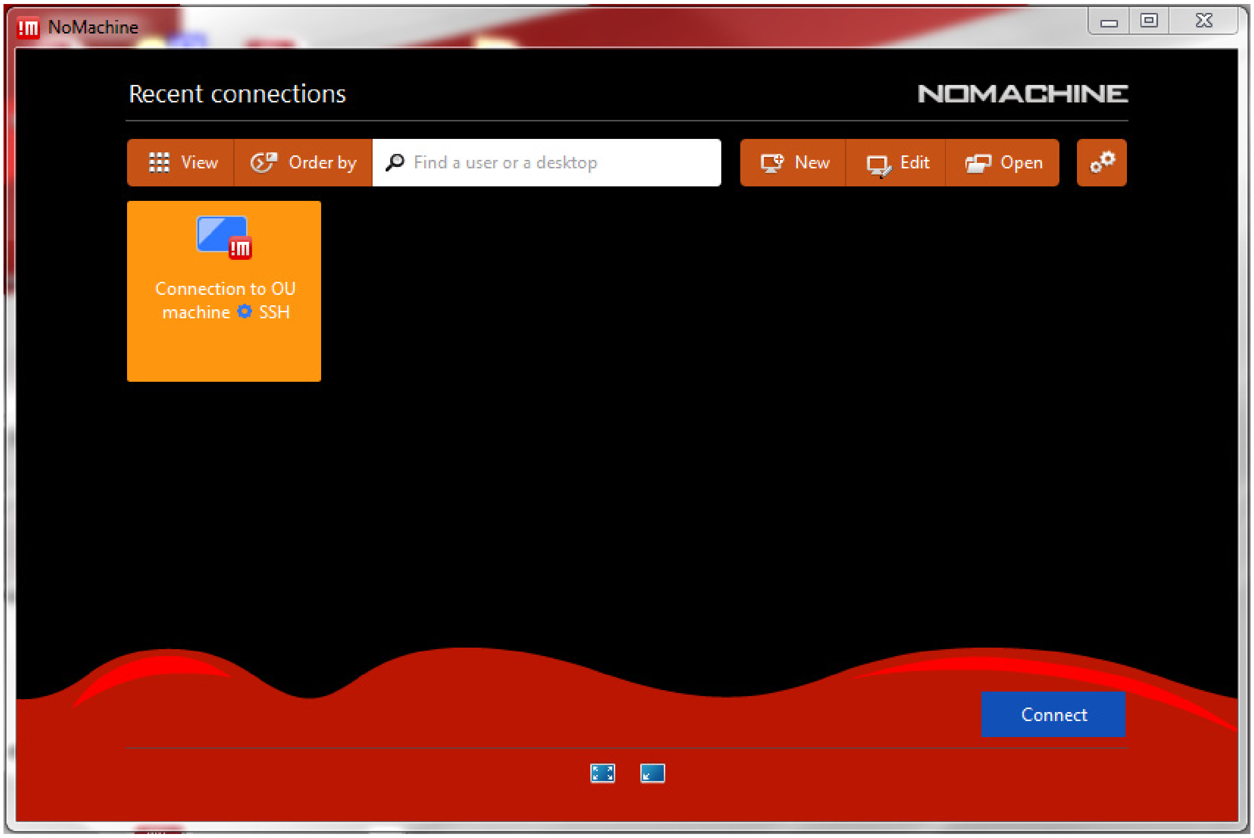
f. In the **Proxy** window, select **Don't use a proxy** option.

- Click **Continue**.
- The following window appears:



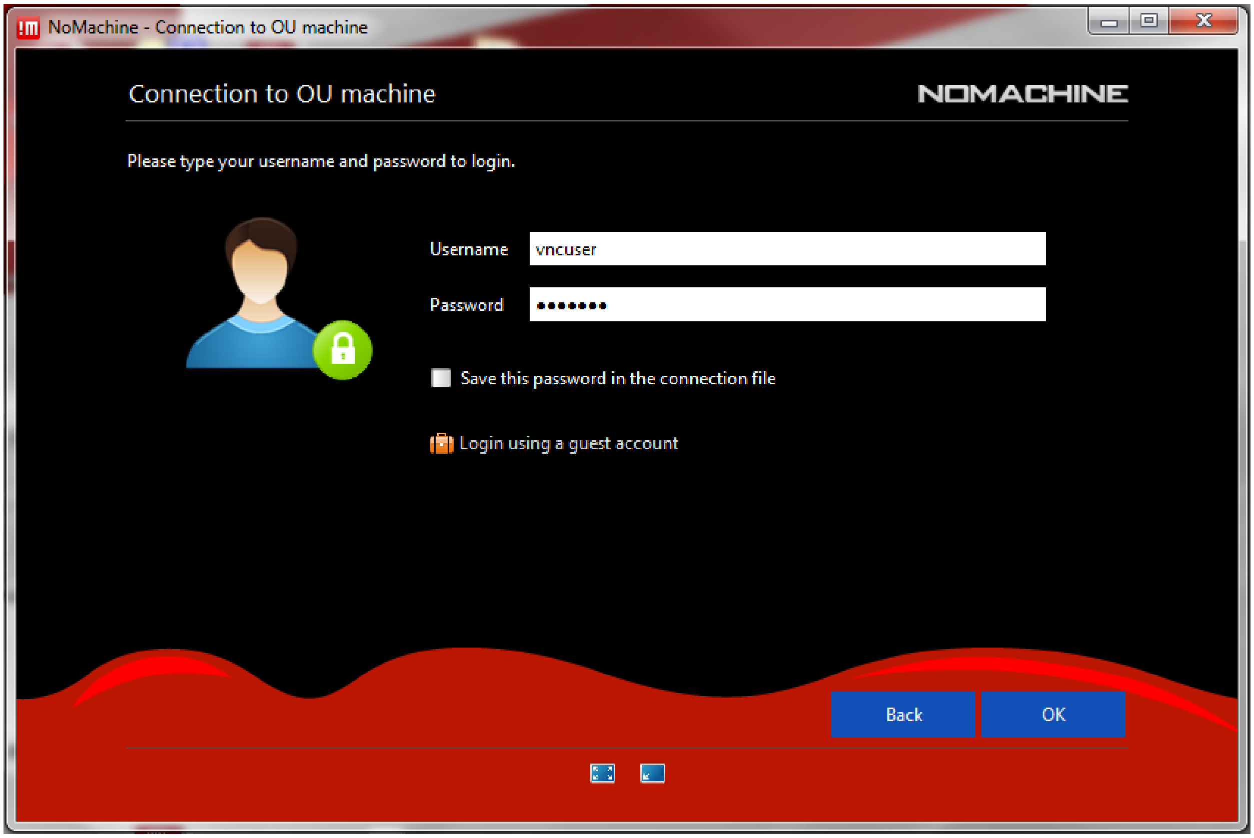
g. In the **Save as** window, enter a **Name** for you new connection.

- The example shows Connection to OU machine but you can enter any name you want.
- Click **Done**.
- The following window appears:



h. Your new connection appears on this screen.

- Select the icon for the connection to your machine and click **Connect**.
- The following window appears:

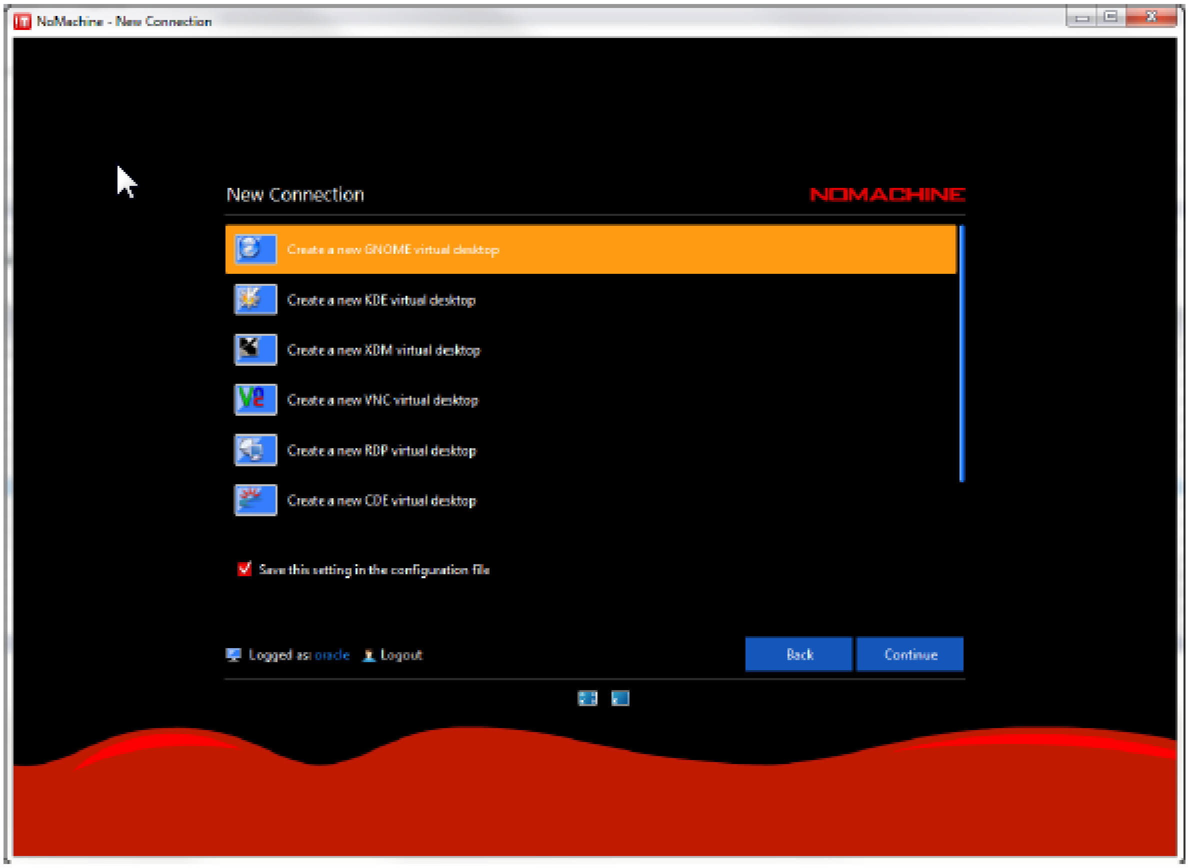


i. For **Username**, enter `vncuser`.

j. For **Password**, enter `vnctech`.

k. Click **OK**.

- The following window appears:



l. Select **Create a new GNOME virtual desktop**.

m. Click **Continue**.

- The **dom0** GNOME virtual desktop window appears.