ORACLE UNIVERSITY

Oracle Linux 7: Advanced Administration

Activity Guide – Volume I
D90758GC10
Edition 1.0 | September 2015 | D92967

Learn more from Oracle University at **oracle.com/education/**

ORACLE

Author

Craig McBride

Technical Contributors and Reviewers

Avi Miller, Elena Zannoni, Wim Coekaerts, Harald Van Breederode, Joel Goodman, Manish Kapur, Yasar Akthar, Antoinette O'Sullivan, Gavin Bowe, Steve Miller, Herbert Van Den Bergh, Todd Vierling and John Haxby

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Course Introduction

Chapter 1

Course Practice Environment: Security Credentials

For OS usernames and passwords, see the following:

- If you are attending a classroom-based or live virtual class, ask your instructor or LVC producer for OS credential information.
- If you are using a self-study format, refer to the communication that you received from Oracle University for this course.

For product-specific credentials used in this course, see the following table:

Product-Specific Credentials		
Virtual Machines/Application	**Username**	**Password**
host01/OS	root	oracle
host01/OS	oracle	oracle
host02/OS	root	oracle
host02/OS	oracle	oracle
host03/OS	root	oracle
host03/OS	oracle	oracle

Practices for Lesson 1: Overview

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

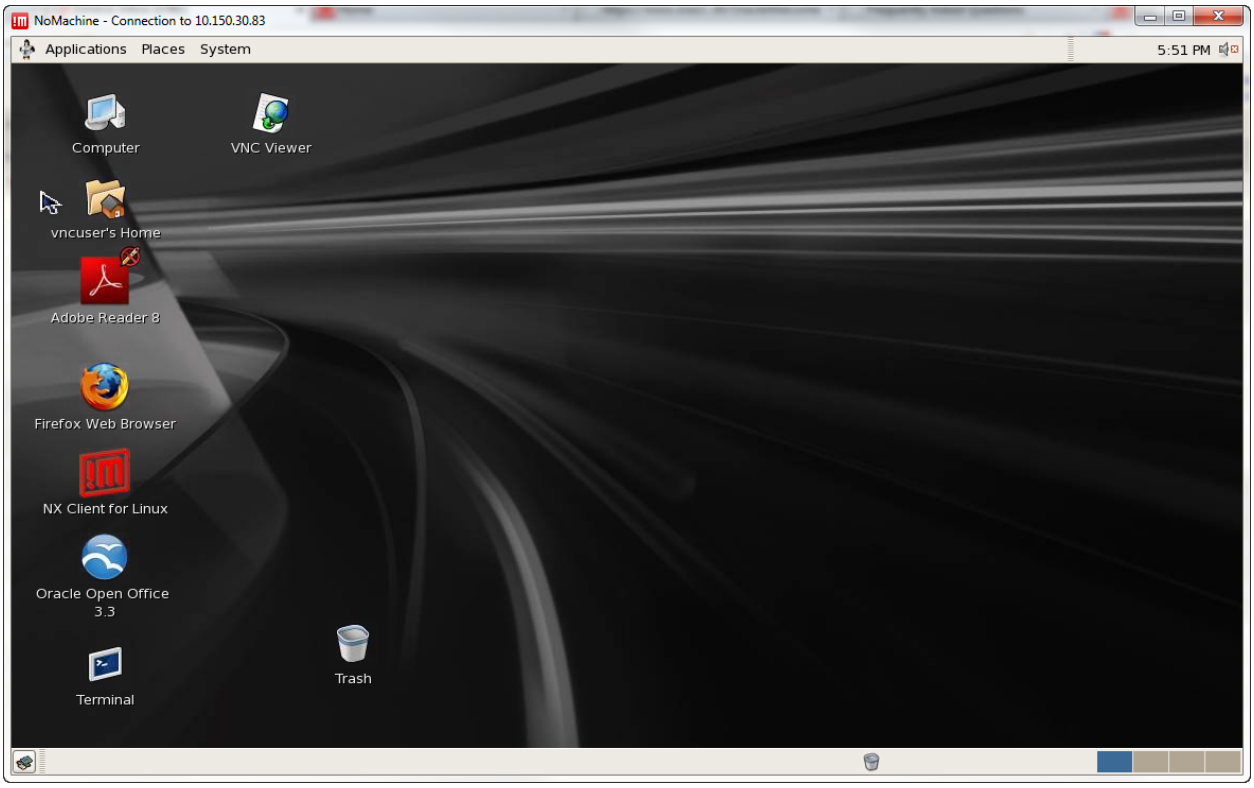
In this practice, you explore the **dom0** configuration and directory structure.

Assumptions

- Your instructor has assigned a student PC to you.
- Your student PC is running Oracle VM Server for x86 version 3.2.1.
- 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 7 installed.

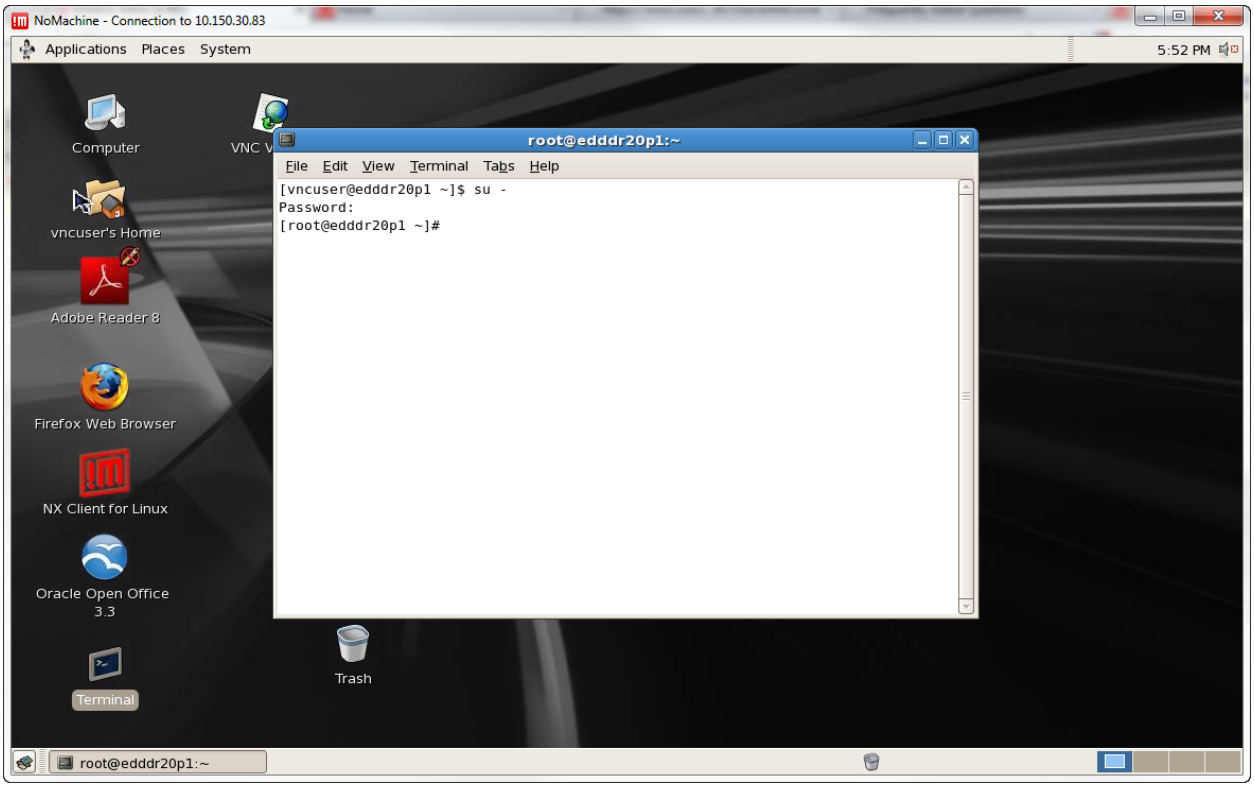
Tasks

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 edddr20p1 2.6.39-300.22.2.el5uek #1 SMP Fri Jan 4 12:40:29
PST 2013 x86_64 x86_64 x86_64 GNU/Linux
```

   - In this example, the operating system is Linux.
   - The Linux kernel is `2.6.39-300.22.2.el5uek`.
   - The host name is `edddr20p1`. (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
...
bond0     Link encap:Ethernet ...
          inet addr:10.150.30.83 ...
...
eth0      Link encap:Ethernet ...
...
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 ...
...
virbr2    Link encap:Ethernet ...
          inet addr:192.168.2.1 ...
...
virbr3    Link encap:Ethernet ...
          inet addr:192.168.3.1 ...
...
```

   - In this example, the network interface for **dom0** is `bond0` and is assigned an IP address of `10.150.30.83`. 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`.
   - The `virbr2` interface is a third `xen` bridge interface used by VM guests. It is assigned an IP address of `192.168.2.1`.
   - The `virbr3` interface is a fourth `xen` bridge interface used by VM guests. It is assigned an IP address of `192.168.3.1`.
   - 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
drwxrwxrwx ... iso_pool
drwxrwxrwx ... publish_pool
drwxrwxrwx ... running_pool
drwxrwxrwx ... seed_pool
drwxrwxrwx ... sharedDisk
```

   - There are five directories in the `/OVS` directory.

   b. Explore the `/OVS/runing_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 ... vpn-host1
drwxr-xr-x ... vpn-host2
```

   - The files needed to create the VMs are in separate directories in the `/OVS/running_pool` directory.
   - This example shows that seven VM directories exist, for VMs **host01**, **host02**, **host03**, **host04**, **host05**, **vpn-host1**, and **vpn-host2**.
   - The **host04** VM is preconfigured with access to Oracle's Public Yum Server. This VM is used in "Practices for Lesson 8: 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 "**vpn**" VMs are used in "Practice 10-7: Configuring a Site-to-Site Virtual Private Network (VPN)."

   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,xvda,w',
         'file:/OVS/running_pool/host01/u01.img,xvdb,w',
         'file:/OVS/running_pool/host01/u02.img,xvdd,w',
         'file:/OVS/seed_pool/OracleLinux-R7-U1-Server-x86_64-
dvd.iso,xvdc:cdrom,r']
vif = [ 'mac=00:16:3e:00:01:01, bridge=virbr0',
        'mac=00:16:3e:00:02:01, bridge=virbr1',
        'mac=00:16:3e:00:03:01, bridge=virbr2',
        'mac=00:16:3e:00:04:01, bridge=virbr3']
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 four virtual network interfaces. The interface on the `virbr0` bridge is `eth0`, the interface on the `virbr1` bridge is `eth1`, the interface on the `virbr2` bridge is `eth3`, and the interface on the `virbr3` bridge is `eth4`.

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

```
# cd /OVS/sharedDisk
# ls -l
-rw-r--r-- ... physDisk1.img
```

- The `physDisk1.img` file is used as a shared disk (shared by all VM guests) in "Practices for Lesson 11: OCFS2 and Oracle Clusterware."

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

```
# cd /OVS/seed_pool
# ls -l
drwxr-xr-x  ...  debug
drwxr-xr-x  ...  dtrace_rpms
drwxr-xr-x  ...  host07
-rw-r--r--  ...  OracleLinux-R7-U1-Server-x86_64-dvd.iso
-rw-r--r--  ...  physDisk1.tgz
drwxr-xr-x  ...  sfws
-rw-r--r--  ...  system01.tgz
-rw-r--r--  ...  system02.tgz
-rw-r--r--  ...  system03.tgz
-rw-r--r--  ...  system04.tgz
-rw-r--r--  ...  system05.tgz
-rw-r--r--  ...  u01_01.tgz
-rw-r--r--  ...  u01_03.tgz
-rw-r--r--  ...  u02_01.tgz
-rw-r--r--  ...  u02_02.tgz
-rw-r--r--  ...  u02_03.tgz
-rw-r--r--  ...  u03_02.tgz
-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  ...  vmvpn1.cfg
-rwxr-xr-x  ...  vmvpn2.cfg
-rw-r--r--  ...  vpn-host1.tgz
-rw-r--r--  ...  vpn-host2.tgz
```

- This directory contains many files that are used to create the initial environment.
- Oracle Linux 7.1 is installed on the **host01**, **host02**, and **host03** VMs from the `OracleLinux-R7-U1-Server-x86_64-dvd.iso` file in this directory.
- Other files in this directory are used in various practices.

g. Explore the `/var/www/html/repo/OracleLinux/OL7/1/x86_64` directory:

- Your system is configured as a local Yum repository.
- This directory contains the contents of the Oracle Linux 7.1 ISO.
- Note that the RPM software packages are in the `Packages` directory.
- A `.repo` file exists on each VM pointing to this Yum repository.

```
# ls -l /var/www/html/repo/OracleLinux/OL7/1/x86_64
addons  images    RELEASE-NOTES-U1-en  RPM-GPG-KEY-oracle
EFI     isolinux  RELEASE-NOTES-U1-en.html  TRANS.TBL
EULA    LiveOS    repodata
GPL     Packages  RPM-GPG-KEY
```

Practice 1-2: Starting, Stopping, and Listing VM 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 **host03** 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.
   - Note that **host03** is no longer active.

```
# xm shutdown -w host03
Domain host03 terminated
All domains terminated
# xm list
Name                  ID     Mem  VCPUs       State      Time(s)
Domain-0               0    2048      2      r-----       289.6
host01                 1    1536      1      -b----       157.6
host02                 2    1536      1      -b----       159.0
```

3. Start a VM.

   Use the `xm create <config_file>` command to start the **host03** 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.

   - Note that **host03** is now active.
   - The State column for **dom0** and **host03** shows 'r' (run state). The State column for **host01** and **host02** 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.

```
# cd /OVS/running_pool/host03
# xm create vm.cfg
Using config file "./vm.cfg".
Started domain host03 (id=#)
# 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      r-----        13.2
```

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

```
# 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
```

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

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes, 25165824 sectors
/dev/xvda1   *    2048    1026047     512000   83   Linux
/dev/xvda2     1026048   25165823   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

```
Disk /dev/mapper/ol-root: 11.0 GB, 11022630912 bytes, ...
Disk /dev/mapper/ol-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 7.1 installed – Minimal Install Base Environment.
- This system disk uses LVM volumes for the `root` and `swap` partitions.

c. Use the `ip addr` command to display the network interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP>  mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.101/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:01 brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:03:01 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:04:01 brd ff:ff:ff:ff:ff:ff
```

- The system has four Ethernet network interfaces, `eth0`, `eth1`, `eth2`, and `eth3`.
- The `eth0` interface is on the `192.0.2` subnet, and provides access to **dom0** and the other VM guest systems. The remaining interfaces do not have IP addresses.
- The `eth1` interface is configured in “Practices for Lesson 2: Network Addressing and Name Services.”
- The `eth2` and `eth3` interfaces are configured as part of a bonded network interface in “Practices for Lesson 10: Advanced Networking.”

- The `eth1` interface is configured on a private subnet, `192.168.1`, and is used in "Practices for Lesson 11: 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.

- The kernel is UEK Release 3.

```
# uname -r
3.8.13-55.1.6.el7uek.x86_64
```

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

- Two `.repo` files exist, the Public Yum repository file for Oracle Linux 7 and a custom repository file, `vm.repo`, for the local Yum repository on **dom0**.

```
# cd /etc/yum.repos.d
# ls
public-yum-ol7.repo  vm.repo
```

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

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

```
# grep enabled *
public-yum-ol7.repo:enabled=0
public-yum-ol7.repo:enabled=0
...
vm.repo:enabled=1
```

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

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

```
# cat vm.repo
[OL7.1Dom0]
Name="Oracle Linux 7.1 Dom0 Repo"
baseurl=http://192.0.2.1/repo/OracleLinux/OL7/1/x86_64
enabled=1
gpgkey=http://192.0.2.1/repo/OracleLinux/OL7/1/x86_64/RPM-GPG-
KEY-oracle
gpgcheck=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, 41943040 sectors
/dev/xvda1   *     2048    1026047     512000   83   Linux
/dev/xvda2      1026048   41943040   20458496   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvde: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/mapper/ol-root: 18.8 GB, 18798870528 bytes, ...
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, ...
```

   - Four devices are available: `/dev/xvda`, `/dev/xvdb`, `/dev/xvdd` and `/dev/xvde`.

- Do not run the following commands; this is for 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/xvde` disk device represents a 10 GB utility image file created with the following command (in the `/OVS/running_pool/host02` directory on **dom0**):
    ```
    # dd if=/dev/zero of=u03.img bs=1M count=10240
    ```
- The `/dev/xvda` device has Oracle Linux 7.1 installed – Server with GUI Base Environment.
- This system disk uses LVM volumes for the `root` and `swap` partitions.

c. Use the `ip addr` command to display the network interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:02 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.102/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.102/24 brd 192.168.1.255 scope global eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:04:02 brd ff:ff:ff:ff:ff:ff
```

- The system has four Ethernet network interfaces, `eth0`, `eth1`, `eth2`, and `eth3`.
- 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 11: OCFS2 and Oracle Clusterware."

- The `eth2` and `eth3` interfaces are configured as part of a bonded network interface in “Practices for Lesson 10: Advanced Networking.”

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

```
# 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
```

   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, 25165824 sectors
/dev/xvda1   *    2048    1026047     512000   83   Linux
/dev/xvda2     1026048   25165823   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/mapper/ol-root: 11.0 GB, 11022630912 bytes, ...
Disk /dev/mapper/ol-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 7.1 installed – Server with GUI Base Environment.
- This system disk uses LVM volumes for the `root` and `swap` partitions.

c. Use the `ip addr` command to display the network interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.103/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.104/24 brd 192.0.2.255 scope global eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:03:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth2
...
```

- 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 12: iSCSI and Multipathing."
- The `eth2` interface is on a private subnet, `192.168.1`, and is used in "Practices for Lesson 11: OCFS2 and Oracle Clusterware."

d. Use the `cat` command to view the `/etc/resolv.conf` file.

- This file provides access to Domain Name Service (DNS) for host-to-IP address resolution. It identifies three DNS nameservers and the search domain.

```
# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
nameserver 152.68.154.3
nameserver 10.216.106.3
```

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

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

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

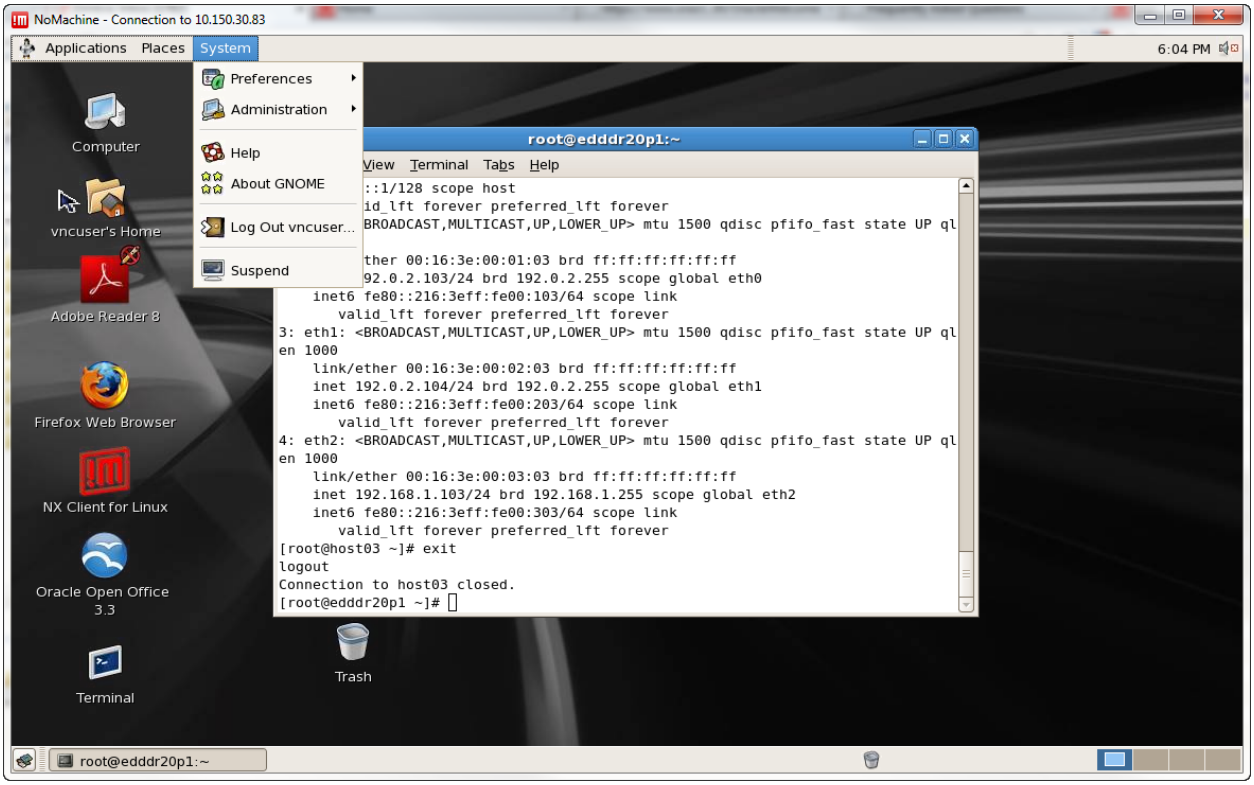
Practice 1-6: Logging Off from Your Student PC

Overview

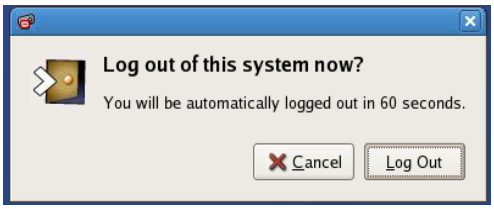
In this practice, you learn how to log off from your system.

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 can click the Log Out button to log out.
      - However, do not log out 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: Overview

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 and test the DNS server configuration on **dom0**
- Configure **host03** to as a caching-only nameserver
- 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.

   - In this example, only the `dhcp-libs` and `dhcp-common` packages are installed.

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

   b. Use the `yum list available` command, pipe the output to the `grep` command, and search for the string "`dhcp`".

   - Only partial output is shown.
   - The `dhcp.x86_64` package needs to be installed in this example.

```
# yum list available | grep dhcp
dhcp.x86_64 ...
dhcp-libs.i686 ...
```

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

   - You do not need to include the "`.x86_64`" extension.
   - Answer `y` when prompted "Is this ok".
   - 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
```

```
Total download size: 509 k
Installed size: 1.4 M
Is this ok [y/d/N]: y
...
Retrieving key from http://192.0.2.1/repo/OracleLinux/OL7/...
...
Is this ok [y/N]: y
...
Complete!
```

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

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

   - You can edit the `dhcpd.conf` file as follows by 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. Before enabling and starting the `dhcpd` service, specify a command-line argument to instruct the `dhcpd` service to only listen for DHCP requests on the `eth2` network interface.

   a. Use the `cp` command to copy the `dhcpd.service` file from the `/usr/lib/systemd/system/` directory to the `/etc/systemd/system/` directory.

      - The `/usr/lib/systemd/system/ systemd` units are included with the RPM packages and are not to be edited.
      - The `/etc/systemd/system/ systemd` units are created and managed by the system administrator and take precedence.

```
# cp /usr/lib/systemd/system/dhcpd.service /etc/systemd/system/
```

b. Use the `vi` editor to edit the `/etc/systemd/system/dhcpd.service` file and append `eth2` to the "`ExecStart`" line.

```
# vi /etc/systemd/system/dhcpd.service
...
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user
dhcpd -group dhcpd --no-pid eth2
...
```

5. Enable and start the `dhcpd` service.

a. Use the `systemctl` command to enable the `dhcpd` service to start at boot time.

- Note that a symbolic link is created for the `/etc/systemd/system/dhcpd.service` file.

```
# systemctl enable dhcpd
ln –s '/etc/systemd/system/dhcpd.service'
'/etc/systemd/system/multi-user.target.wants/dhcpd.service'
```

b. Use the `systemctl` command to start the `dhcpd` service.

```
# systemctl start dhcpd
```

c. Use the `systemctl` command to view the status of the `dhcpd` service.

- Note that the server is only listening on `eth2`.

```
# systemctl status dhcpd
dhcpd.service – DHCPv4 Server Daemon
   Loaded: loaded (/etc/systemd/system/dhcpd.service; enabled)
   Active: active (running) since ...
...
<date_time> host03...: Listening on LPF/eth2/00:16...
...
```

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: ...
[root@host01]#
```

2. Use the `rpm` command to verify that the `dhclient` package is installed on **host01**.
   - In this example, the package is already installed.

```
[host01]# rpm -q dhclient
dhclient-4.2.5-36.0.1.el7.x86_64
```

3. 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.

```
[host01]# vi /etc/sysconfig/network-scripts/ifcfg-eth1
TYPE=Ethernet
BOOTPROTO=dhcp
...
ONBOOT=yes
```

4. Use the `ip addr` command to display the network interfaces on **host01**.

   - Note that `eth1` does not have an IP address.

```
[host01]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP>   mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.101/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:01 brd ff:ff:ff:ff:ff:ff
...
```

5. From **host01**, request a lease for `eth1` from the DHCP server.

   a. Use the `dhclient` command to request a lease for `eth1` from the DHCP server.

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

b. Use the `ip addr` command on **host01** to verify that `eth1` obtained an IP address.
   - In this example, `eth1` now has an IP address of `192.168.1.200`.

```
[host01]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP>  mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.101/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.0.2.255 scope global eth1
...
```

6. View information about the lease.
   a. 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;
  option subnet-mask 255.255.255.0;
  ...
}
```

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

```
[host03]# cat /var/lib/dhcpd/dhcpd.leases
...
lease 192.168.1.200 {
  starts ...
  ends ...
  ...
  hardware ethernet 00:16:3e:00:02:01
}
```

7. 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: Viewing and Testing the DNS Configuration

Overview

In this practice, you:

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

Assumptions

- **Dom0** is already configured as an authoritative nameserver for the `example.com` domain.
- 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**.
   - In this example, the package is installed.

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

2. Ensure that the `named` service is enabled and running on **dom0**.
   - Use the `service` and `chkconfig` commands on **dom0** because **dom0** is running Oracle VM Server for x86 version 3.2.1
   - Use the `systemctl` command on the **host01**, **host02**, and **host03** virtual machines because the VMs are running Oracle Linux 7.1.

   a. Use the `service` command to verify that the `named` service is started on **dom0**.
      - In this example, the service is running.

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

b. Use the `chkconfig` command to verify that the `named` service is configured to start at boot time on **dom0**.
   - In this example, the service is configured to start when the system boots at either run level 2, 3, or 4.

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

3. View the DNS configuration on **dom0**.
   a. View the main BIND configuration file, `/etc/named.conf`.
      - This file lists location and characteristics of your domain's zone files.
      - Note that the zone file, `/var/named/data/master-example.com`, is defined.
      - Note that a reverse lookup zone file, `/var/named/data/reverse-192.0.2`, is also defined.

```
[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;
};
...
```

   b. View the `/var/named/data/master-example.com` zone file.
      - This file defines IPv4 addresses ("A" records) for the DNS server, the DNS domain, and the four VM guest systems.

```
[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
...
```

c. View the `/var/named/data/reverse-192.0.2` file.

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

```
[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.
```

Perform the next task from **host03**.

4. Test host name to IP resolution on **host03**.

a. Use the `ping` command to contact **host01** and **host02**.

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

```
[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
```

b. 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
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
```

c. Use the `ping` command to contact **host01** and **host02**.

- You can still successfully contact these systems by name, because DNS is resolving host names to IP addresses.

```
[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
```

d. Use the `grep` command to search for the "`hosts`" string in the `/etc/nsswitch.conf` file.

- The first "`hosts`" entry is a comment.
- In the second "`hosts`" entry, "`files`" means to use the local `/etc/hosts` file to resolve host names to IP addresses.
- Also in the second "`hosts`" entry, "`dns`" means to use DNS to resolve host names to IP addresses when unable to resolve by using the `/etc/hosts` file.

```
[host03]# grep hosts /etc/nsswitch.conf
#hosts:  db files nisplus nis dns
hosts:   files dns
```

e. Use the `vi` editor to edit the `/etc/nsswitch.conf` file and remove the "`dns`" argument from the "`hosts`" entry as follows.

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

f. Use the `ping` command to contact **host01** and **host02**.

- You cannot contact these systems by name now because DNS is no longer used.

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

g. Use the `vi` editor to edit the `/etc/nsswitch.conf` file and restore the "`dns`" argument to the "`hosts`" entry as follows.

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

h. Use the `ping` command to contact **host01** and **host02**.

- You can now successfully contact these systems by name, because DNS is resolving host names to IP addresses.

```
[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
```

i. View the `/etc/resolv.conf` file.

- DNS is only able to resolve host names to IP addresses because the `/etc/resolv.conf` file contains a valid search domain, `example.com`, and valid nameserver information.
- The `nameserver 192.0.2.1` for the `example.com` domain stores the zone files that provide host name to IP address resolution.

```
[host03]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
nameserver 152.68.154.3
nameserver 10.216.106.3
```

j. Use the `vi` editor to edit the `/etc/resolv.conf` file and comment out all lines as follows.

```
[host03]# vi /etc/resolv.conf
# Generated by NetworkManager
#search example.com
#nameserver 192.0.2.1
#nameserver 152.68.154.3
#nameserver 10.216.106.3
```

k. Use the `ping` command to contact **host01** and **host02**.

- You cannot contact these systems by name now.

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

l. Use the `vi` editor to edit the `/etc/resolv.conf` file and remove the `#` signs to uncomment the "`search`" and "`nameserver`" entries as follows.

```
[host03]# vi /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
nameserver 152.68.154.3
nameserver 10.216.106.3
```

m. Use the `ping` command to contact **host01** and **host02**.

- You can now successfully contact these systems by name, because DNS is resolving host names to IP addresses.

```
[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
```

5. Note that NetworkManager generates the `/etc/resolv.conf` entries on **host03**.

a. View the `/etc/resolv.conf` file.

- Note the commented line indicating that `NetworkManager` generated the `/etc/resolv.conf` file.

```
[host03]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
nameserver 152.68.154.3
nameserver 10.216.106.3
```

b. View the `/etc/sysconfig/network-scripts/ifcfg-eth0` file.

- Note that the `DNS[123]` entries in the `ifcfg-eth0` file correspond to the nameserver entries in the `resolv.conf` file.
- Note that the `DOMAIN` entry in the `ifcfg-eth0` file corresponds to the `search` entry in `resolv.conf`.
- NetworkManager uses the information in the `ifcfg-eth0` file to populate the `resolv.conf` file.

```
[host03]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
...
DNS1=192.0.2.1
DNS2=152.68.154.3
DNS3=10.216.106.3
DOMAIN=example.com
```

6. Use the `host` command to perform DNS lookups on **host03**.

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
```

d. Use the `-v` option to display verbose information about the `example.com` domain.

```
[host03]# host -v example.com
Trying "example.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65099
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ...

;; QUESTION SECTION:
;example.com.              IN      A

;; AUTHORITY SECTION:
example.com.       86400   IN      SOA   dns.example.com. ...
...
```

7. Use the `dig` command to perform DNS lookups on **host03**.

   Query DNS for the information about **host02.example.com**.

```
[host03]# dig host02.example.com
...
;; QUESTION SECTION.
;host02.example.com.            IN      A

;; ANSWER SECTION.
host02.example.com.  86400     IN      A  192.0.2.102

;; AUTHORITY SECTION.
example.com.         86400     IN      A  dns.example.com

;; ADDITIONAL SECTION.
dns.example.com.     86400     IN      A  192.0.2.1
...
```

Practice 2-4: Configuring a Caching-Only Nameserver

Overview

In this practice, you configure **host03** as a caching-only nameserver.

Assumptions

- You are the `root` user on **host03**.
- All commands in this practice with one exception are executed on **host03**.
- The one command that needs to be run on **dom0** includes **dom0** in the prompt.

Tasks

1. Install the `bind` software package on **host03**.
   a. Use the `rpm` command to determine if the `bind` package is already installed.
      - In this example, there are several package names that returned from the `rpm` command but the `bind` package is not installed.

```
# rpm -qa | grep bind
rpcbind-...
PackageKit-device-rebind-...
keybinder3-...
bind-utils-...
bind-libs-...
bind-license-...
bind-libs-lite-...
```

   b. Use the `yum` command to install the `bind` package.
      - Answer `y` when prompted "Is this ok".

```
# yum install bind
...
Transaction Summary
================================================================
Install  1 Package

Total download size: 1.8 M
Installed size: 4.3 M
Is this ok [y/d/N]: y
...
Complete!
```

2. View the BIND configuration files and directories.
   a. View the `/etc/named.conf` file.
      - This is the main BIND configuration file.
      - Note that the default BIND configuration files provide a caching-only nameserver.

- Note that only one zone is defined, whose name is a period (.).
- This zone is a `hint` zone type and specifies that the nameserver look in the `/var/named/named.ca` file for IP addresses of authoritative servers for the root domain when the nameserver starts or does not know which nameserver to query.
- The `/etc/named.conf` also includes the `/etc/named.rfc1912.zones` file.

```
# cat /etc/named.conf
...
// Provided by Red Hat bind package to configure the ISC BIND
// named(8) DNS server as a caching only nameserver ...
...
options {
          ...
          directory  "/var/named";
          ...
          /*
           - If you are building an AUTHORITATIVE DNS server,
             do NOT enable recursion.
           - If you are building an RECURSIVE (caching) DNS
             server, you need to enable recursion.
          recursion yes;
          ...
};

logging {
          ...
};

zone "." IN {
          type hint;
          file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

b. View the `/etc/named.rfc1912.zones` file.

- This is the base configuration file for implementing a caching-only nameserver.
- There are five zones defined in this file.
- Zone options are included for each of these five zones:
  - `type`: Specifies the zone type which is set to "master" for all five zones. Type "master" designates the nameserver as authoritative for this zone. A zone is set as master if the zone file resides on this system.

- `file`: Specifies the name of the zone file, which is stored in the working directory defined by the directory option (`/var/named` in this example)
- `allow-update`: Specifies which hosts are allowed to dynamically update information in their zone. Dynamic updates are set to none for these zones, meaning they are not allowed.

```
# cat /etc/named.rfc1912.zones
...
// Provided by Red Hat caching-nameserver package
...
zone "localhost.localdomain" IN {
        type master;
        file "named.localhost";
        allow-update { none; };
};

zone "localhost" IN {
        type master;
        file "named.localhost";
        allow-update { none; };
};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN {
        type master;
        file "named.loopback";
        allow-update { none; };
};

zone "1.0.0.127.in-addr.arpa" IN {
        type master;
        file "named.loopback";
        allow-update { none; };
};

zone "0.in-addr.arpa" IN {
        type master;
        file "named.empty";
        allow-update { none; };
};
```

c. View the `/var/named/` directory.

- This is the default directory in which zone files are stored.

```
# ls -l /var/named
total 16
drwxrwx--- ... named named ... data
drwxrwx--- ... named named ... dynamic
-rw-r----- ... root  named ... named.ca
-rw-r----- ... root  named ... named.empty
-rw-r----- ... root  named ... named.localhost
-rw-r----- ... root  named ... named.loopback
drwxrwx--- ... named named ... slaves
```

d. View the `/var/named/named.ca` file.

- This file contains a list of the 13 root authoritative DNS servers.

```
# cat /var/named/named.ca
...
a.root-servers.net   3600000  IN     A        198.41.0.4
...
b.root-servers.net   3600000  IN     A        192.228.79.201
c.root-servers.net   3600000  IN     A        192.33.4.12
d.root-servers.net   3600000  IN     A        199.7.91.13
...
e.root-servers.net   3600000  IN     A        192.203.230.10
f.root-servers.net   3600000  IN     A        192.5.5.241
...
g.root-servers.net   3600000  IN     A        192.112.36.4
h.root-servers.net   3600000  IN     A        128.63.2.53
...
i.root-servers.net   3600000  IN     A        192.36.148.17
...
j.root-servers.net   3600000  IN     A        192.58.128.30
...
k.root-servers.net   3600000  IN     A        193.0.14.129
...
l.root-servers.net   3600000  IN     A        199.7.83.42
...
m.root-servers.net   3600000  IN     A        202.12.27.33
...
```

3. Start a DNS caching-only nameserver on **host03**.

   a. Use the `vi` editor to add the following entry to the beginning of the list of nameservers in the `/etc/resolv.conf` file:

      `nameserver 127.0.0.1`

      - This line indicates use of the local system as the primary nameserver.

```
# vi /etc/resolv.conf
search example.com
nameserver 127.0.0.1            # add this line only
nameserver 192.0.2.1
...
```

   b. Use the `systemctl` command to enable the `named` service.

```
# systemctl enable named
ln -s '/usr/lib/systemd/system/named.service'
'/etc/systemd/system/multi-user.target.wants/named.service'
```

   c. Use the `systemctl` command to start the `named` service.

      - This command takes a few seconds to complete.

```
# systemctl start named
```

   d. From the second terminal window on **dom0**, `ssh` to **host03** as the `root` user, and monitor the journal in real time before proceeding to step 3e.

      - Monitoring the journal in real time allows you to see the host name to IP resolution occurring.
      - The `root` password on **host03** is `oracle`.
      - You might want to enlarge this window to see more of the journal entries.

```
[dom0]# ssh root@host03
root@host03's password: oracle
[root@host03 ~]# journalctl -f
-- Logs begin at ...
...
```

   e. In the original window, use the `ping` command to contact **host01** and **host02**.

      - You can now successfully contact these systems by name, because DNS is resolving host names to IP addresses.
      - Press Ctrl + C to exit after a few lines of output.

```
# 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
# 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
```

f. Notice the “resolving” messages in the journal window.

```
[root@host03 ~]# journalctl -f
-- Logs begin at ...
<date_time> host03... error (network unreachable) resolving ...
...
```

g. Use the `CTRL-C` command to stop the `journalctl -f` command.

```
# journalctl -f
...
CTRL-C
```

h. Use the `exit` command to log off **host03** from this second window.

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

4. In the first terminal window on **host03**, use the `rndc` command to obtain status of the named service.

```
# rndc status
Version: ...
CPUs found: 1
worker threads: 1
UDP listeners per interface: 1
number of zones: 101
debug level: 0
...
```

5. Stop the `named` service on **host03** and restore to original configuration.

a. Use the `systemctl` command to stop the `named` service.

```
# systemctl stop named
```

b. Use the `systemctl` command to disable the `named` service.

```
# systemctl disable named
rm '/etc/systemd/system/multi-user.target.wants/named.service'
```

c. Use the `vi` editor to remove the following entry from the `/etc/resolv.conf` file:

nameserver 127.0.0.1

```
# vi /etc/resolv.conf
search example.com
nameserver 127.0.0.1            # delete this line only
nameserver 192.0.2.1
...
```

d. Use the `vi` editor to edit the `/etc/hosts` file and remove the comment (`#` sign) from the entries previously commented out.

```
# 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
```

6. Log off **host03**.

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

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

Practices for Lesson 3: Authentication and Directory Services

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

In these practices, you configure:

- OpenLDAP server and enable LDAP authentication
- OpenLDAP client and log in as an LDAP user

Practice 3-1: Configuring an OpenLDAP Server

Overview

In this practice, you:

- Configure an OpenLDAP server in preparation to implement LDAP authentication
- Install the OpenLDAP packages and the `migrationtools` package
- Configure the `slapd.d` configuration database
- Configure the base domain and test the LDAP server
- Migrate users and groups into the LDAP directory
- Modify `firewalld` to allow access from LDAP clients

Assumptions

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

Tasks

1. Connect to **host03** by using `vncviewer`.

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

   - The sample shown indicates that the port number is `5904`. Your port number might be different.

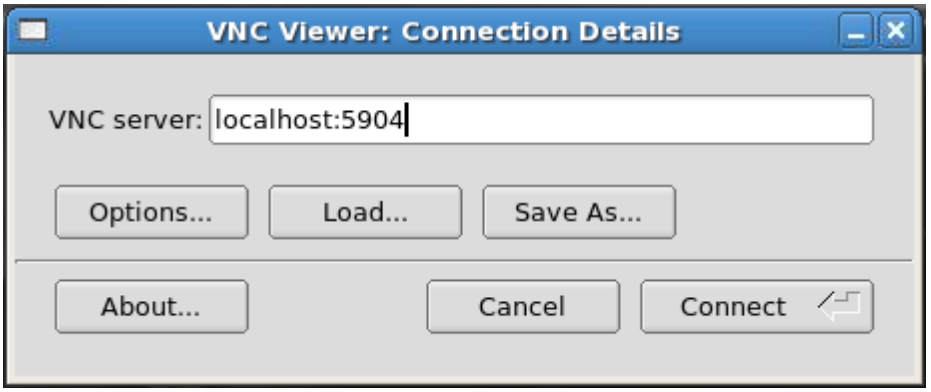
```
# xm list -l host03 | grep location
                (location 0.0.0.0:5904)
                (location 3)
```

   b. Run the `vncviewer&` command.

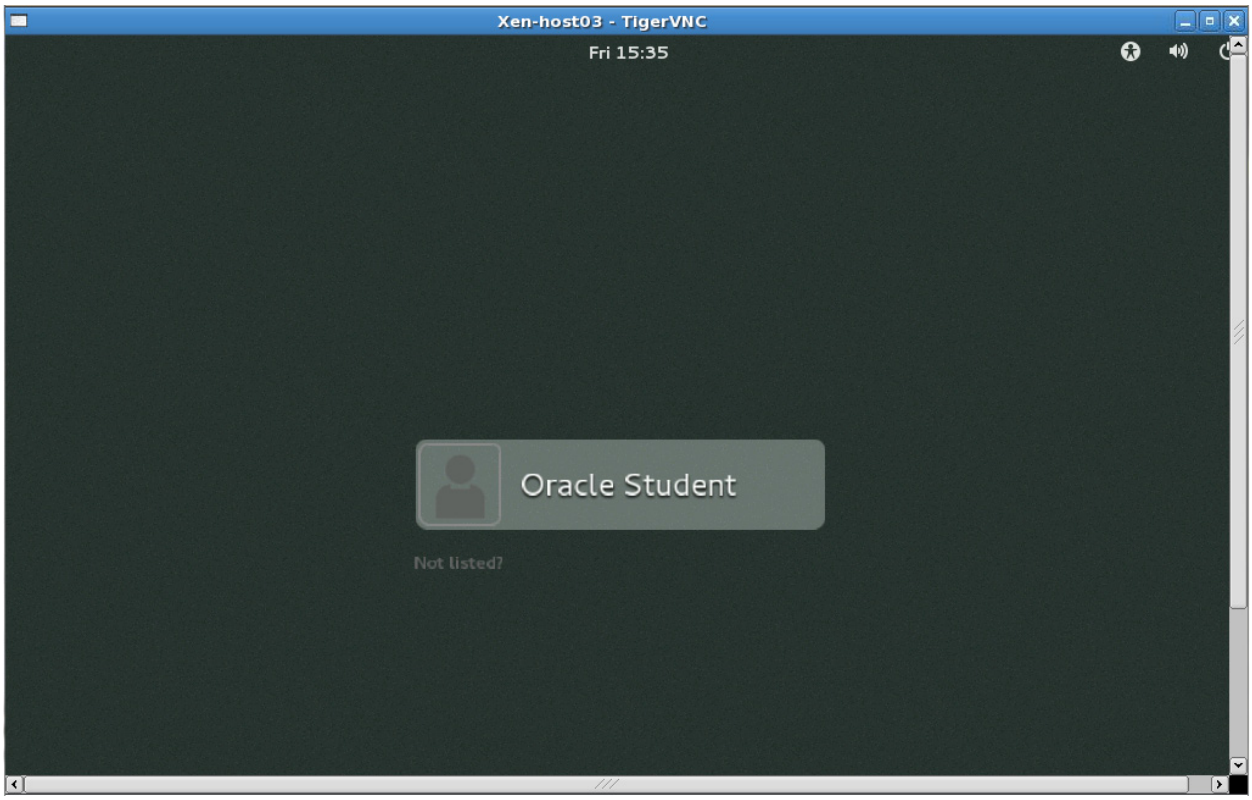
```
# vncviewer&
```

   - 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 host03 | grep location` command. For example, if the port number is `5904`, enter `localhost:5904` and click "Connect."



- The GNOME login screen appears. You might need to press Enter to display the login screen.

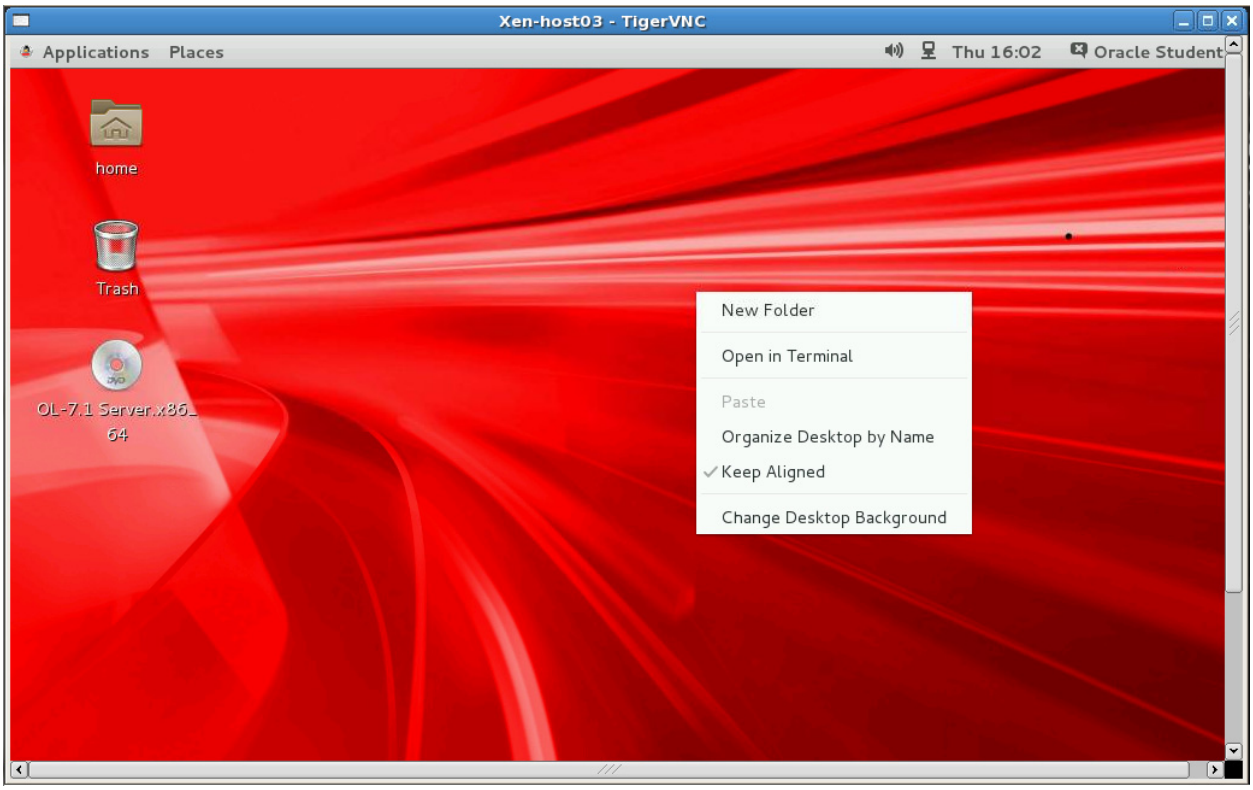


d. Click “Oracle Student” in the list of users. You are prompted for the Password.

e. Enter `oracle` for the Password and click “Sign In.”

- The GNOME desktop appears.

f. Right-click 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
#
```

2. Install the required RPM packages on **host03**.

Use the `yum` command to install the following packages:
- `openldap-servers`
- `openldap-clients`
- `migrationtools`

- Answer `y` when prompted "Is this ok".

```
# yum install openldap-servers openldap-clients migrationtools
...
Transaction Summary
================================================================
Install  3 Packages

Total download size: 2.3 M
```

```
Installed size: 5.3 M
Is this ok [y/d/N]: y
...
Complete!
```

3. Copy default DB_CONFIG template file.

   a. Use the ls command to view the contents of the /var/lib/ldap directory.
      - Note that the directory is empty.

```
# ls /var/lib/ldap
```

   b. Use the ls command to view the contents of the /usr/share/openldap-servers directory.
      - A default DB_CONFIG template file is installed in the /usr/share/openldap-servers directory.
      - The default DB_CONFIG template file name is DB_CONFIG.example file.

```
# ls /usr/share/openldap-servers
DB_CONFIG.example  slapd.ldif
```

   c. 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
```

   d. Use the ls -l command to list the contents of the /var/lib/ldap directory.
      - Note that the current owner and group is root.
      - Both the owner and group need to be changed to ldap.

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

   e. 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
```

   f. Use the ls -l command to show the new owner and group.
      - Note that the owner and group are now set to ldap.

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

4. Start the slapd service.

   a. Use the systemctl command to enable and start the slapd service.

```
# systemctl enable slapd
ln -s '/usr/lib/systemd/system/slapd.service'
'/etc/systemd/system/multi-user.target.wants/slapd.service'
# systemctl start slapd
```

b. Use the `ls -l` command to list the contents of the `/var/lib/ldap` directory.
   - Note that the initial database now exists.

```
# ls -l /var/lib/ldap
-rw-r--r--. 1 ldap ldap ... alock
-rw-------. 1 ldap ldap ... __db.001
-rw-------. 1 ldap ldap ... __db.002
-rw-------. 1 ldap ldap ... __db.003
-rw-r--r--. 1 ldap ldap ... DB_CONFIG
-rw-------. 1 ldap ldap ... dn2id.bdb
-rw-------. 1 ldap ldap ... id2entry.bdb
-rw-------. 1 ldap ldap ... log.0000000001
```

5. View 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.
      - Note 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.

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

   b. Use the `cd` command to change to the `slapd.d` directory.
      - Use the `ls -l` command to display the contents of the directory.

```
# cd slapd.d
# ls -l
drwxr-x---. 3 ldap ldap ... cn=config
-rw-------. 1 ldap ldap ... cd=config.ldif
```

   c. Use the `cd` command to change to the `cn=config` directory.
      - Use the `ls -l` command to display the contents of the configuration directory.

```
# cd cn=config
# ls -l
drwxr-x---. 2 ldap ldap ... cn=schema
-rw-------. 1 ldap ldap ... cn=schema.ldif
-rw-------. 1 ldap ldap ... olcDatabase={0}config.ldif
-rw-------. 1 ldap ldap ... olcDatabase={-1}frontend.ldif
-rw-------. 1 ldap ldap ... olcDatabase={1}monitor.ldif
-rw-------. 1 ldap ldap ... olcDatabase={2}hdb.ldif
```

6. Update the OpenLDAP configuration database domain component.
   - The default is "dc=my-domain,dc=com".
   - Change all occurrences to "dc=example,dc=com".

   Use the grep command to search for the "my-domain" string in all files in the configuration directory.
   - Note that the following files contain the "my-domain" string:
     - olcDatabase={1}monitor.ldif
     - olcDatabase={2}hdb.ldif
   - Change each occurrence of "my-domain" to "example".

```
# grep my-domain *
grep: cn=schema: Is a directory
olcDatabase={1}monitor.ldif: ,cn=auth" read by
dn.base="cn=Manager,dc=my-domain,dc=com" read by * none
olcDatabase={2}hdb.ldif:olcSuffix: dc=my-domain,dc=com
olcDatabase={2}hdb.ldif:olcRootDN: cn=Manager,dc=my-
domain,dc=com
```

7. Update the Database Suffix.
   a. Use the cat command to view the olcDatabase={2}hdb.ldif file.
      - The .ldif extension begins with the lowercase letter l, not the number 1.
      - Note the comment, "DO NOT EDIT!! Use ldapmodify."
      - Note that there are two parameters that contain the "dc=my-domain" string.
        - olcRootDN
        - olcSuffix

```
# cat olcDatabase={2}hdb.ldif
# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
...
olcRootDN: cn=Manager,dc=my-domain,dc=com
...
olcSuffix: dc=my-domain,dc=com
...
```

   b. Use the cp command to make a backup copy of the olcDatabase={2}hdb.ldif file.

```
# cp olcDatabase={2}hdb.ldif hdb_BAK
```

   c. Use the ldapmodify command to set the Database Suffix.
      - The -Q option means "Enable SASL Quiet mode. Never prompt."
        - SASL is "Simple Authentication and Security Layer".
        - It is a framework for authentication and data security in Internet protocols.
        - It decouples authentication mechanisms from application protocols, allowing you to use any authentication mechanism supported by SASL.

- The `-Y EXTERNAL` option specifies the SASL mechanism to be used for authentication.
  - A SASL mechanism implements a series of challenges and responses.
  - "`EXTERNAL`" means authentication is implicit in the context (for example, for protocols already using IPsec or Transport Layer Security).
- The `-H ldapi:///` option specifies URI(s) referring to the ldap server(s). Only the `protocol/host/port` fields are allowed. A list of URI separated by whitespace or commas is expected.
  - LDAPI allows LDAP connections to run over IPC connections, meaning the LDAP operations can run over UNIX sockets.
- After issuing the `ldapmodify` command, the prompt changes to `>`.
- Enter the entries in bold as shown.

```
# ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
> dn: olcDatabase={2}hdb,cn=config
> changetype: modify
> replace: olcSuffix
> olcSuffix: dc=example,dc=com
>
> EOF
```

- Press the Enter key after entering "`EOF`".
- This terminates the `ldapmodify` command and displays the following message:

```
Modifying entry "olcDatabase={2}hdb,cn=config"
```

d. Use the `ldapmodify` command to set the Database RootDN.

- After issuing the `ldapmodify` command, the prompt changes to `>`.
- Enter the entries in bold as shown.

```
# ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
> dn: olcDatabase={2}hdb,cn=config
> changetype: modify
> replace: olcRootDN
> olcRootDN: cn=Manager,dc=example,dc=com
>
> EOF
```

- Press the Enter key after entering "`EOF`".
- This terminates the `ldapmodify` command and displays the following message:

```
Modifying entry "olcDatabase={2}hdb,cn=config"
```

e. Use the `diff` command to view the differences between the `olcDatabase={2}hdb.ldif` file and the `hdb_BAK` file.

- Ensure the differences in `olcSuffix` and `olcRootDN` match the following.
  - If not, repeat steps 6c and 6d as needed to make the corrections.

- Ignore the other differences such as `entryCSN`, `modifiersName`, and `modifyTimestamp`.

```
# diff olcDatabase={2}hdb.ldif hdb_BAK
...
> olcSuffix: dc=my-domain,dc=com
> olcRootDN: cn=Manager,dc=my-domain,dc=com
...
> olcSuffix: dc=example,dc=com
> olcRootDN: cn=Manager,dc=example,dc=com
...
```

f. Use the `grep` command to search for the "`my-domain`" string in all files in this directory.

- Note that one database file still contains the "`my-domain`" string:
  - `olcDatabase={1}monitor.ldif`
- Ignore the occurrences in the `hdb_BAK` file.

```
# grep my-domain *
grep: cn=schema: Is a directory
hdb_BAK:olcSuffix: dc=my-domain,dc=com
hdb_BAK:olcRootDN: cn=Manager,dc=my-domain,dc=com
olcDatabase={1}monitor.ldif: ,cn=auth" read by
dn.base="cn=Manager,dc=my-domain,dc=com" read by * none
```

8. Update the Database Access.

a. Use the `cat` command to view the `olcDatabase={1}monitor.ldif` file.

- Note the comment to use `ldapmodify` to edit this file.
- Note that there is one parameter that contains the "`dc=my-domain`" string.
  - `olcAccess`
- The "`my-domain`" value for this `olcAccess` parameter needs to be changed to "`example`".

```
# cat olcDatabase={1}monitor.ldif
# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
...
olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
" read by dn.base="cn=Manager,dc=my-domain,dc=com" read by *
none
...
```

b. Use the `cp` command to make a backup copy of the `olcDatabase={1}monitor.ldif` file.

```
# cp olcDatabase={1}monitor.ldif monitor_BAK
```

c. Use the `ldapmodify` command to set the Database Access.

- After issuing the `ldapmodify` command, the prompt changes to `>`.
- Enter the entries in bold as shown.

```
# ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
> dn: olcDatabase={1}monitor,cn=config
> changetype: modify
> replace: olcAccess
> olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
" read by dn.base="cn=Manager,dc=example,dc=com" read by * none
>
> EOF
```

- Press the Enter key after entering "`EOF`".
- This terminates the `ldapmodify` command and displays the following message:

```
Modifying entry "olcDatabase={1}monitor,cn=config"
```

d. Use the `diff` command to view the differences between the `olcDatabase={1}monitor.ldif` file and the `monitor_BAK` file.

- Ensure the differences in `olcAccess` match the following.
  - If not, repeat step 7c to make the correction.
- Ignore the other differences such as `entryCSN`, `modifiersName`, and `modifyTimestamp`.

```
# diff olcDatabase={1}monitor.ldif monitor_BAK
...
> olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
" read by dn.base="cn=Manager,dc=my-domain,dc=com" read by *
none
...
> olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
" read by dn.base="cn=Manager,dc=example,dc=com" read by * none
...
```

e. Use the `grep` command to search for the "`my-domain`" string in all files in this directory.
   - Note that no database files now contain the "`my-domain`" string.
   - Ignore the occurrences in the "`_BAK`" files.

```
# grep my-domain *
grep: cn=schema: Is a directory
hdb_BAK:olcSuffix: dc=my-domain,dc=com
hdb_BAK:olcRootDN: cn=Manager,dc=my-domain,dc=com
monitor_BAK: ,cn=auth" read by dn.base="cn=Manager,dc=my-
domain,dc=com" read by * none
```

9. Create an encrypted user password.
   a. Use the `slappasswd` command to create an encrypted user password.
      - Enter a password of `oracle`.
      - Note that the encrypted password is displayed. This is a sample only; yours is different.

```
# slappasswd
New password: oracle
Re-enter new password: oracle
{SSHA}CsLkwW6B9+yBlzrGuHBdIT0z2Mj4q4l+
```

   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.

```
[root@host03 cn=config]# slappasswd
New password:
Re-enter new password:
{SSHA}CsLkwW6B9+yBlzrGuHBdIT0z2Mj4q4l+
[root@host03 cn=config]#
```

10. Use the `ldapmodify` command to set the `olcRootPW` directive.
    - You are adding this new directive to the `olcDatabase={2}hdb.ldif` file.
    - After issuing the `ldapmodify` command, the prompt changes to `>`.
    a. Enter the entries in bold as shown.
       - Ensure that you include a space after the "`olcRootPW:` " directive.

```
# ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
> dn: olcDatabase={2}hdb,cn=config
> changetype: modify
> add: olcRootPW
> olcRootPW:
```

b. Paste the encrypted password from the buffer by selecting Edit > Paste from the terminal window menu.
   - The "olcRootPW:" directive appears as follows:

```
> olcRootPW: {SSHA}CsLkwW6B9+yBlzrGuHBdIT0z2Mj4q4l+
```

c. Press the Enter key twice to add a blank line.
   - The final entry is "EOF". The complete list of commands is shown:

```
# ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
> dn: olcDatabase={2}hdb,cn=config
> changetype: modify
> add: olcRootPW
> olcRootPW: {SSHA}CsLkwW6B9+yBlzrGuHBdIT0z2Mj4q4l+
>
> EOF
```

d. Press the Enter key after entering "EOF".
   - This terminates the ldapmodify command and displays the following message:

```
Modifying entry "olcDatabase={2}hdb,cn=config"
```

11. Load the standard schemas.
   - The standard schemas are provided as LDIF files, which can be loaded by using the ldapadd command.
   - The standard schema files are located in the /etc/openldap/schema directory.

a. Use the ls command to view the contents of the /etc/openldap/schema directory.
   - Each one is offered in both the original LDAP schema form and in LDIF.

```
# ls /etc/openldap/schema
collective.ldif    cosine.schema  java.ldif    openldap.schema
collective.schema  duaconf.ldif   java.schema  pmi.ldif
...
```

b. Use the ldapadd command to load the following schemas.
   - core, cosine, inetorgperson, nis
   - These four schemas define the basic objects and attributes needed to describe a typical organization.
   - Use the -f <*filename*> option for each schema.
   - Ignore any "Duplicate attributeType" messages.

```
# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/schema/core.ldif
adding new entry "cn=core,cn=schema,cn=config"

# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/schema/cosine.ldif
adding new entry "cn=cosine,cn=schema,cn=config"
```

```
# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/schema/inetorgperson.ldif
adding new entry "cn=inetorgperson,cn=schema,cn=config"

# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f
/etc/openldap/schema/nis.ldif
adding new entry "cn=nis,cn=schema,cn=config"
```

12. 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 1005 -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.
       - Ignore the "BAD PASSWORD" warning, continuing to use `password` as the password.

```
# passwd student1
Changing password for user student1.
New password: password
BAD PASSWORD: The password fails the dictionary check ...
Retype new password: password
passwd: all authentication tokens updated successfully.
```

    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.
       - The output shows that the GID for the `students` group is `1008`.

```
# tail /etc/group
...
students:x:1008:
```

    e. Use the `usermod` command to add `oracle`, `student1`, and `student2` users to the `students` group.
       - Repeat the `tail /etc/group` command to view the changes.

```
# usermod -aG 1008 oracle
# usermod -aG 1008 student1
# usermod -aG 1008 student2
# tail /etc/group
```

```
...
students:x:1008:oracle,student1,student2
```

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

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

```
# cd /etc/openldap
```

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

    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 by 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
```

    c. Use the `ldapadd` command to add the base information to the LDAP directory.

    - The `-x` option uses simple authentication instead of SASL.
    - The `-W` option prompts for simple authentication. This is used instead of specifying the password on the command line.
    - The `-D "cn=Manager,dc=example,dc=com"` option uses the Distinguished Name (DN) to bind to the LDAP directory. For SASL binds, the server ignores this option.

- 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"
```

d. 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
```

14. 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, change the value of `$DEFAULT_MAIL_DOMAIN` from `padl.com` to `example.com`.

```
$DEFAULT_MAIL_DOMAIN = "padl.com";                (old value)
$DEFAULT_MAIL_DOMAIN = "example.com";             (new value)
```

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

```
$DEFAULT_BASE = "dc=padl,dc=com";                 (old value)
$DEFAULT_BASE = "dc=example,dc=com";              (new value)
```

    d. Save the `migrate_common.ph` file and exit `vi`.

15. Migrate the users.
    a. Use the `grep` command to list users in the `/etc/passwd` file with UID in the `1000-1009` range.
       - The purpose of step 12 was to populate this file as shown.
       - Do not be concerned if your entries do not match exactly.

```
# grep ":100[0-9]" /etc/passwd
oracle:x:1000:1000:Oracle Student:/home/oracle:/bin/bash
student1:x:1001:1001:Oracle Student1:/home/student1:/bin/bash
student2:x:1005:1005:Oracle Student2:/home/student2:/bin/sh
student3:x:1006:1006:Oracle Student3:/home/student3:/bin/sh
new_user:x:1007:1007::/home/new_user:/bin/bash
```

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

```
# grep ":100[0-9]" /etc/passwd > passwd
```

    c. Run the `migrate_passwd.pl` command to migrate user information in the `passwd` file into an LDIF format.
       - Redirect the output to `users.ldif`.
       - Use the absolute path name with the command 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=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: 1000
gidNumber: 1000
homeDirectory: /home/oracle
gecos: Oracle Student

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

16. Migrate the user groups.

    a. Use the `grep` command to list groups in the `/etc/group` file with GID in the `1000-1009` range.
    - This was the purpose of step 12, to populate this file as shown.
    - Do not be concerned if your entries do not match exactly.

```
# grep ":100[0-9]" /etc/group
oracle:x:1000:oracle
student1:x:1001:
student2:x:1005:
student3:x:1006:
new_user:x:1007:
students:x:1008:oracle,student1,student2
```

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

```
# grep ":100[0-9]" /etc/group > group
```

    c. Run the `migrate_group.pl` command to migrate group information in the `group` file into an LDIF format.
    - Redirect the output to `group.ldif`.
    - Use the absolute path name with the command 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=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: 1008
memberUid: oracle
memberUid: student1
memberUid: student2

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

17. Trust the LDAP service for `firewalld`.

a. Use the `firewall-cmd` command to permanently permit access by LDAP clients for the `public` zone.

```
# firewall-cmd --permanent --zone=public --add-service=ldap
success
```

b. Use the `systemctl` command to restart the `firewalld` service.

```
# systemctl restart firewalld
```

c. Use the `firewall-cmd` command to list everything for the active zone.

- Note that the `ldap` service is trusted.

```
# firewall-cmd --list-all
public (default, active)
...
  services: dhcpv6-client ldap ssh
...
```

Practice 3-2: 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**, use the `yum` command to install the `authconfig-gtk` software package.
   - This package provides the `system-config-authentication` utility.
   - Answer `y` when prompted "Is this ok".

```
# yum install authconfig-gtk
...
Transaction Summary
============================================================
Install  1 Package

Total download size: 105 k
Installed size: 247 k
Is this ok [y/d/N]: y
...
Complete!
```

2. Open the Authentication Configuration Tool by running the `system-config-authentication` command.

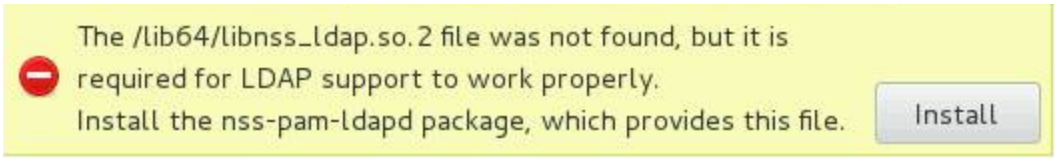
```
# system-config-authentication
```

- The GUI appears as follows.

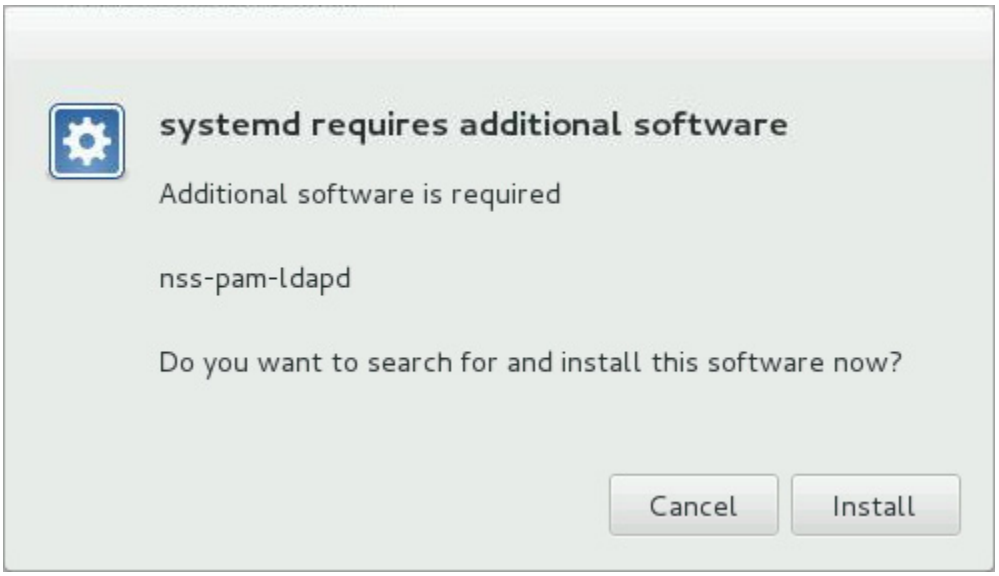
3. Make the following changes.

a. Select `LDAP` from the User Account Database drop-down list.

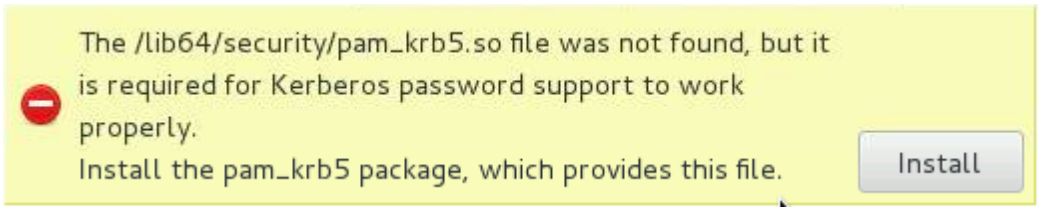
- The following dialog box is displayed.



b. Click “Install.”
   - The following dialog box is displayed.



c. Click “Install.”
   - The following dialog box is displayed.



   - Do not click “Install.” This dialog box closes when you select `LDAP password` as the Authentication Method in the next step.

d. Continue entering the following information.
   - Enter `dc=example,dc=com` as the LDAP Search Base DN.
   - Enter `ldap://host03.example.com` as the LDAP Server.
   - Click “Use TLS to encrypt connections.”
   - Select `LDAP password` as the Authentication Method.

- Ensure that your screen is configured as follows.

e. Click “Apply” to save your changes.

- After a few seconds, the Authentication Configuration Tool closes.

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 3-3: 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.

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

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

b. Use the `grep` command to search for user `student1` in the local `/etc/passwd` file.
   - The command produces no output indicating `student1` is not a local user on **host01**.

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

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".
      - 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

Total download size: 183 k
Installed size: 575 k
Is this ok [y/d/N]: y
...
Retrieving key from http://192.0.2.1/repo/OracleLinux/OL7/1/...
...
Is this ok [y/N]: y
...
Complete!
```

   b. Use the `yum` command to install the `nss-pam-ldapd` package.
      - Answer `y` when prompted "Is this ok".
      - Note that the `nscd` package is installed as a dependency.

```
# yum install pam_ldap
...
Transaction Summary
================================================================
Install   1 Package (+1 Dependent package)

Total download size: 413 k
Installed size: 586 k
Is this ok [y/d/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.
   - Change the IP address for the `URI` directive to the IP address of **host03**.

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

5. Configure the `/etc/nslcd.conf` file on **host01**.
   - This is the configuration file for the Naming Services LDAP Client Daemon.

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

```
# cd /etc
```

   b. Use the `vi` editor to edit the `nslcd.conf` file.
      - Use the `:set nu` command to turn on line numbers.

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

   c. At around line number 18, make the following change.

```
uri           ldap://127.0.0.1/                (old value)
uri           ldap://192.0.2.103/              (new value)
```

   d. At around line number 25, view the "`base`" setting.
      - You do not need to change the "`base`" setting.

```
base          dc=example,dc=com
```

   e. Save the `/etc/nslcd.conf` file and exit `vi`.

6. 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.
      - This backup file is used later in this practice to restore the original configuration.

```
# cp system-auth system-auth.BAK
```

   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 by 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 6g.

```
# vi system-auth
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is ...
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 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 < 1000 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_pwquality.so try_first_pass ...
password    sufficient    pam_unix.so sha512 shadow nullok ...
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     optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so ...
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`.

7. 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 `sss` and add `ldap` to the `passwd`, `shadow`, and `group` directives as shown.

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

c. Save the file and exit `vi`.

8. 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)
```

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

```
# systemctl start nslcd
```

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

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

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

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

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

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

```
# ls /home
```

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: 1001
gidNumber: 1001
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`.
- 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`.

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

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

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

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

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

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

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

```
# systemctl stop nslcd
```

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

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

c. Use the `vi` editor to replace `ldap` with `sss` for the `passwd`, `shadow`, and `group` directives as shown.

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

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

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

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

- This confirms OpenLDAP client authentication is disabled.

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

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

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

Perform the next step from **host03**.

12. 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 a few seconds, the Authentication Configuration Tool closes.

d. Use the `systemctl` command to stop and disable the `slapd` service.

```
# systemctl stop slapd
# systemctl disable slapd
rm '/etc/systemd/system/multi-user.target.wants/slapd.service'
```

Do not log off **host03**. The next practice (Practice 4-1) assumes that you are logged in as the `root` user on **host03**.

Practices for Lesson 4: Pluggable Authentication Modules (PAM)

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In these practices, you configure PAM authentication modules first to allow a single login only, and then to disable all non-`root` logins.

Practice 4-1: Configuring PAM for a Single Login Session

Overview

In this practice, you configure a PAM authentication module on **host03** to allow only a single login session for a user.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- You open a terminal window on each system.
- You log in as the `root` user on **host03**.
- The prompts in the solution section include either **host01** or **host03** to indicate which system to enter the command from.

Tasks

1. On **host03**, view PAM configuration files and directories.
   a. Use the `ls` command to view the PAM configuration directory, `/etc/pam.d`.
      - This directory contains files that describe the authentication procedure for an application.

```
[host03]# ls /etc/pam.d
atd     gdm-pin          ppp       sudo
chfn    gdm-smartcard    remote    sudo-i
...
```

   b. Use the `cat` command to view the `sshd` configuration file in `/etc/pam.d`.
      - This file contains a group of directives that define the authentication modules as well as any controls or arguments.
      - The authentication modules are listed in the third column.

```
[host03]# cat /etc/pam.d/sshd
#%PAM-1.0
auth       required     pam_sepermit.so
auth       substack     password-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be
executed in the user context
session    required     pam_selinux.so open env_params
session    optional     pam_keyinit.so force revoke
```

```
session    include     password-auth
session    include     postlogin
```

c. Use the `find` command to locate the `pam_sepermit.so` authentication module.

- In this example, the authentication module is located in `/usr/lib64/security`.

```
[host03]# find / -name pam_sepermit.so
/usr/lib64/security/pam_sepermit.so
```

d. Use the `ls` command to view the authentication modules directory.

- Note that all authentication modules are located in this directory.

```
[host03]# ls /usr/lib64/security
pam_access.so    pam_limits.so         pam_smbpass.so
pam_cap.so       pam_listfile.so       pam_sss.so
...
pam_sepermit.so
...
```

2. On **host03**, view the man pages for the `pam_sepermit` authentication module and the associated configuration file.

a. Most of the authentication modules have a man page describing their purpose and usage. Use the `man pam_sepermit` command to view the man page for the `pam_sepermit` authentication module.

- Note that this module uses a configuration file, `sepermit.conf`, which controls access when SELinux is in enforcing mode.
- SELinux stands for “Security-Enhanced Linux” and is covered in a subsequent lesson.

```
[host03]# man pam_sepermit
...
    pam_sepermit - PAM module to allow/deny login depending
    On SELinux enforcement state
...
    When the user which is logging in matches an entry in the
    config file he is allowed access only when the SELinux
    is in enforcing mode. Otherwise he is denied access...


...
    See sepermit.conf(5) for details.
...
```

b. Use the `man sepermit.conf` command to view the man page for the `sepermit.conf` file.

```
[host03]# man sepermit.conf
...
    sepermit.conf - configuration file for the pam_sepermit
module
...
    The lines of the configuration file have the following
syntax:
...
```

3. SELinux is covered in a subsequent lesson but for the purposes of this practice, use the `sestatus` command to display information about SELinux.
   - The output shown is a sample showing that SELinux is enabled and is in enforcing mode.
   - With SELinux in enforcing mode, the `pam_sepermit` authentication module allows or denies login.

```
[host03]# sestatus
SELinux status:    enabled
...
Current mode:      enforcing
...
```

4. From **host01**, confirm you can remotely log in to **host03**.
   a. From **dom0**, use the `ssh` command to log in to **host01** as the `oracle` user.
      - The password is `oracle`.

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

   b. From **host01**, use the `ssh` command to connect to **host03**.
      - Answer `yes` to "Are you sure".
      - The password is `oracle`.

```
[oracle@host01 ~]$ ssh host03
The authenticity of host 'host03 (192.0.2.103)' can't be ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
...
oracle@host03's password: oracle
Last login:...
[oracle@host03 ~]$
```

c. Use the `hostname` command to confirm that you successfully logged in to **host03**.
   - Note that you are successfully able to log in to **host03**.

```
[oracle@host03 ~]$ hostname
host03.example.com
```

d. Use the `logout` command to close the connection to **host03**.
   - Note that you are now logged off of **host03** and back to **host01**.

```
[oracle@host03 ~]$ logout
Connect to host03 closed.
[oracle@host01 ~]$ hostname
host01.example.com
```

5. On **host03**, configure the `pam_sepermit` authentication module to deny login.
   a. Use the `find` command to locate the `sepermit.conf` file.
      - Note that the `sepermit.conf` file is located in the `/etc/security` directory.

```
[host03]# find / -name sepermit.conf
/etc/security/sepermit.conf
```

   b. Use the `vi` editor to add the following entry to `/etc/security/sepermit.conf`.
      - This entry, when read by the PAM module `pam_sepermit.so`, allows only a single login session for the `oracle` user.

```
[host03]# vi /etc/security/sepermit.conf
oracle:exclusive
```

6. From **host01**, attempt to log in to **host03**.
   a. Use the `ssh` command to connect to **host03**. Password is `oracle`.
      - Note that the connection is denied.

```
[oracle@host01 ~]$ ssh host03
oracle@host03's password: oracle
Permission denied, please try again.
oracle@host03's password: CTRL-C
[oracle@host01 ~]$
```

   b. From **host03**, use the `tail` command to view the latest entries in the `/var/log/secure` log file.
      - Note that the connection is denied by the PAM authentication module, `pam_sepermit`.

```
[host03]# tail /var/log/secure
...
<date_time> host03 sshd[...]: pam_sepermit(sshd:auth): User
oracle processes are running. Exclusive login not allowed
...
```

   To permit the `oracle` user login from **host01**, you can do either of the following:
   - Remove the entry in the `/etc/pam.d/sshd` file to use the `pam_sepermit.so` module.

- Remove the entry in the `/etc/security/sepermit.conf` file to allow only a single login session.

7. From **host03**, permit user `oracle` to log in from **host01** by using the `vi` editor to comment out the entry to use the `pam_sepermit.so` module from the `/etc/pam.d/sshd` file.
   - Comment out this line by inserting a `#` sign at the beginning of the line as follows:

```
[host03]# vi /etc/pam.d/sshd
auth   required   pam_sepermit.so          (current entry)
#auth   required   pam_sepermit.so         (insert # sign)
```

8. From **host01**, attempt to log in to **host03**.
   a. Use the `ssh` command to connect to **host03**.
      - Password is `oracle`.
      - Note that the connection is allowed, and no longer denied by the PAM authentication module.

```
[oracle@host01 ~]$ ssh host03
oracle@host03's password: oracle
Last failed login: ...
[oracle@host03 ~]$ hostname
host03.example.com
```

   b. Use the `logout` command to close the connection to **host03**.
      - Note that you are now logged off of **host03** and back to **host01**.

```
[oracle@host03 ~]$ logout
Connect to host03 closed.
[oracle@host01 ~]$ hostname
host01.example.com
```

   c. From **host01**, log out as `oracle` user.

```
[oracle@host01 ~]$ logout
Connect to host01 closed.
```

9. Return **host03** back to the original state.
   a. From **host03**, use the `vi` editor to edit `/etc/pam.d/sshd` and uncomment the entry to use the `pam_sepermit.so` module (remove the `#` sign).

```
[host03]# vi /etc/pam.d/sshd
#auth   required   pam_sepermit.so          (current entry)
auth   required   pam_sepermit.so          (remove # sign)
```

   b. From **host03**, use the `vi` editor to edit `/etc/security/sepermit.conf` and remove the entry to allow only a single login for user `oracle`.

```
[host03]# vi /etc/security/sepermit.conf
~~oracle:exclusive~~                       (delete this entry)
```

Practice 4-2: Configuring PAM to Prevent Non-root Login

Overview

In this practice, you configure a PAM authentication module on **host01** to prevent all non-root user logins.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- Open a terminal window on each system.
- Log in as the root user on **host01**.
- The prompts in the solution section include either **host01** or **host03** to indicate which system to enter the command from.

Tasks

1. On **host01**, configure a PAM authentication module on **host01** to prevent all non-root user logins.
   a. From **dom0**, use the ssh command to log in to **host01** as root.
      - Password is oracle.

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

   b. Use the cat command to view the login configuration file in /etc/pam.d.
      - The login utility uses the pam_nologin.so authentication module as well as several other PAM modules.

```
[host01]# cat /etc/pam.d/login
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=...
auth       substack    system-auth
auth       include     postlogin
account    required    pam_nologin.so
...
```

   c. Use the man pam_nologin command to view the man page for the pam_nologin authentication module.
      - Note that this module uses a configuration file – /etc/nologin – which, if it exists, disables non-root logins.

```
[host01]# man pam_nologin
...
    pam_nologin - Prevent non-root users from login
...
    pam_nologin is a PAM module that prevents users from
    logging into the system when /var/run/nologin or
```

```
    /etc/nologin exists. The contents of the file are displayed
    to the user...no effect on the root user's ability to ...
...
```

d. Use the `vi` editor and create the `/etc/nologin` file with the following contents:

```
[host01]# vi /etc/nologin
No logins allowed at this time.
```

2. From **host03**, attempt to log in to **host01**.

a. Use the `ssh` command to connect to **host01** as user `oracle`.

- Answer `yes` to "Are you sure".
- The password is `oracle`.
- Note that the connection is denied.

```
[host03]# ssh oracle@host01
The authenticity of host 'host01 (192.0.2.101)' can't be ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
...
oracle@host01's password: oracle
No logins allowed at this time.

Connection closed by 192.0.2.101
```

b. From **host01**, use the `tail` command to view the latest entries in the `/var/log/secure` log file.

- Note that the connection is denied by the PAM authentication module.

```
[host01]# tail /var/log/secure
...
<date_time> host01 sshd[...]: fatal: Access denied for user
oracle by PAM account configuration [preauth]
```

To permit the non-`root` user logins, you can do either of the following:

- Delete the `/etc/nologin` file from **host01**.
- Remove the entry in the `/etc/pam.d/login` file to use the `pam_nologin.so` module.

3. From **host01**, permit non-`root` user logins from **host03** by using the `vi` editor to comment out the entry to use the `pam_nologin.so` module from the `/etc/pam.d/login` file.

- Comment out this line by inserting a `#` sign at the beginning of the line as follows:

```
[host01]# vi /etc/pam.d/login
...
account    required    pam_nologin.so          (current entry)
#account    required    pam_nologin.so         (insert # sign)
```

4. From **host03**, attempt to log in to **host01**.

Use the `ssh` command to connect to **host01** as user `oracle`.

- Note that the connection is still denied.

```
[host03]# ssh oracle@host01
oracle@host01's password: oracle
No logins allowed at this time.

Connection closed by 192.0.2.101
```

5. From **host01**, use the `grep` command to search for the string "`pam_nologin`" in all the files in the `/etc/pam.d` directory.
   - Note that this module also is called from the `ppp`, `remote`, and `sshd` files.
   - Because you are using `ssh` to log in, you would need to comment out the line in the `sshd` file as well.
   - Alternatively, remove the `/etc/nologin` file to allow non-`root` logins.

```
[host01]# grep pam_nologin /etc/pam.d/*
/etc/pam.d/login:#account   required   pam_nologin.so
/etc/pam.d/ppp:account   required   pam_nologin.so
/etc/pam.d/remote:account   required   pam_nologin.so
/etc/pam.d/sshd:account   required   pam_nologin.so
```

6. Return **host01** back to the original state.
   a. Use the `rm` command to remove the `/etc/nologin` file.

```
[host01]# rm /etc/nologin
rm: remove regular file '/etc/nologin'? y
```

   b. Use the `vi` editor to edit `/etc/pam.d/login` and uncomment the entry to use the `pam_nologin.so` module (remove the # sign).

```
[host01]# vi /etc/pam.d/login
...
#account   required   pam_nologin.so       (current entry)
account   required   pam_nologin.so        (remove # sign)
```

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

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

Do not log off **host03**. The next practice (Practice 5-1) assumes that you are logged in as the `root` user on **host03**.

Practices for Lesson 5: Web and Email Services

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In these practices, you configure the Apache Web Server.

Practice 5-1: Configuring the 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 connected to **host03** by using `vncviewer`.
- You are the `root` user on **host03** VM.

Tasks

1. Install the `httpd` software package and enable and start the `httpd` service.
   a. Use the `yum` command to install the `httpd` package.
      - Answer `y` to "Is this ok".

```
# yum install httpd
...
Transaction Summary
================================================================
Install  1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 4.3 M
Is this ok [y/d/N]: y
...
Complete!
```

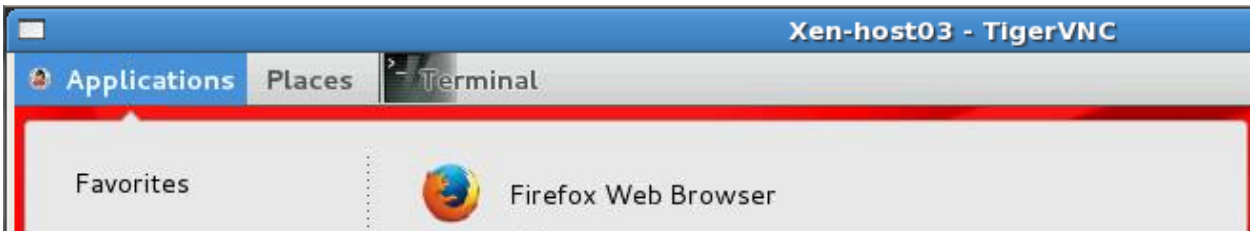
   b. Use the `systemctl` command to enable the `httpd` service to start at boot time.

```
# systemctl enable httpd
ln -s '/usr/lib/systemd/system/httpd.service'
'/etc/systemd/system/multi-user.target.wants/httpd.service'
```

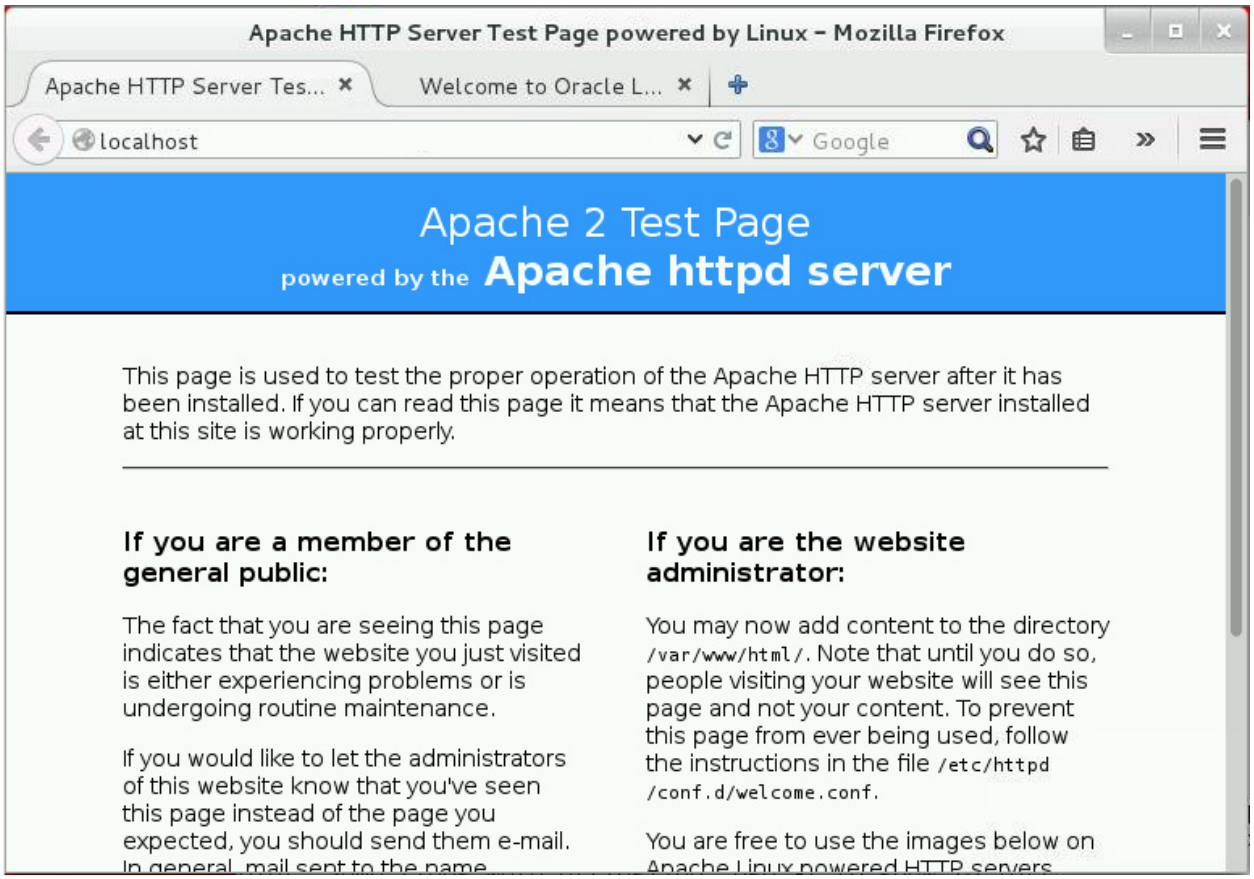
   c. Use the `systemctl` command to start the `httpd` service.

```
# systemctl start httpd
```

2. Confirm that Apache is working, by pointing a browser on **host03** to http://localhost.
   a. On the GNOME menu bar, click “Applications” to view the drop-down menu.
      - Under “Favorites,” select the “Firefox Web Browser” icon to start the Firefox web browser.



   b. Enter http://localhost in the browser and press Enter.
      - The Apache Test Page appears and confirms that Apache is working correctly.



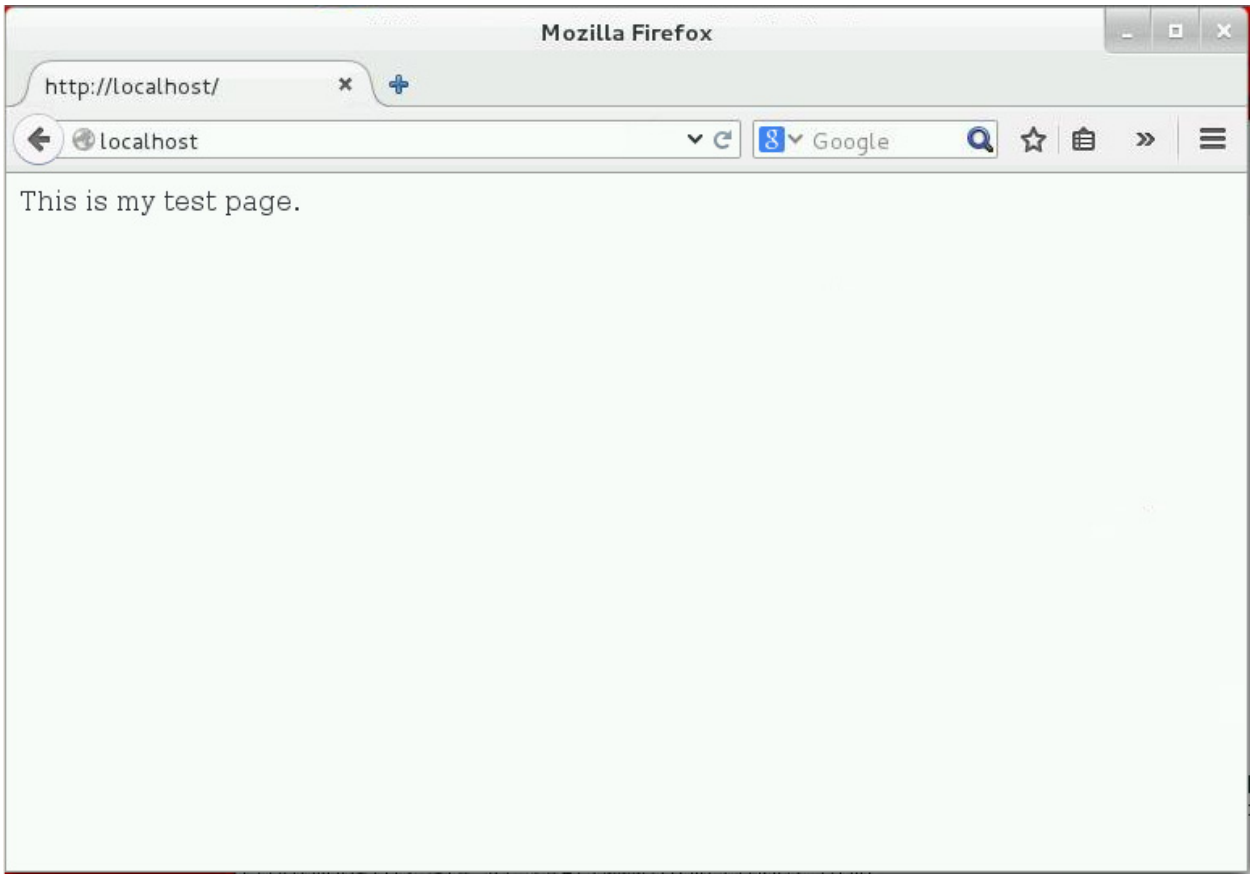
   c. Close the Firefox web browser by clicking the “X” in the top-right corner of the window.
      - A “Confirm close” dialog box might appear. If so, click the “Close tabs” button to close the window.

3. Create and view a test webpage.
   a. Use the `vi` editor to create the `/var/www/html/index.html` file with the following entry:

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

b. Restart the Firefox browser and point to http://localhost.

- The test webpage appears.



c. Close the Firefox web browser by clicking the "X" in the top-right corner of the window.

4. 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:

```
# vi /etc/httpd/conf/httpd.conf
<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 `apachectl configtest` command to check the configuration file for possible errors.

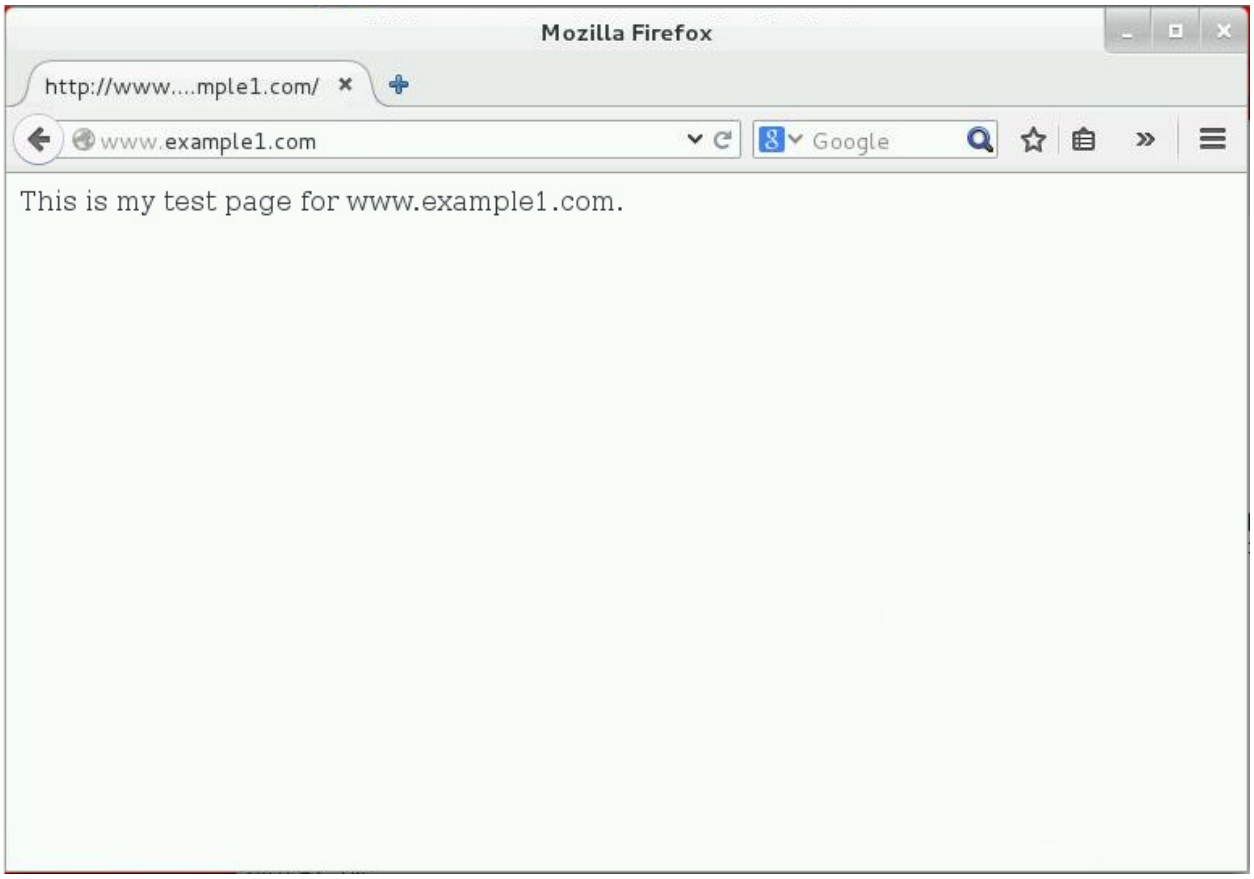
- In this example there are no errors.
- Fix any errors you might have made.

```
# apachectl configtest
Syntax OK
```

g. Use the `apachectl graceful` command to reload the configuration without affecting active requests.

```
# apachectl graceful
```

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



   b. Close the Firefox web browser by clicking the "X" in the top-right corner of the window.
6. 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 `apachectl configtest` command to check the configuration file for possible errors.

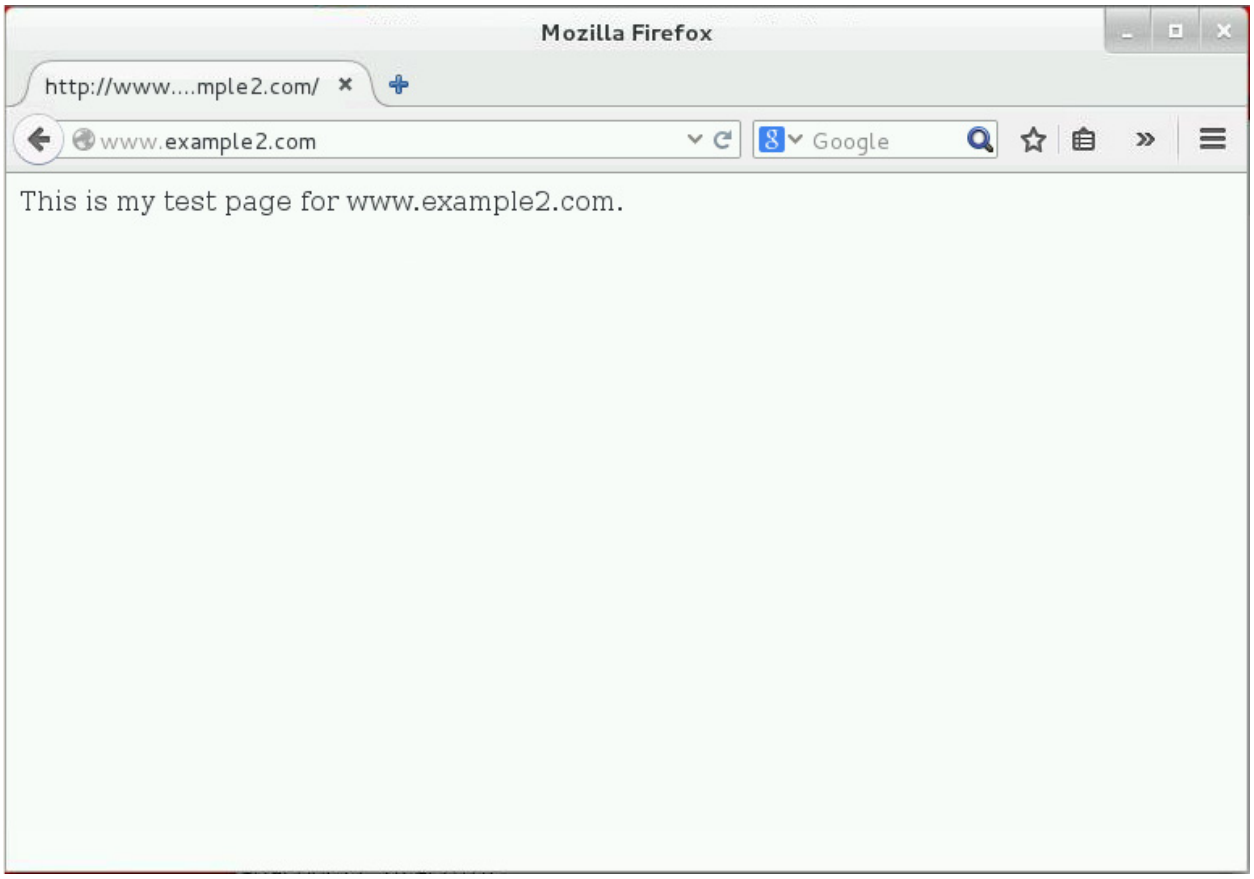
- In this example there are no errors.
- Fix any errors you might have made.

```
# apachectl configtest
Syntax OK
```

g. Use the `apachectl graceful` command to reload the configuration without affecting active requests.

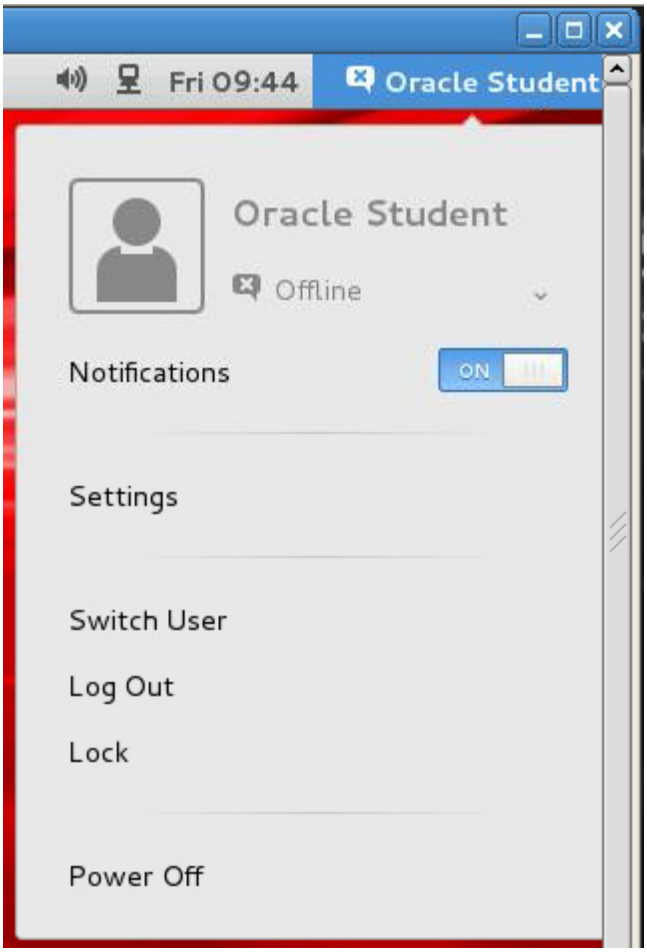
```
# apachectl graceful
```

7. View the test webpage for www.example2.com.
   a. Restart the Firefox browser and point to `http://www.example2.com`.
      - The test webpage for www.example2.com appears.

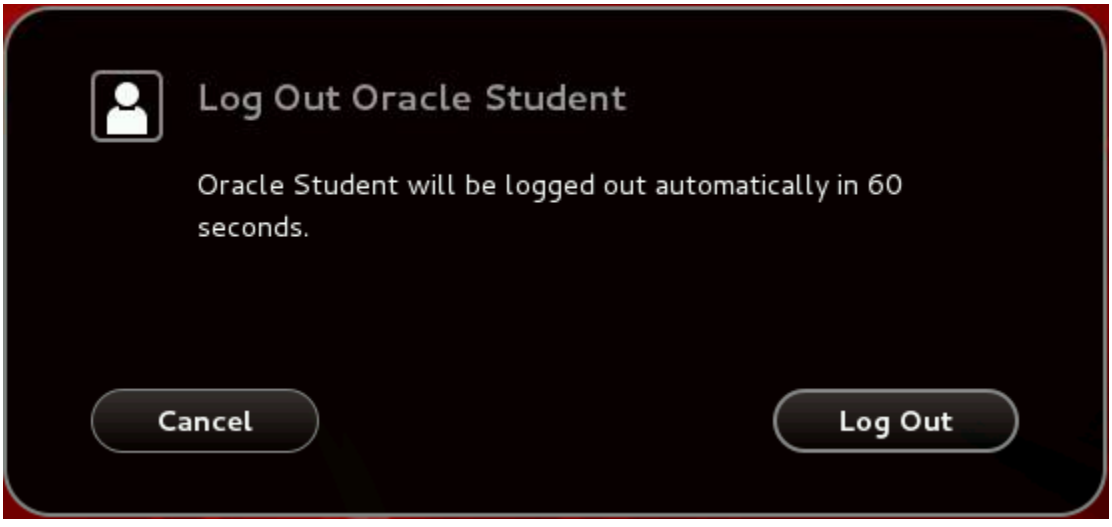


   b. Close the Firefox web browser by clicking the “X” in the top-right corner of the window.

8. Log off from **host03**.
   a. Click the “Oracle Student” in the top-right corner of the GNOME desktop to display the drop-down menu.



   b. Select “Log Out” from the menu.
      - The following window appears.



   c. Click “Log Out.”
   d. Close the VNC window by clicking the “X” in the top-right corner of the window.

Practices for Lesson 6: Installing Oracle Linux 7 by Using Kickstart

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In these practices, you:

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

Practice 6-1: 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.
      - The package is installed.

```
# rpm -qa | grep http
httpd-2.2.3-53.0.1.el5
```

   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 7.1 DVD image is the `OracleLinux-R7-U1-Server-x86_64-dvd.iso` file in the `/OVS/seed_pool` directory.

```
# cd /OVS/seed_pool
# ls
...
OracleLinux-R7-U1-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 OL7.1 DVD image on `/mnt/iso`.

```
# mount -t iso9660 -o loop OracleLinux-R7-U1-Server-x86_64-
dvd.iso /mnt/iso
```

   d. Use the `mkdir` command to create the `/var/www/html/OL71` directory.

```
# mkdir /var/www/html/OL71
```

   e. Use the `cp` command to copy all files and directories from `/mnt/iso` to `/var/www/html/OL71`.
      - This command takes a few minutes to complete.

```
# cp –r /mnt/iso/* /var/www/html/OL71/
```

      - 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 **host01** to `/var/www/html/ks.cfg` on **dom0**.
      - The password is `oracle`.

```
# cd /var/www/html
# scp host01:~/anaconda-ks.cfg ks.cfg
root@host01'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 preconfigured `ks.cfg` file exists on **dom0** in the `/OVS/seed_pool/host07` directory.
      - If you do not want 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 need not 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 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 in the following.
      - Note that the "`network`" line is all one line ending in "`--activate`".

```
# vi ks.cfg
#version=RHEL7
# System authorization information
authconfig --enableshadow --passalgo=sha512

url --url http://192.0.2.1/OL71/

ignoredisk --only-use=xvda
# Keyboard layouts
Keyboard --vckeymap=us --xlayouts='us'
# System language
lang en_US.UTF-8

# Network information
network --bootproto static --device eth0 --gateway 192.0.2.1
```

```
--ip 192.0.2.107
--nameserver=10.216.106.3,192.0.2.1,152.68.154.3
--netmask 255.255.255.0 --ipv6=auto
--hostname=host07.example.com --activate

# Root password
rootpw --iscrypted ...

# System timezone
timezone America/Denver --isUtc --nontp

user --name=oracle --password=$6$... --iscrypted --gecos="Oracle
Student"

# System bootloader configuration
bootloader --location=mbr --boot-drive=xvda
autopart --type=lvm

# Partition clearing information
clearpart --all --drives=xvda

%packages
@core

%end
```

4. Verify the Kickstart file.
   a. From **dom0**, use the `scp` command to copy `/var/www/html/ks.cfg` from **dom0** to `/root/ks.cfg` on **host01**.
      - The password is `oracle`.

```
# scp /var/www/html/ks.cfg host01:~/ks.cfg
root@host01's password: oracle
ks.cfg                                        100%  ...
```

   b. Use the `ssh` command to log on to **host01**.

```
# ssh host01
root@host01's password: oracle
```

   c. From **host01**, use the `yum` command to install the `pykickstart` package.
      - Answer `y` to "Is this ok".

```
# yum install pykickstart
...
Transaction Summary
============================================================
```

```
Install  1 Package

Total download size: 390 k
Installed size: 1.6 M
Is this ok [y/d/N]: y
...
Complete!
```

d. Use the `ksvalidator` utility to verify the `ks.cfg` file in the `/root` directory on **host01**.

- In this example, the command produces no output indicating there are no errors in the `ks.cfg` file.

```
# ksvalidator /root/ks.cfg
```

e. Create an error in the `/root/ks.cfg` file.

- The following example removes the comment (`#` sign) from the first line.

```
# vi /root/ks.cfg
#version=RHEL7                                    (old entry)
version=RHEL7                                     (new entry)
```

f. Repeat step 4d and rerun the `ksvalidator` utility.

```
# ksvalidator /root/ks.cfg
The following problem occurred on line 1 of the kickstart file:

Unknown command: version=RHEL7
```

g. Fix the error in the `/root/ks.cfg` file.

```
# vi /root/ks.cfg
version=RHEL7                                   (old entry)
#version=RHEL7                                  (new entry)
```

h. Repeat step 4d and rerun the `ksvalidator` utility.

- The command produces no output indicating there are no errors in the `ks.cfg` file.

```
# ksvalidator /root/ks.cfg
```

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

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

5. 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 preconfigured `vm.cfg` file exists on **dom0** in the `/OVS/seed_pool/host07` directory.

- If you do not want to edit the `vm.cfg` file as instructed in step 5e, 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 need not edit the file in step 5e.

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

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

```
# 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-R7-U1-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'
```

6. 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 0.0.0.0:5902)
                (location 3)
```

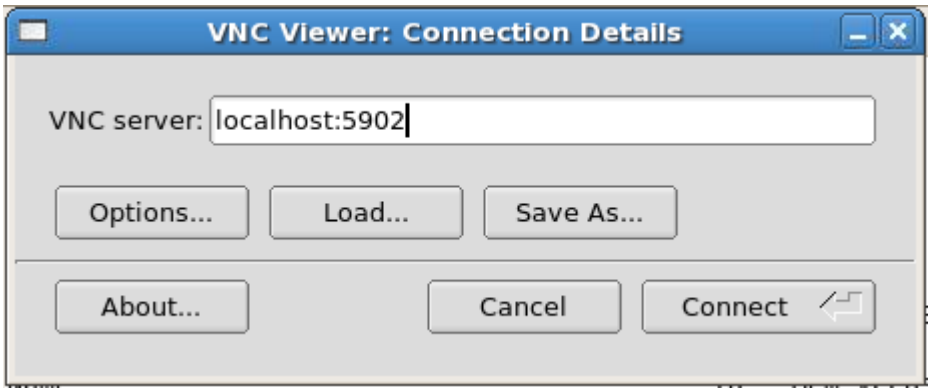
      - The sample shown indicates that the port number is `5902`. Your port number might be different.

   d. Run the `vncviewer&` command.

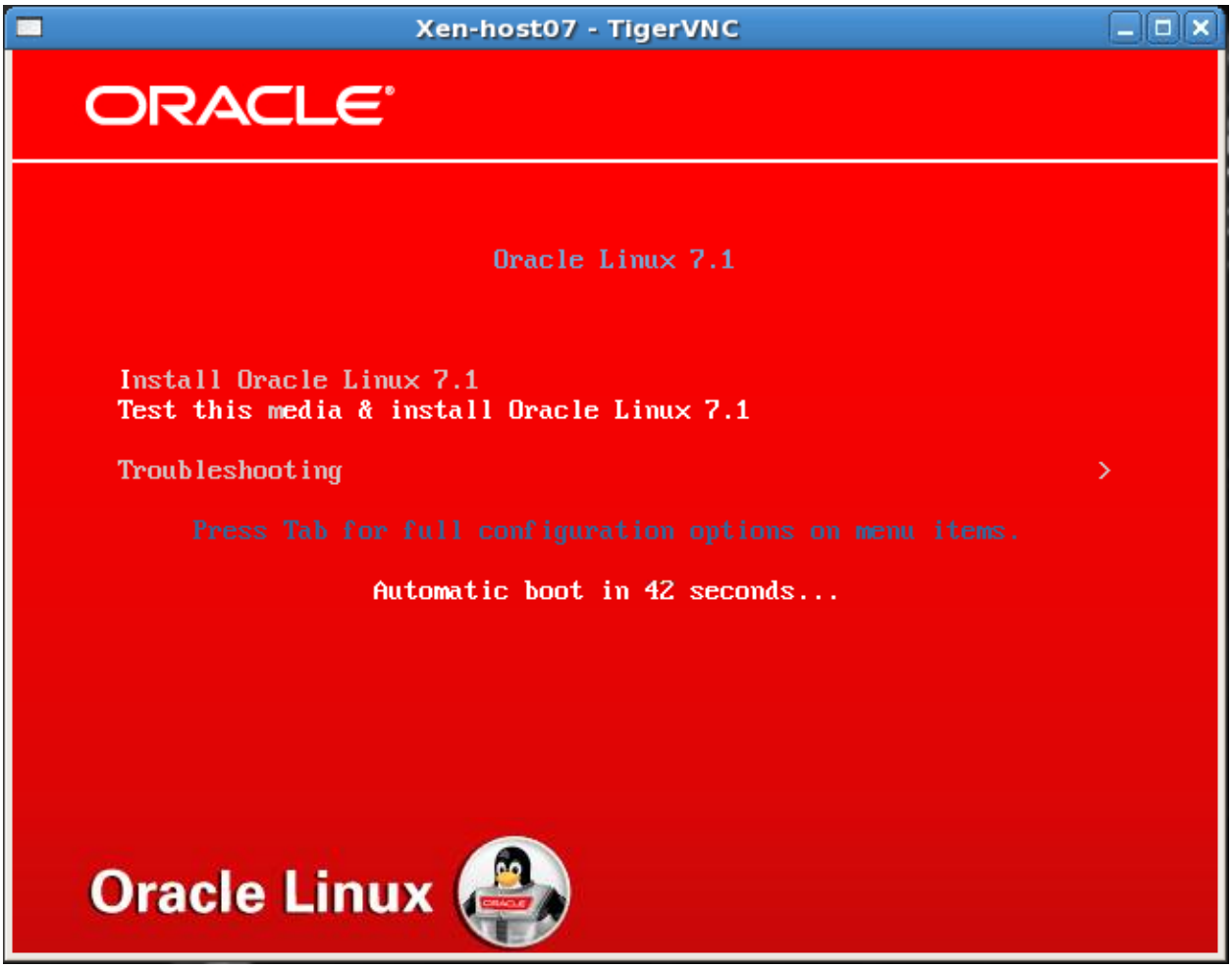
```
# vncviewer&
```

      - 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 `5902`, enter `localhost:5902` and click "Connect."



- The Oracle Linux boot menu appears:

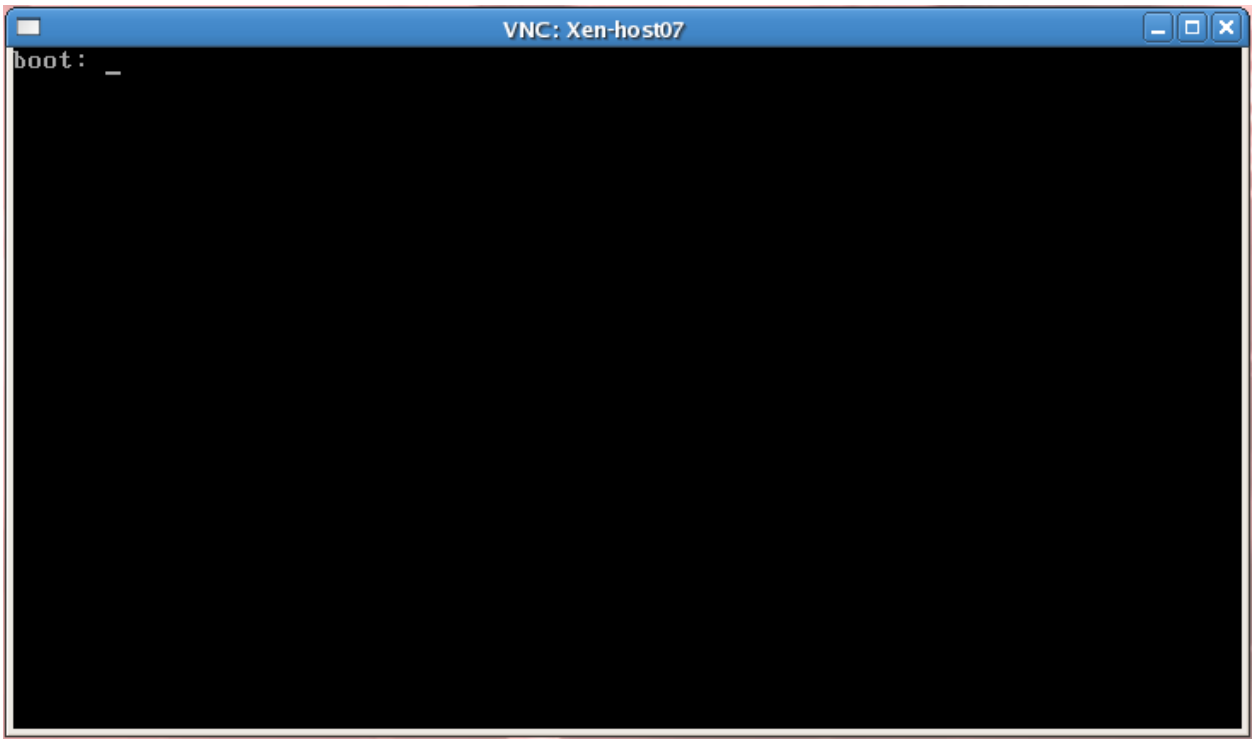


- The Oracle Linux boot menu screen appears for only 60 seconds, after which the "Test this media & install Oracle Linux 7.0" menu option is selected by default.
- If you do not see this screen, meaning the 60-second timeout has expired, click the X in the top-right corner of the current screen to close it, enter the following command from **dom0**, and begin step 6 again starting with 6b.

```
# xm destroy host07
```

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

- The boot prompt is shown:



7. Initiate the Kickstart installation.

- The network interface needs to be configured to retrieve the Kickstart file, `ks.cfg`, from the `192.0.2.1` HTTP server.
- In many cases, the network interface obtains an IP address from DHCP running on the server. However, in this example, DHCP is not running on the installation server.
- Therefore, you need to provide initial network configuration information as part of the boot command.

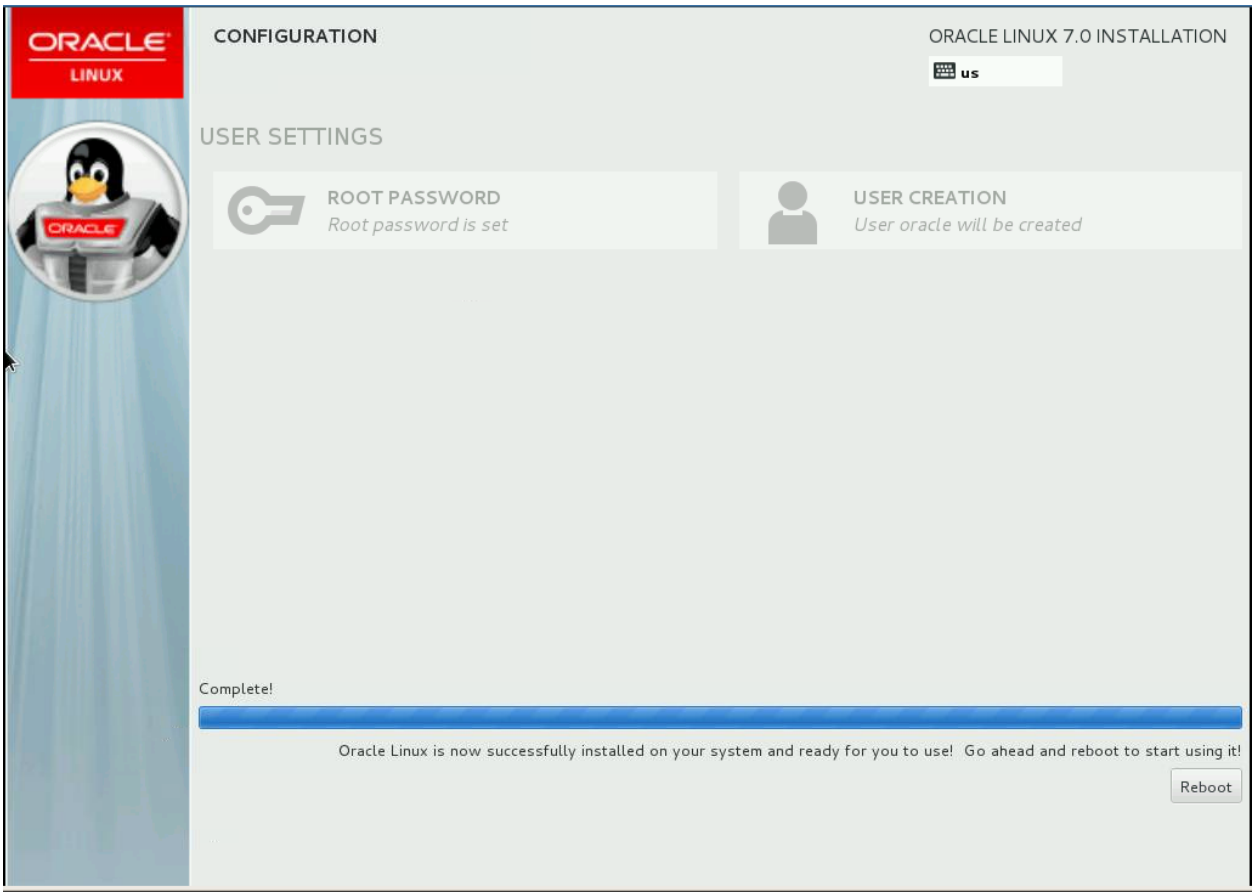
a. Enter the following command from the boot prompt, and press Enter to continue.

- 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`)
- This address information allows an initial network connection required to retrieve the Kickstart file from the installation server.
- The information in the Kickstart file is then used to configure the network interface.

```
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
```

- There is a slight delay before the Kickstart installation begins.

- When the installation is complete, you are prompted to reboot.



b. Click Reboot when prompted.

8. 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 `ip addr` command to display the network configuration.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet addr:127.0.0.1
    ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 0:16:3e:00:01:07 brd ff:ff:ff:ff:ff:ff
    inet addr:192.0.2.107/24 brd 192.0.2.255 scope global eth0
    ...
```

d. Use the `df` command to list the mounted partitions.

```
# df -h
Filesystem                    Size  Used  Avail  Use%  Mounted on
/dev/mapper/ol_host07-root     11G  979M   9.4G   10%  /
...
/dev/xvda1                    497M  142M   355M   29%  /boot
```

e. Use the `cat` command to view the `/etc/resolv.conf` file.

```
# cat /etc/resolv.conf
...
search example.com
nameserver 10.26.106.3
nameserver 192.0.2.1
nameserver 152.68.154.3
```

Practice 6-2: 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 `/boot/grub2/grub.cfg` file to cause boot failure.
   a. Make a backup of `/boot/grub2/grub.cfg`.

```
# cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.BAK
```

   b. Use the `vi` command to edit the `/boot/grub2/grub.cfg` file.
      - Use the `:set nu` command to turn on line numbers.
      - At around line number 103, change `linux16 /vmlinuz-3.8.13-55.1.6.el7uek.x86_64` to `linux16 /vmlinuz-3.13-55.1.6.el7uek.x86_64`.

```
# vi /boot/grub2/grub.cfg
:set nu
linux16 /vmlinuz-3.8.13-55.1.6.el7uek.x86_64    (old entry)
linux16 /vmlinuz-3.13-55.1.6.el7uek.x86_64      (new entry)
```

   c. Use the `systemctl reboot` command to reboot **host07**.

```
# systemctl reboot
Connection to 192.0.2.107 closed by remote host.
Connection to 192.0.2.107 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 0.0.0.0:5902)
                (location 3)
```

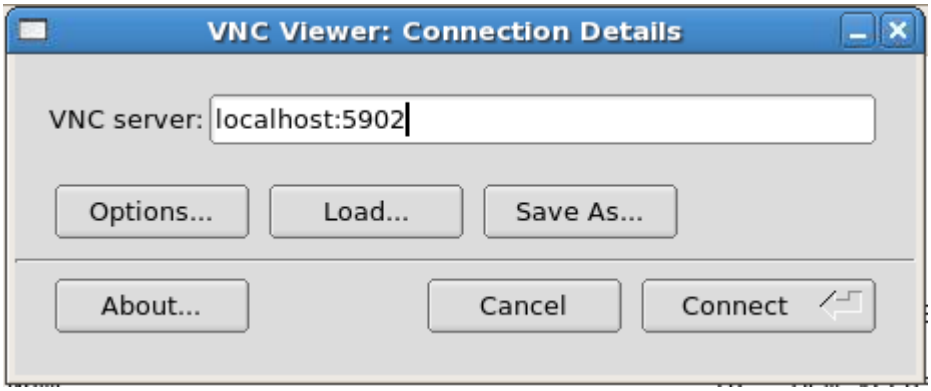
      - In this example, the VNC port number is `5902`. This might not be true in your case.

   b. Run the `vncviewer&` command.

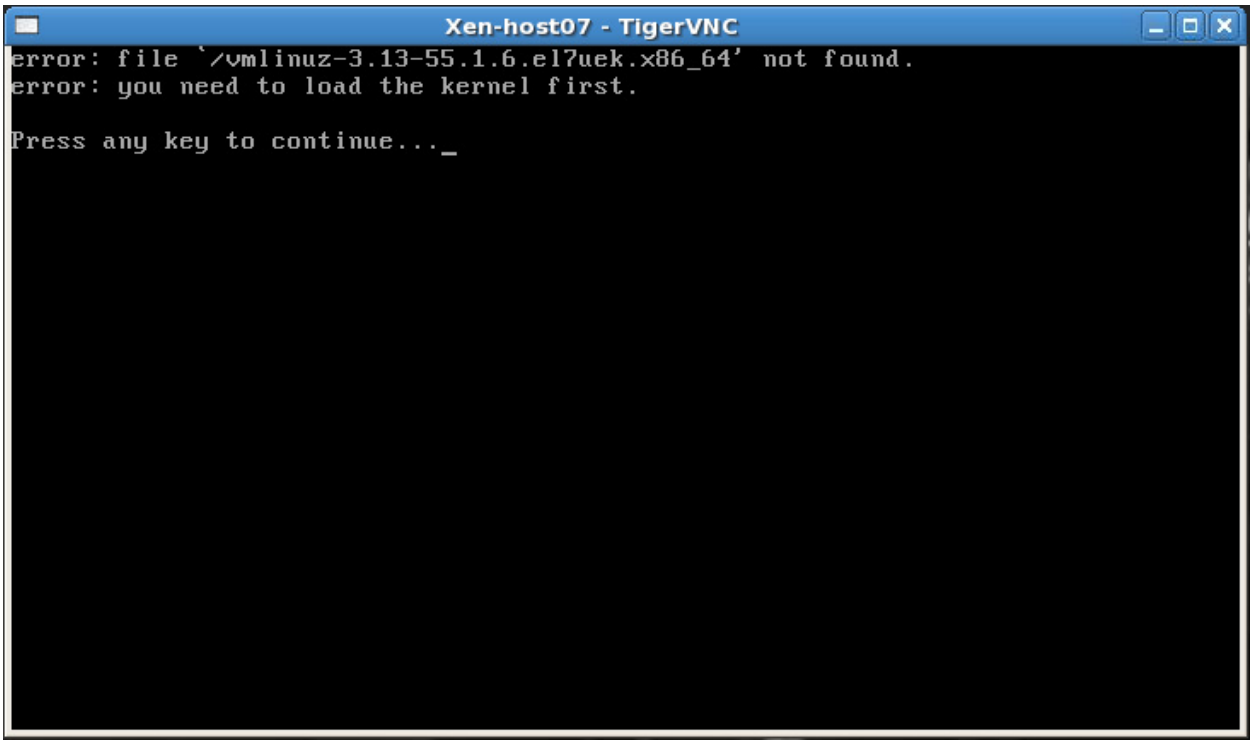
```
# vncviewer&
```

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

c. Enter `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 `5902`, enter `localhost:5902` and click Connect.



- The following screen shows that an error occurred 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 are 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 7 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'`.

   If the `vm.cfg` file is read-only, use `:wq!` to save the file.

```
# 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
```

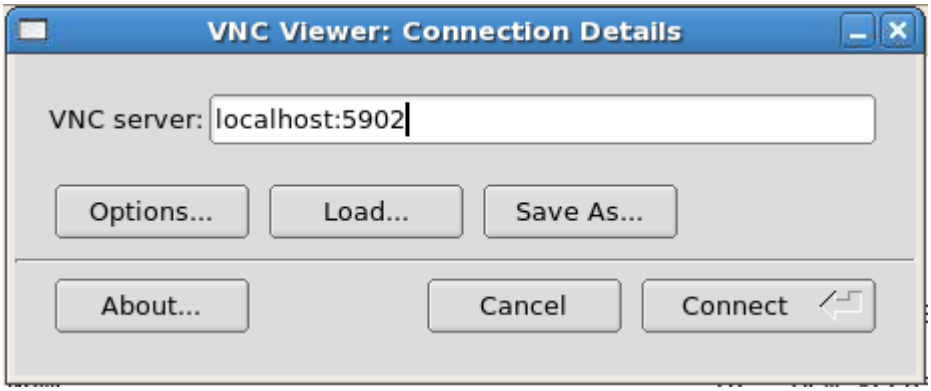
   - Notice that **host07** is now active.

6. Log in to **host07**.
   a. Run the `vncviewer&` command.

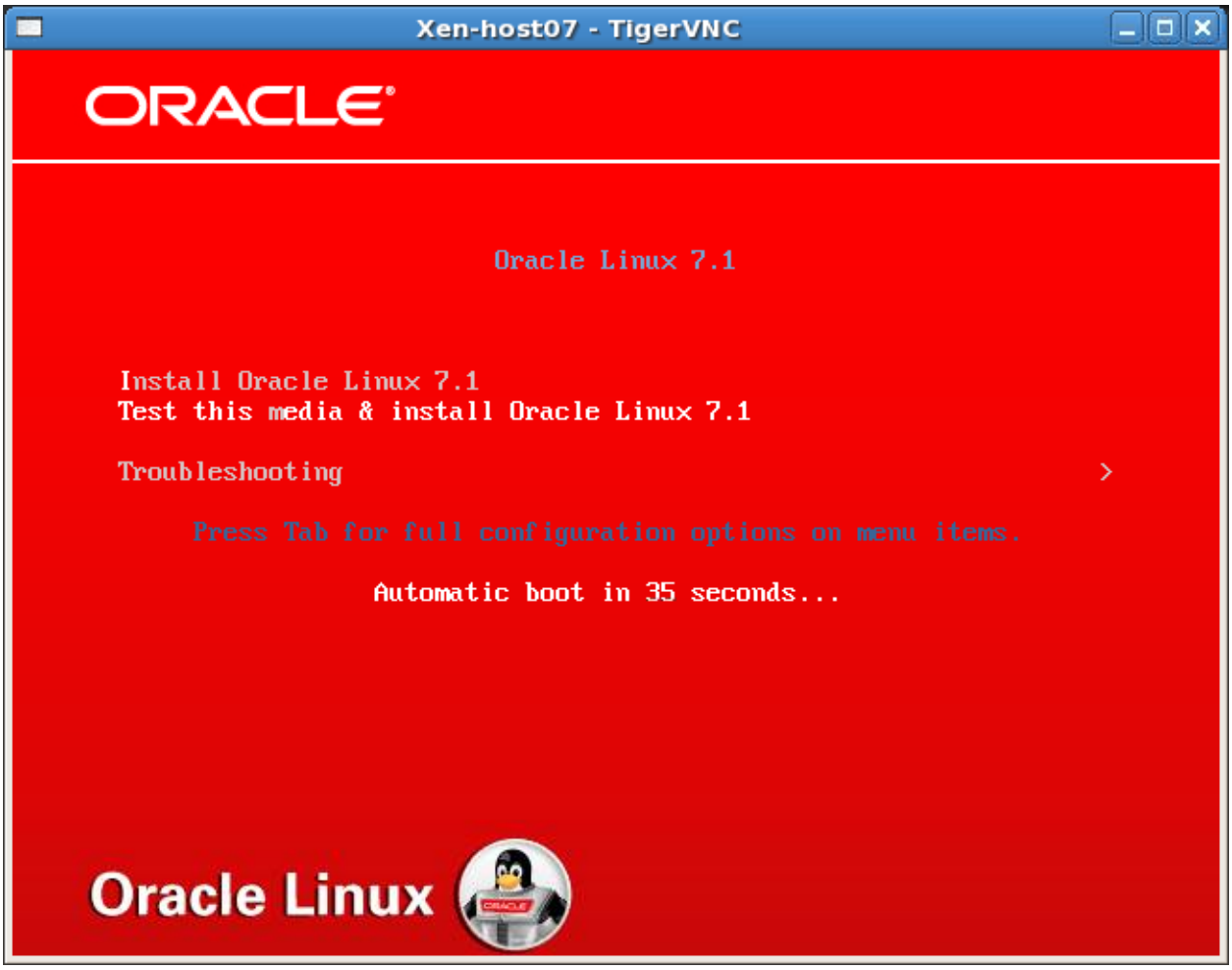
```
# vncviewer&
```

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

b. Enter `localhost:<port_number>`, substituting the port number displayed from the `xm list -l host07 | grep location` command in step 2a. For example, if the port number is `5902`, enter `localhost:5902` and click Connect.



- The Oracle Linux boot menu appears as shown:



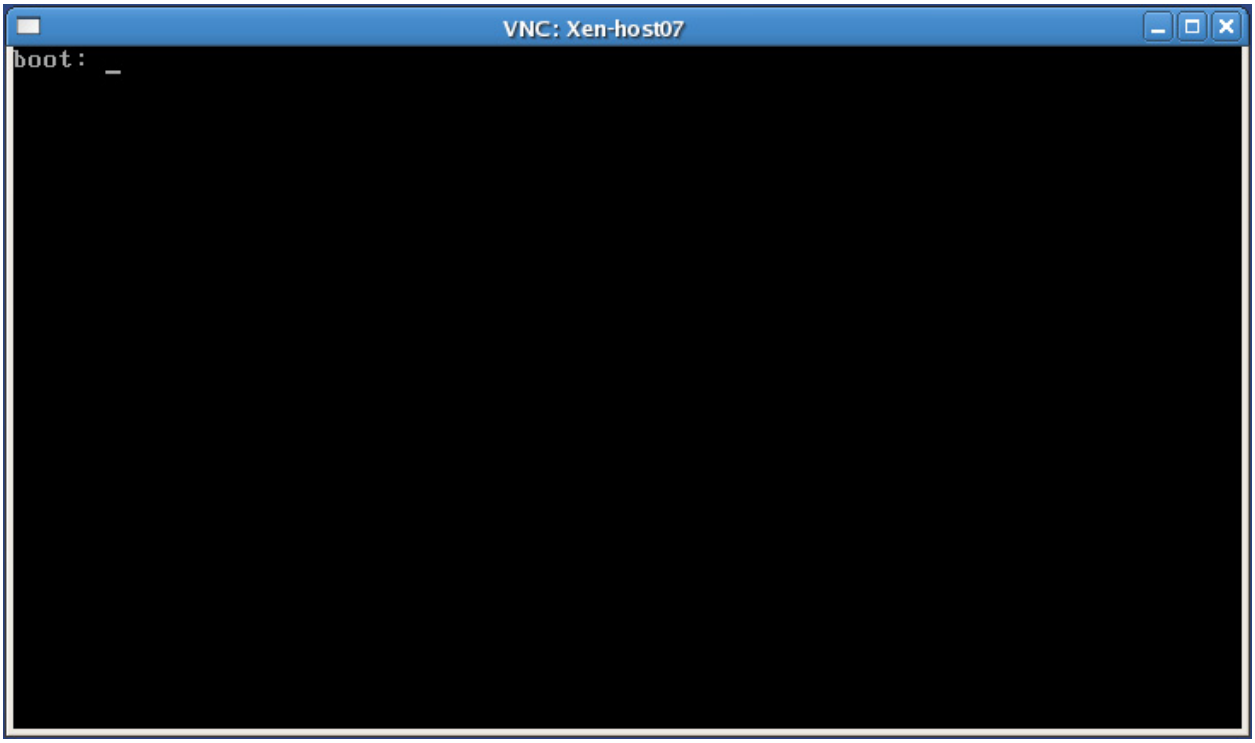
- The Oracle Linux boot menu screen appears for only 60 seconds after which the “Test this media & install Oracle Linux 7.0” menu option is selected by default.

- 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 selecting “Troubleshooting” to display a new menu, and then select “Rescue a Oracle Linux system” from the Troubleshooting menu.



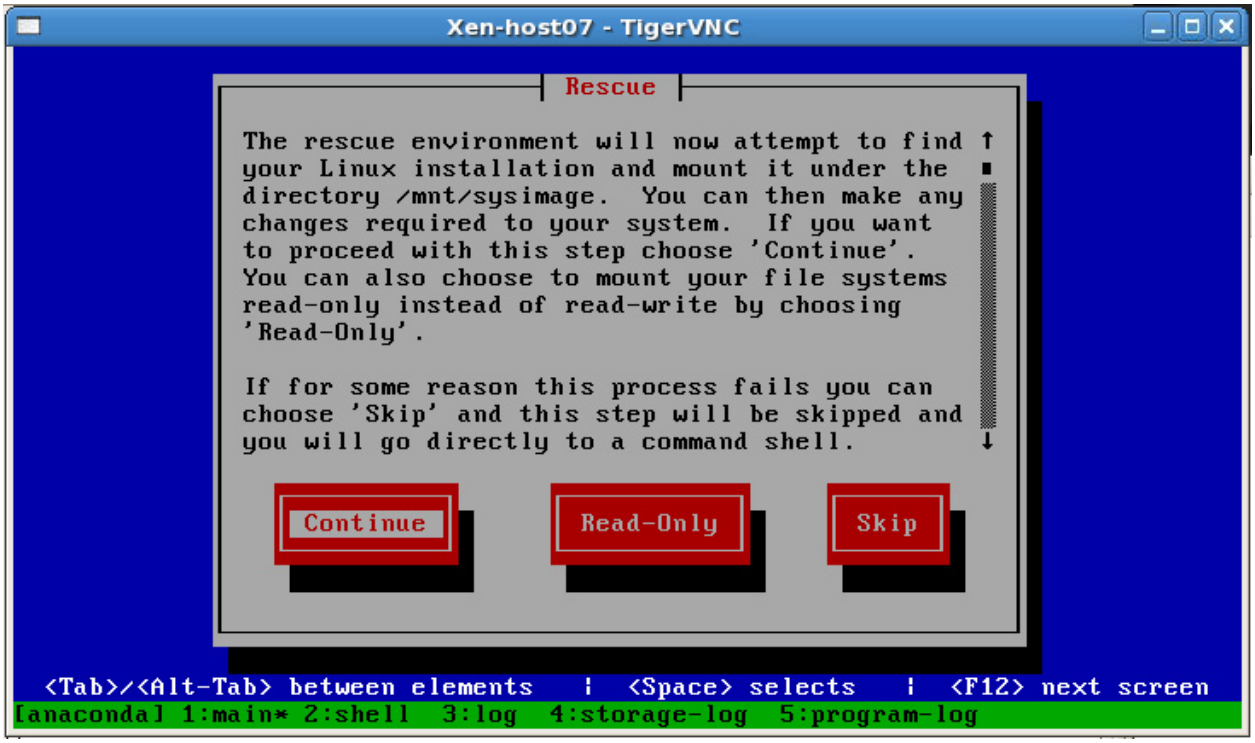
7. Boot into Rescue Mode.

a. Enter `linux rescue` at the `boot:` prompt and press **Enter**.

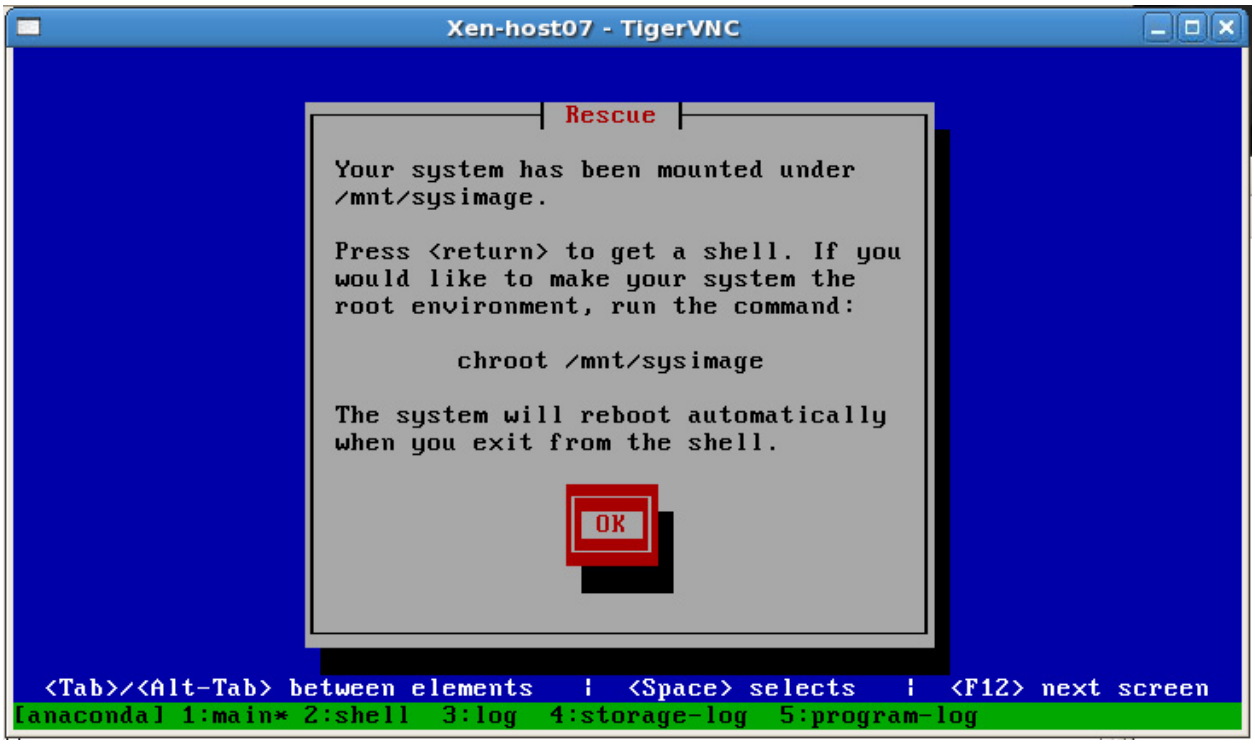
- It takes a few seconds for the rescue process to begin.

```
boot: linux rescue
```

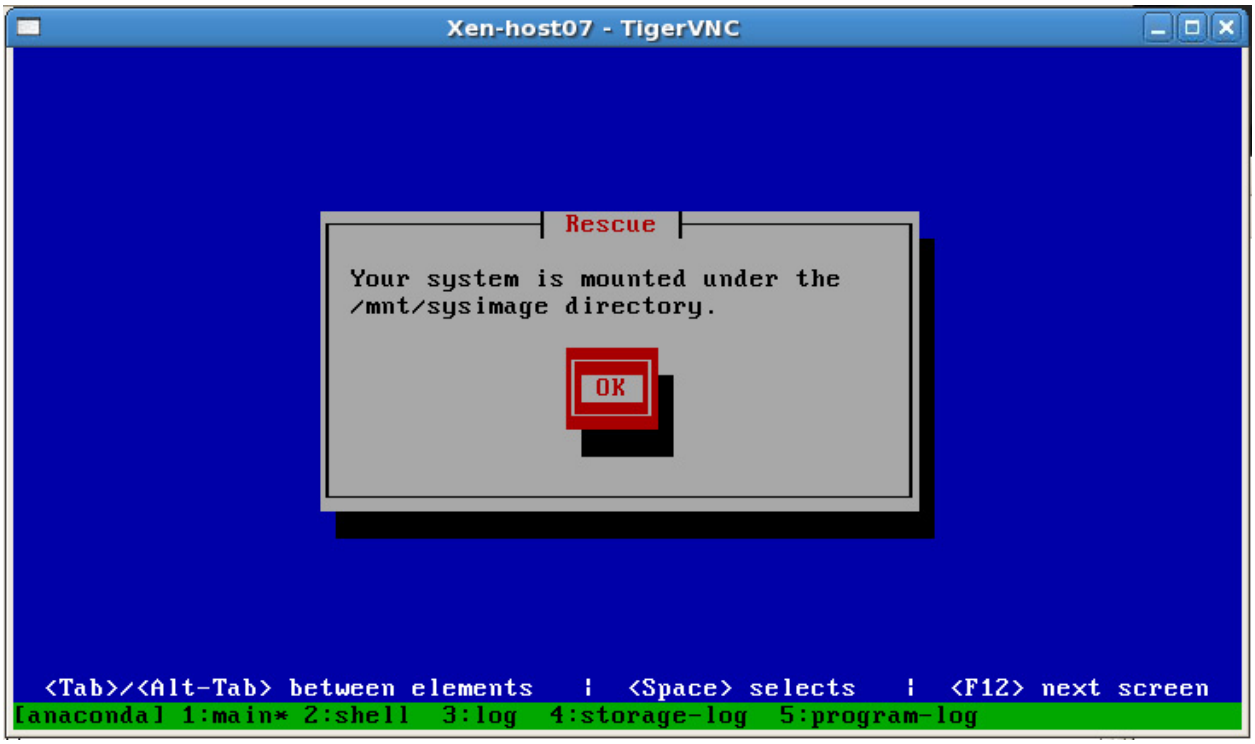
b. Review the information displayed on the following screen. Use the “Tab” key to select “Continue” and press “Enter.”



c. Review the information displayed on the following screen. Press “Enter” to continue.



d. Review the information displayed on the following screen. Press “Enter” to continue.



- A shell prompt is displayed.

8. Repair the corrupted `/boot/grub2/grub.cfg` file.

a. Use the `df` command to view the mounted file systems.

- Notice that the file systems are mounted under the `/mnt/sysimage` directory.

```
# df -h
Filesystem                      ...          Mounted on
...
/dev/mapper/ol_host07-root      ...          /mnt/sysimage
/dev/xvda1                      ...          /mnt/sysimage/boot
...
```

b. Use the `ls` command to view the contents of the current `/` directory.

```
# ls /
bin   dev  firmware  lib64       mnt      proc  run   sys  usr
boot  etc  lib       lost+found  modules  root  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.

- Notice that the file system mount points are different.

```
# df -h
Filesystem                      ...          Mounted on
/dev/mapper/ol_host07-root      ...          /
...
/dev/xvda1                      ...          /boot
...
```

e. Use the `ls` command to view the contents of the current `/` directory.

- Notice that the contents of the `/` directory are different.

```
# ls /
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
```

f. Use the `cp` command to restore `/boot/grub2/grub.cfg` from `/boot/grub2/grub.cfg.BAK`.

```
# cp /boot/grub2/grub.cfg.BAK /boot/grub2/grub.cfg
```

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@192.0.2.107'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 `systemctl poweroff` command to shut down **host07**.

```
# systemctl poweroff
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
```

    a. Use the `/bin/rm -r` command to remove the `/var/www/html/OL71/` directory.

```
# cd /var/www/html
# /bin/rm -r OL71/
```

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=...)
```

    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 7: Samba Services

Chapter 7

Practices for Lesson 7: Overview

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 7-1: Configuring a Samba Server

Overview

In this practice, you do the following:

- Install the packages necessary to configure Samba services on the **host03** VM.
- Start the `smb` service.
- Add the `samba` service to `firewalld`.
- 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 **dom0**.

Tasks

1. Install the `samba` packages on **host03**
   a. As the `root` user on **dom0**, use the `ssh` command to log in to **host03**.
      - The `root` password is `oracle`.

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

   b. From **host03**, use the `rpm -qa` command to list the installed `samba` packages.
      - In this example, two `samba` packages are installed.
      - The `samba` package and the `samba-client` package needs to be installed.

```
# rpm -qa | grep samba
samba-libs-4.1.12-21.el7.x86_64
samba-common-4.1.12-21.el7.x86_64
```

   c. Use the `yum` command to install the `samba` package and the `samba-client` package.
      - The `samba-client` package includes the `smbpasswd` utility.
      - Answer `y` to "Is this ok."

```
# yum install samba samba-client
...
Transaction Summary
============================================================
Install  2 Packages

Total download size: 1.0 M
Installed size: 3.0 M
Is this ok [y/d/N]: y
```

```
...
Complete!
```

2. Start the `smb` service on **host03**.
   a. Use the `systemctl` command to obtain status of the `smb` service.

```
# systemctl status smb
smb.service - Samba SMB Daemon
  Loaded: loaded (/usr/lib/systemd/system/smb.service; disabled)
  Active: inactive
```

   b. Use the `systemctl` command to enable the `smb` service.

```
# systemctl enable smb
ln -s '/usr/lib/systemd/system/smb.service'
/etc/systemd/system/multi-user.target.wants/smb.service'
```

   c. Use `systemctl` to start the `smb` service.
      - Use the `systemctl` command to obtain status of the `smb` service.

```
# systemctl start smb
# systemctl status smb
smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled)
   Active: active (running ) since ...
 Main PID: ... (smbd)
   Status: "smbd: ready to serve connections..."
   CGroup: /system.slice/smb.service
...
```

3. Modify `firewalld` to allow access to the samba service.
   a. Use the `firewall-cmd` command to determine the active `firewalld` zone.
      - In this example, the active zone is "`public`".

```
# firewall-cmd --get-active-zone
public
  interfaces: eth0 eth1 eth2
```

   b. Use the `firewall-cmd` command to list the services that are trusted for the active zone.
      - In this example, the `dhcpv6-client` , `ldap`, and `ssh` services are trusted.

```
# firewall-cmd --list-services
dhcpv6-client  ldap  ssh
```

   c. Use the `firewall-cmd` command to trust the samba service for the "`public`" zone.
      - Update both the runtime configuration and the permanent configuration.

```
# firewall-cmd --zone=public --add-service=samba
success
# firewall-cmd --permanent --zone=public --add-service=samba
success
```

4. 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: The password is shorter than 8 characters
Retype new password: oracle
passwd: all authentication tokens updated successfully.
```

5. 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
```

   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 89, change `workgroup = MYGROUP` to `workgroup = GROUPA`.
      - The `workgroup` parameter defines the workgroup name for your environment. In the classroom environment, this parameter has no effect.

```
    workgroup = GROUPA
```

   d. At around line number 92, change `netbios name = MYSERVER` to `netbios name = SMB-HOST03`.
      - Remove the semicolon at the beginning of the line.
      - 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.

```
    netbios name = SMB-HOST03
```

   e. At around line number 123, 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 282, 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 288, 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`.

6. Use the `testparm` command to check the syntax of the `sbm.conf` file.

- 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`.
- 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
```

7. Reload the `smb.conf` file.
   a. Run the `systemctl` command to reload the `smb` service.
      - This command reloads the `smb.conf` file without stopping the `smb` service.

```
# systemctl reload smb
```

   b. Run the `systemctl` command to view the status of the `smb` service.

```
# systemctl status smb
smb.service - Samba SMB Daemon
  Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled)
  Active: active (running ) since ...
 Main PID: ...
...
<date_time> host03.example.com systemd[1]: Reloaded Samba ...
...
```

8. 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`.
   - You use this password when accessing a Samba share from another Linux system or a Windows system as `user01`.

```
# smbpasswd -a user01
New SMB password:MyOracle1
Retype new SMB password:MyOracle1
Added user user01.
```

Practice 7-2: 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. Install the `samba-client` package on **host01**.
   a. If necessary, open a new terminal window on **dom0**.
   b. Use the `su -` command to become the `root` user on **dom0**.
      - The `root` password is `oracle`.

```
[dom0]$ su -
Password: oracle
```

   c. As the `root` user on **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: ...
```

   d. From **host01**, use the `yum` command to install the `samba-client` package.
      - Answer `y` to "Is this ok."

```
[host01]# yum install samba-client
...
Is this ok [y/N]: y
...
Complete!
```

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`.

```
[host01]# smbclient //host03/tmp -U user01
Enter user01's password: MyOracle1
Domain=[GROUPA] OS=[Unix] Server=[Samba 4.1.12]
smb: \>
```

b. If the `smbclient` command returns "`session setup failed: NT_STATUS_LOGON_FAILURE`," use the `systemctl` command on **host03** to restart the `smb` service.

- After restarting the `smb` service, run the `smbclient` command in step 2a again.

```
[host03]# systemctl restart smb
```

c. At the `smb:` prompt on **host01**, use the `ls` command to list the files in the `/tmp` directory on **host03**.

```
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`.

```
[host01]# smbclient //host03/user01 -U user01
Enter user01's password: MyOracle1
Domain=[GROUPA] OS=[Unix] Server=[Samba 4.1.12]
smb: \>
```

f. Use the `ls` command to list the files in the home directory for user `user01`.

- The command fails because SELinux is in "Enforcing" mode.
- SELinux is covered in a subsequent lesson in this course.

```
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
smb: \>
```

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 the 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 4.1.12]
smb: \>
```

j. Use the `ls` command to list the files in the home directory for user `user01`.

- Because of the change in the SELinux mode, you can now list and access the files in `user01`'s home directory.

```
smb: \> ls
  .                   D        ...
  ..                  D        ...
  .mozilla           DH        ...
  .bash_logout        H        ...
  .bash_profile       H        ...
  .bashrc             H        ...
...
smb: \>
```

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.

```
[host01]# mkdir /homedir
```

b. Use the `yum` command to install the `cifs-utils` package.

- Answer `y` to "Is this ok."

```
[host01]# yum install cifs-utils
...
Is this ok [y/N]: y
...
Complete!
```

c. Use the `mount.cifs` command to mount `user01`'s home directory on the newly created mount point.

- Specify read-only in the mount options.
- The Samba password for `user01` is `MyOracle1`.

```
[host01]# mount.cifs -o username=user01,ro  //host03/user01
/homedir
Password for user01@//host03/user01: MyOracle1
```

d. Use the `df -hT` command to verify that the mount operation was successful.
   - Notice that the file system type for `//host03/user01` is `cifs`.

```
[host01]# df -hT
Filesystem          Type  Size  Used  Avail  Use%  Mounted on
...
//host03/user01     cifs   11G  3.2G   7.1G   32% /homedir
```

e. Verify that the `/homedir` directory is read-only by using the `mount` command.

```
[host01]# mount | grep homedir
//host03/user01 on /homedir type cifs (ro,relatime,vers=1.0...)
```

f. List the contents of `/homedir`.

```
[host01]# ls /homedir
```

   - Notice that the directory is empty.

g. On **host03**, use the `touch` command to create the `/home/user01/testfile` file.

```
[host03]# touch /home/user01/testfile
```

h. On **host01**, list the contents of `/homedir`.
   - Notice that the `testfile` can now be seen from **host01**.

```
[host01]# ls /homedir
testfile
```

i. On **host01**, use the `umount` command to unmount the Samba share.
   - Using the `cd` command ensures you are not in the `/homedir` directory.

```
[host01]# cd
[host01]# umount /homedir
```

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

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

k. Set SELinux to “Enforcing” mode on **host03**.

```
[host03]# getenforce
Permissive
[host03]# setenforce 1
[host03]# getenforce
Enforcing
```

l. Use the `systemctl poweroff` command to shut down **host03**.
   - You are instructed to shut down **host03** in preparation for Practice 8.

```
[host03]# systemctl poweroff
...
```

Practice 7-3: 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.

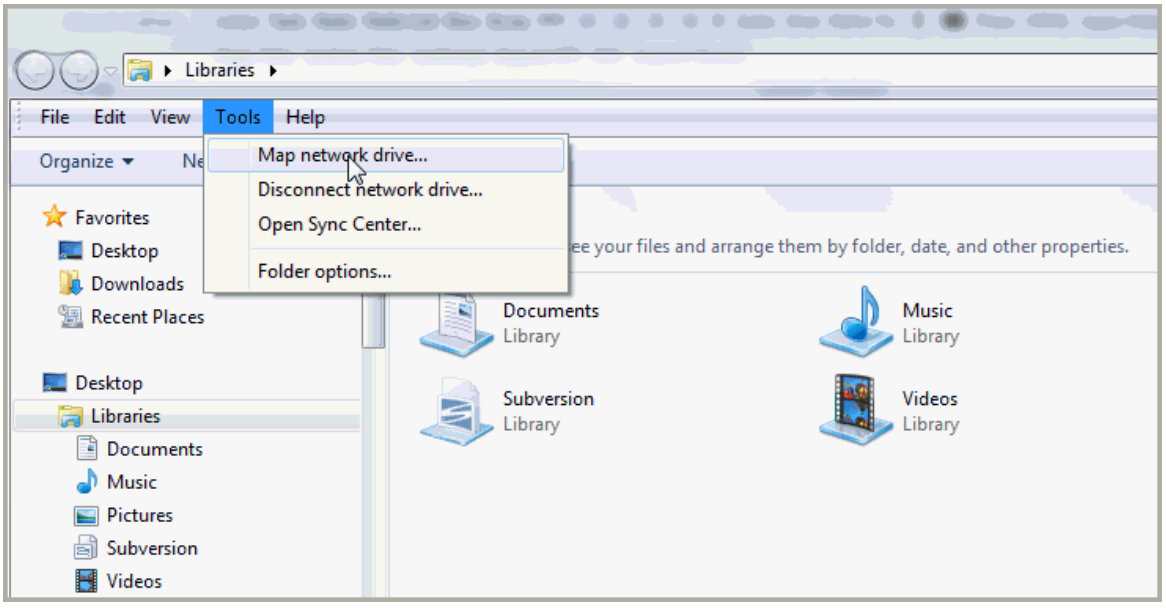
Assumptions

- This practice is not intended to be a hands-on exercise.
- The Linux Samba server is **host01**, IP address is `192.0.2.101`.

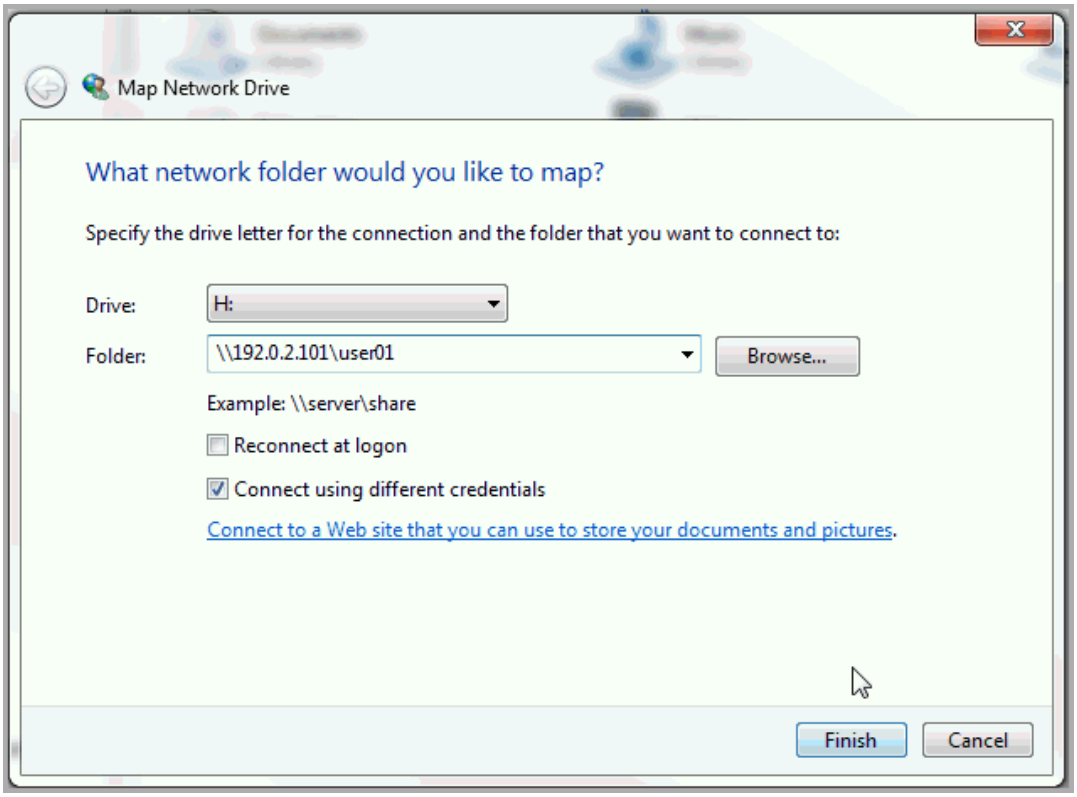
Tasks

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.

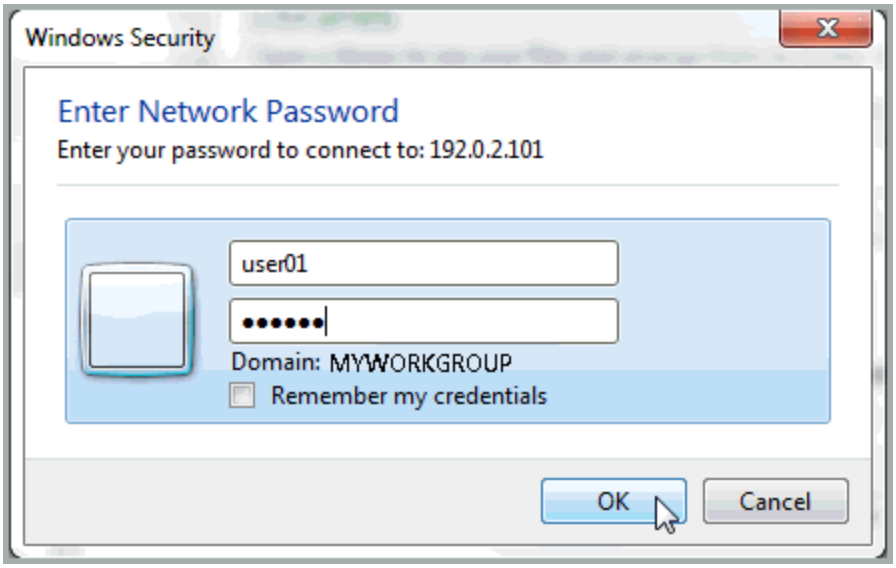


b. Provide the name of the share as \\<server name>\<share name>.



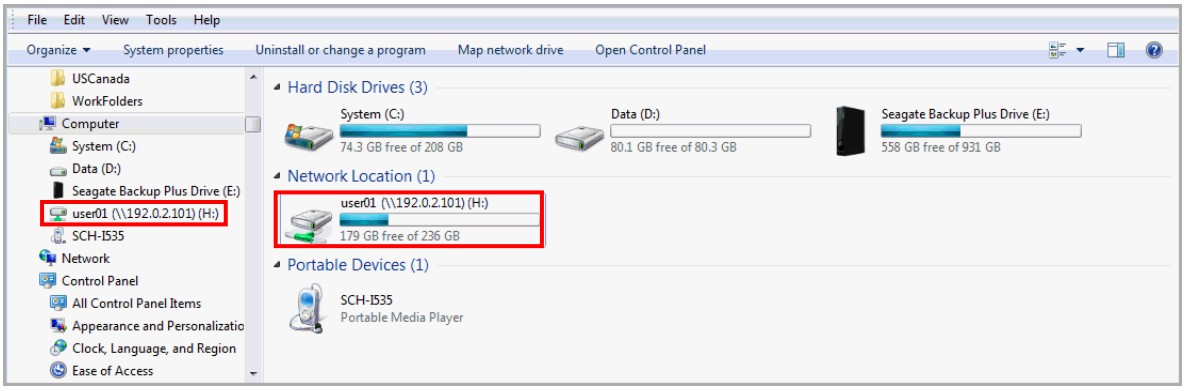
- Select “Connect using different credentials” to provide your Linux username and its associated Samba password.

c. In the Windows Security window, enter the credentials for the share as `user01` and the Samba password as `MyOracle1`.

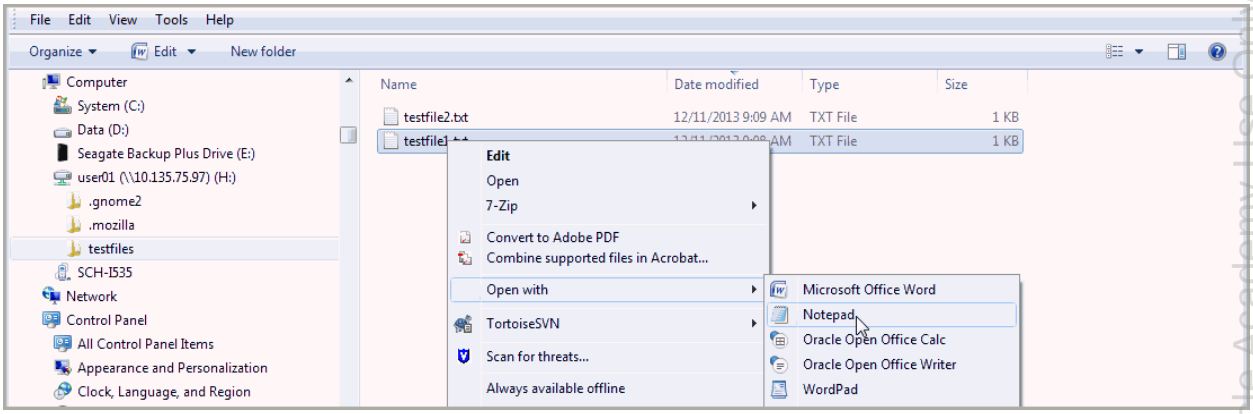


- Click OK to access and map the drive.

- 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 8: Advanced Software Package Management

Chapter 8

Practices for Lesson 8: Overview

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 8-1: Exploring the host04 VM

Overview

In this practice, you do the following:

- Start the **host04** VM.
- Log in to **host04**.
- View Public Yum Server configuration on **host04**.

Assumptions

- You are the `root` user on **dom0**.
- The **host04** VM is preconfigured 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.
      - You were instructed to shut down **host03** at the end of Practice 7.
      - In this example, only **host01** and **host02** VMs are 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
```

   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.
      - 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, press Ctrl + C to kill 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 0.0.0.0:5904)
                (location 3)
```

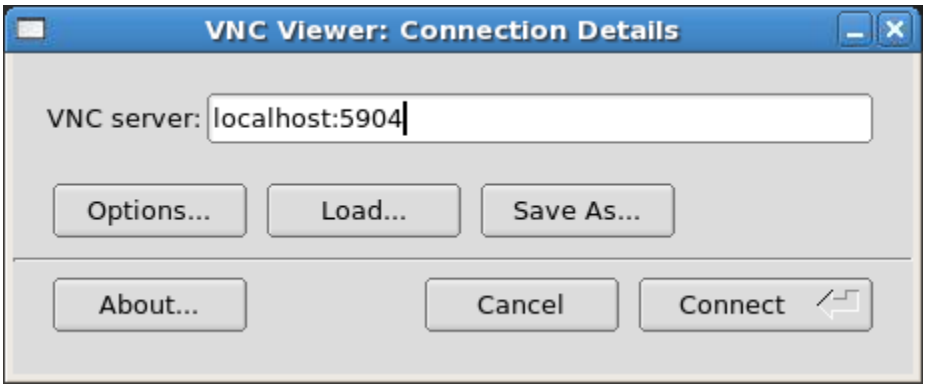
   - The sample shown indicates that the port number is `5904`. Your port number might be different.

   b. Run the `vncviewer&` command.

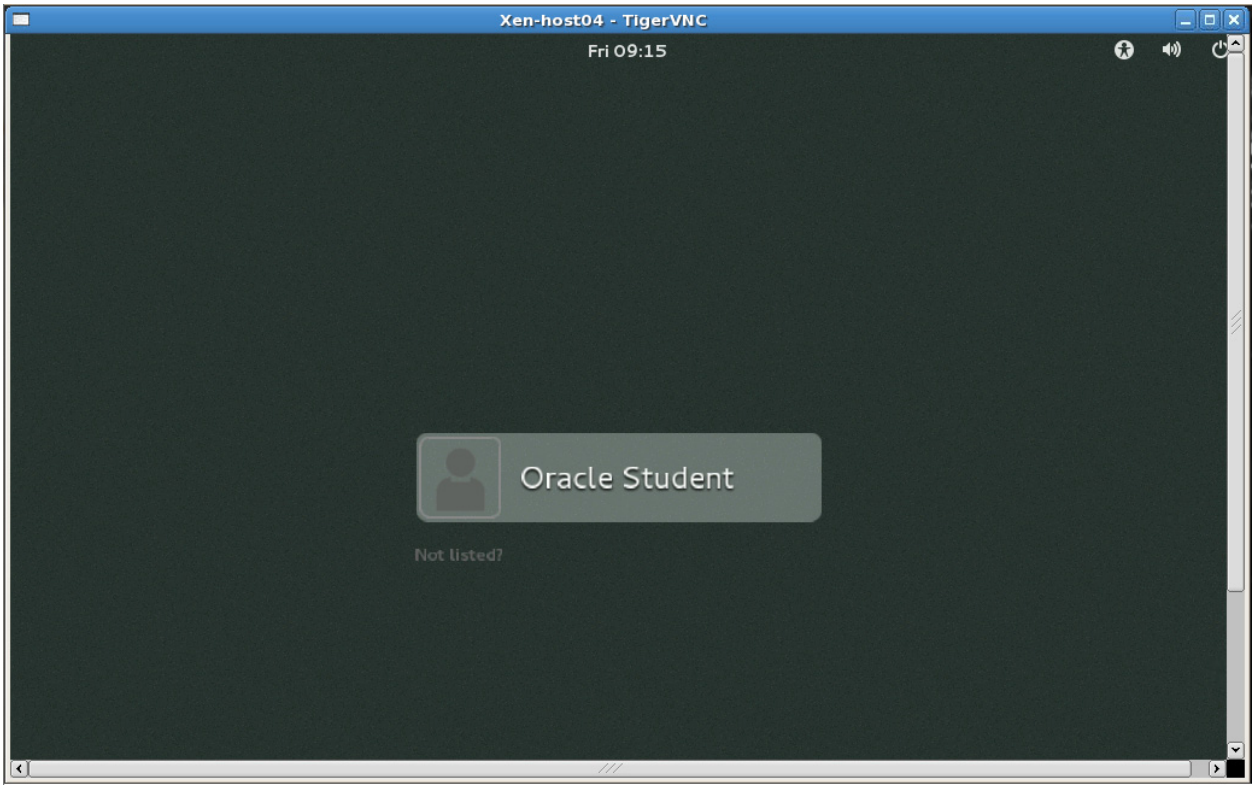
```
# vncviewer&
```

   - 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 `5904`, enter `localhost:5904` and click "Connect."



- The GNOME login screen appears.

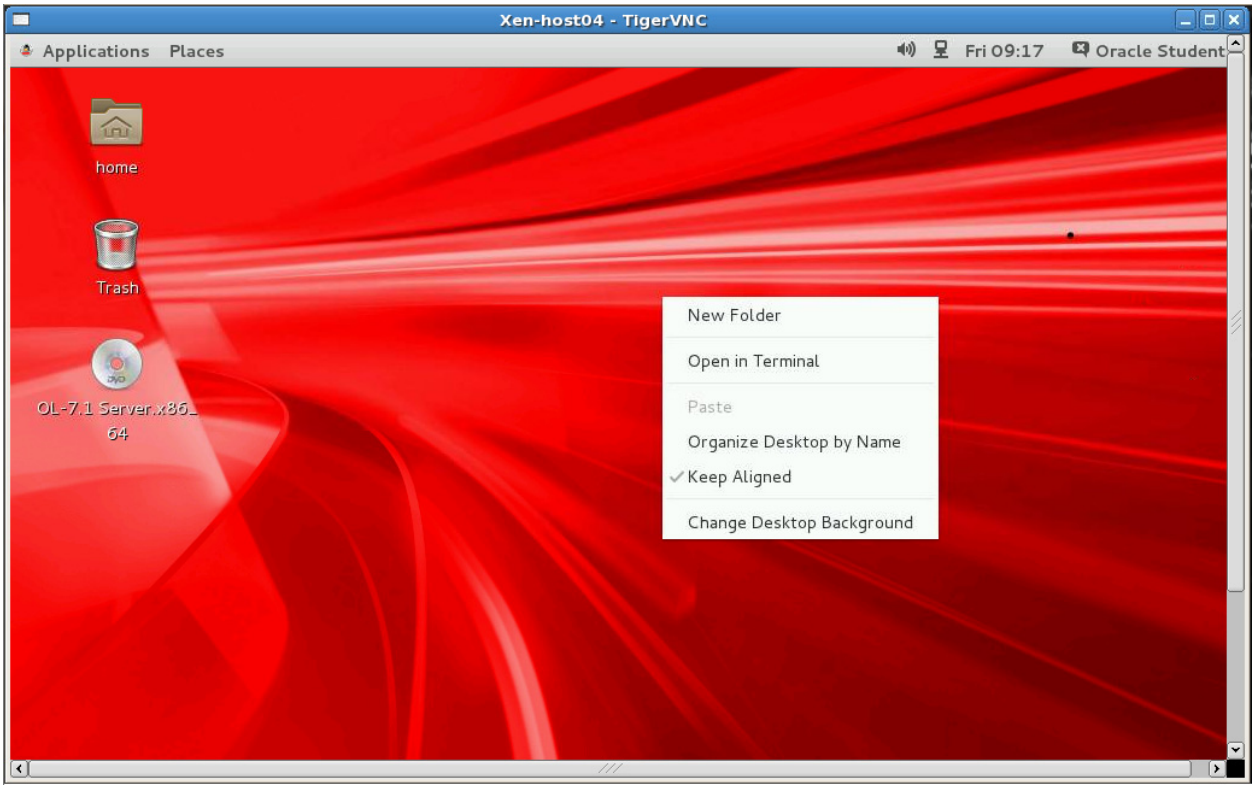


d. Click “Oracle Student” in the list of users. You are prompted for the Password.

e. Enter `oracle` for the Password and click “Sign In.”

- The GNOME desktop appears.

f. Right-click 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
#
```

When starting the **host04** VM, there might be a pop up notice to update the system. Close the pop up window and do not install updates.

3. View the Public Yum Server configuration on **host04**.
   - The **host04** VM is preconfigured 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-ol7.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
...
DNS1=192.0.2.1
DNS2=152.68.154.3
DNS3=10.216.106.3
DNS4=193.32.3.252
DOMAIN=" us.oracle.com example.com"
...
```

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.

```
# cat /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-ol7.repo` file in the `/etc/yum.repos.d` directory.

- The following two Public Yum repositories are enabled:
  - `ol7_latest (enabled=1)`
  - `ol7_UEKR3 (enabled=1)`

```
# cat /etc/yum.repos.d/public-yum-ol7.repo
[ol7_latest]
...
enabled=1
...
[ol7_UEKR3]
...
enabled=1
...
```

Practice 8-2: Managing Yum Plug-Ins

Overview

In this practice, you do the following:

- View currently installed Yum plug-ins.
- Exercise the `langpacks` plug-in.
- Install and exercise the `aliases` plug-in.

Assumptions

You are the `root` user on **host04**.

Tasks

1. On **host04**, display currently installed Yum plug-ins.
   - Sample output is provided throughout this practice. Your output might be different.
   a. Use the `yum clean all` command to clean all cached information.

```
# yum clean all
Cleaning repos: ol7_UEKR3 ol7_latest
Cleaning up everything
```

   - If the following message appears, open another terminal window and kill the `PackageKit` process id (PID).
   - In this example, the PID is `2048`.

```
Another app is currently holding the yum lock; waiting for it to
exit...
  The other application is: PackageKit
   Memory : ...
   Started: ...
   State  : Sleeping, pid: 2048
...
```

   - In a new terminal window, use the `su -` command to become the `root` user (password is `oracle`), then use the `kill <PID>` command to kill the `PackageKit` process.
   - In this example, the PID is `2048`.

```
$ su -
Password: oracle
# kill 2048
```

   b. Run the `yum repolist` command.
   - 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.
   - 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 `langpacks` Yum plug-in is currently enabled.

```
# yum repolist
Loaded plugins: langpacks
ol7_UEKR3 ...
ol7_latest ...
...
repo id           repo name                              status
ol7_UEKR3/x86_64  Latest Unbreakable Enterprise Kernel ...
ol7_latest/x86_64 Oracle Linux 7Server Latest (x86_64) ...
...
```

c. Use the `cd` command to change to the `/etc/yum/pluginconf.d` directory.

- Use the `ls` command to list the contents of the directory.
- This directory contains a configuration file for each installed Yum plug-in.
- Note that there are two configuration files but only one plug-in listed in the output of step 1a. The `rhnplugin.conf` file is the configuration file for the `yum-rhn-plugin`. The `yum-rhn-plugin` is used to connect to the Red Hat Network (RHN) and this plug-in is not enabled when running Oracle Linux.

```
# cd /etc/yum/pluginconf.d
# ls -l
total 12
-rw-r--r--. ... langpacks.conf
-rw-r--r--. ... rhnplugin.conf
```

d. Use the `cat` command to view the contents of the `langpacks.conf` file.

- This plug-in is enabled.

```
# cat langpacks.conf
[main]
enabled=1
...
```

e. Use the `cat` command to view the contents of the `rhnplugin.conf` file.

- This plug-in is not enabled.
- The contents of a Yum plug-in configuration file vary from one plug-in to another.

```
# cat rhnplugin.conf
[main]
enabled=0
...
```

2. Exercise the `langpacks` plug-in.
   a. Use the `rpm -qa` command to find the package name of the `langpacks` plug-in.
      - The package name is `yum-langpacks`.

```
# rpm -qa | grep langpacks
yum-langpacks-0.4.2-3.el7.noarch
```

   b. Use the `rpm -ql` command to view the files that are included with the `yum-langpacks` package.
      - The man page installed with the `yum-langpacks` package is `yum-langpacks(8)`.

```
# rpm -ql yum-langpacks
/etc/yum/pluginconf.d/langpacks.conf
/usr/lib/python2.7/site-packages/yum_langpacks-0.4.2-py2.7...
...
/usr/share/man/man8/yum-langpacks.8.gz
```

   c. View the `yum-langpacks(8)` man page.
      - After viewing the man page, press `q` to quit.

```
# man yum-langpacks
...
DESCRIPTION
       yum-langpacks is a plugin for yum to install language
       packs. This plug-in allows various user commands.

       command is one of:
        * langavailable [language1] [language2] [...]
        * langinfo [language1] [language2] [...]
        * langinstall [language1] [language2] [...]
        * langlist [language1] [language2] [...]
        * langremove [language1] [language2] [...]

        langavailable
               This command allows user to find if language
               support is available for the given input
               languages.

        langinfo
               This command allows user to check which packages
               get installed when the given input language
               support is installed.

        langinstall
               This command allows user to install language
```

```
                packs for the given input languages.

        langlist
                This command prints list of the installed
                languages.

        langremove
                This command allows user to remove the installed
                language packs for a given input languages.
...
```

d. Use the `langpack` plug-in to list the available languages.
   - Pipe the output to `less` to view one page at a time.

```
# yum langavailable
Loaded plugins: langpacks
Displaying all available language:-
Afrikanns [af]
Akan [ak]
Albanian [sq]
Amharic [am]
...
Yiddish [yi]
Zulu [zu]
```

e. Use the `langpack` plug-in to list the packages that get installed with `Yiddish` language support.
   - You can use either `Yiddish` or the language ID, `yi`, as an argument to this command.

```
# yum langinfo Yiddish
Loaded plugins: langpacks
Language-Id=Yiddish
  hunspell-yi
```

f. Use the `langpack` plug-in to install `Yiddish` language support.
   - Answer `y` to "Is this ok."
   - You are asked about the GPG key only the first time you use `yum` to install or update a package.

```
# yum langinstall Yiddish
Loaded plugins: langpacks
...
Is this ok [y/d/N]: y
...
Importing GPG key ...
...
```

```
Is this ok [y/N]: y
...
Language packs installed for: yi
```

g. Use the `langpack` plug-in to list the installed languages.

- Note that the `Yiddish` language support is now installed.

```
# yum langlist
Loaded plugins: langpacks
Installed languages:
        Yiddish
```

h. Use the `langpack` plug-in to remove `Yiddish` language support.

- Answer `y` to "Is this ok."

```
# yum langremove Yiddish
Loaded plugins: langpacks
...
Is this ok [y/N]: y
...
Language packs removed for: Yiddish
```

3. Install the `aliases` Yum plug-in.

a. Use the `yum` command to list available Yum plug-ins that you can install.

- In this example, there are six Yum plug-ins available to install.

```
# yum list available | grep yum-plugin
kabi-yum-plugins.noarch ...
yum-plugin-aliases.noarch ...
yum-plugin-changelog.noarch ...
yum-plugin-tmprepo.noarch ...
yum-plugin-verify.noarch ...
yum-plugin-versionlock.noarch ...
```

b. Use the `yum` command to install the `yum-plugin-aliases` plug-in.

- Answer `y` to "Is this ok."

```
# yum install yum-plugin-aliases
...
Is this ok [y/d/N]: y
...
Complete!
```

4. Exercise the `aliases` plug-in.
   a. Use the `rpm -ql` command to view the files that are included with the `yum-plugin-aliases` package.
      - Note that the man page installed with the `yum-plugin-aliases` package is `yum-aliases(1)`.

```
# rpm -ql yum-plugin-aliases
/etc/yum/aliases.conf
/etc/yum/pluginconf.d/aliases.conf
...
/usr/share/man/man1/yum-aliases.1.gz
```

   b. View the `yum-aliases(1)` man page.
      - After viewing the man page, press `q` to quit.

```
# man yum-aliases
...
DESCRIPTION
       This plugin changes other commands in yum, much like the
       alias command in bash. There are a couple of notable
       differences from shell style aliases though. The alias
       command has three forms:
        * alias
        * alias command
        * alias command result

        The first form lists all current aliases with their
        final result, the second form looks up a "command" and
        shows it's final result or an error message. The last
        form creates a new alias.
...
```

   c. Use the `cat` command to view the `/etc/yum/aliases.conf` file.
      - This file defines a number of Yum command aliases.

```
# cat /etc/yum/aliases.conf
...
DEV --enablerepo=development
UPT --enablerepo=updates-testing
...
SEC --security
CRIT --sec-severity=critical
FORCE --skip-broken --disableexcludes=all
DUPS --showduplicates

up upgrade
```

```
inst install
in install
rm remove
down downgrade
rein reinstall
...
ls list
lsi ls installed
lsa ls available
...
```

d. Use the `aliases` plug-in to list Yum command aliases.

```
# yum alias
Loaded plugins: aliases, langpacks
Alias ALL = --enablerepo=development --enablerepo=updates...
Alias ALLDBG = --enablerepo=fedora-debuginfo --enablerepo=...
Alias CRIT = --sec-severity=critical
...
Alias up = upgrade
Alias upi = updateinfo
Alias v = version
alias done
```

e. Use the `aliases` plug-in to list the available packages to install.

- The command to list available packages to install is `yum list available`.
- The `lsa` alias produces the same list of available packages.

```
# yum lsa
...
zsh.x86_64                  5.0.2-7.el7_1.1       ol7_latest
zziplib.i686                0.13.62-5.el7         ol7_latest
zziplib.x86_64              0.13.62-5.el7         ol7_latest
```

Practice 8-3: Using Yum Utilities

Overview

In this practice, you do the following:

- View available errata for your system.
- View CVE information.
- Update the packages affected by the specific CVE.
- View software package information.
- View dependencies for a software package.
- Use the Yum `--downloadonly` option.
- Use the `yumdownloader` and the `repoquery` utilities.

Assumptions

You are the `root` user on **host04**.

Tasks

1. Manage errata for your system.
   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.
      - 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 (critical, important, moderate)
        - Enhancements

```
# yum updateinfo list
Loaded plugins: aliases, langpacks
ELSA-2015-0672 Moderate/Sec.  bind-libs-32:9.9.4-18.el7_1.1...
ELSA-2015-0672 Moderate/Sec.  bind-libs-lite-32:9.9.4-18.el7...
...
ELBA-2015-0741 bugfix         binutils-2.23.52.0.1-30.el7_1.1...
ELBA-2015-0974 bugfix         binutils-2.23.52.0.1-30.el7_1.2...
...
ELEA-2015-0969 enhancement    crash-7.0.9-5.el7_1.x86_64
ELEA-2015-0732 enhancement    dnsmasq-2.66-13.el7_1.x86_64
...
ELSA-2015-0265 Critical/Sec.  firefox-31.5.0-2.0.1.el7_0...
ELSA-2015-0718 Critical/Sec.  firefox-31.5.3-3.0.1.el7_1...
...
updateinfo list done
```

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: aliases, langpacks
 CVE-2015-1349 Moderate/Sec.  bind-libs-32:9.9.4-18.el7_1.1...
 CVE-2015-1349 Moderate/Sec.  bind-libs-lite-32:9.9.4-18.el7...
...
 CVE-2015-0822 Critical/Sec.  firefox-31.5.0-2.0.1.el7_0...
 CVE-2015-0827 Critical/Sec.  firefox-31.5.0-2.0.1.el7_0...
...
 CVE-2014-8962 Important/Sec. flac-libs-1.3.0-5.el7_1.x86_64
 CVE-2014-9028 Important/Sec. flac-libs-1.3.0-5.el7_1.x86_64
...
 CVE-2015-0255 Moderate/Sec.  xorg-x11-server-common-1.15.0-...
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.
   - 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.

```
# yum updateinfo list --cve CVE-2015-0255
Loaded plugins: aliases, langpacks
ELSA-2015-0797 Moderate/Sec.  xorg-x11-server-Xorg-1.15.0...
ELSA-2015-0797 Moderate/Sec.  xorg-x11-server-common-1.15...
updateinfo list done
```

d. Display additional information about a specific CVE.
   - Use the `info` argument instead of the `list` argument.
   - Your output differs if you choose a different CVE.

```
# yum updateinfo info --cve CVE-2015-0255
Loaded plugins: aliases, langpacks
==============================================================
   xorg-x11-server security update
==============================================================
  Update ID : ELSA-2015-0797
    Release : Oracle Linux 7
       Type : security
     Status : final
     Issued : 2015-04-09
       CVEs : CVE-2015-0255
Description : [1.15.0-26]
```

```
            : - CVE fixes for: CVE-2015-0255
   Severity : Moderate
updateinfo info done
```

e. Update the packages affected by the specific CVE.

- Answer `y` when asked "Is this ok."

```
# yum update --cve CVE-2015-0255
Loaded plugins: aliases, langpacks
...
Is this ok [y/d/N]: y
...
Complete!
```

2. View the Oracle Database preinstallation packages (`oracle-rdbms`).

a. Use the `yum` command to list the Oracle Database preinstallation packages (`oracle-rdbms`) that are available for installation.

- You can use the `lsa` alias instead of `list available`.

```
# 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 preinstallation packages.

- 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.

```
# yum info oracle-rdbms*
Loaded plugins: aliases, langpacks
Available Packages
Name        : oracle-rdbms-server-11gR2-preinstall
Arch        : x86_64
Version     : 1.0
Release     : 3.el7
Size        : 18 k
Repo        : ol7_latest/x86_64
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     : 3.el7
```

```
Size         : 17 k
Repo         : ol7_latest/x86_64
Summary      : Sets the system for Oracle single instance and ...
License      : GPLv2
Description : This package installs software packages and ...
```

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.
   - If a dependency package is missing, it is downloaded along with the `oracle-rdbms-server-12cR1-preinstall` package in the next step.

```
# repoquery --requires oracle-rdbms-server-12cR1-preinstall
/bin/bash
/bin/sh
/etc/redhat-release
bind-utils
...
xorg-x11-utils
xorg-x11-xauth
```

d. Use the `--downloadonly` option to download the `oracle-rdbms-server-12cR1-preinstall` package and any missing dependent packages.
   - In this example, six packages are downloaded in addition to the `oracle-rdbms-server-12cR1-preinstall-1.0-3.el7.x86_64.rpm` package.

```
# yum install oracle-rdbms-server-12cR1-preinstall --
downloadonly
Loaded plugins: aliases, langpacks
...
Transaction Summary
============================================================
Install  1 Package (+6 Dependent packages)

Total download size: 9.8 M
Installed size: 29 M
Background downloading packages, then exiting:
(1/7): compat-libcap1-1.10-7.x86_64.rpm ...
...
exiting because "Download Only" specified
```

e. Verify that the package and its dependency packages are downloaded by examining the content of the `/var/cache/yum/x86_64/7Server/ol7_latest/packages` directory.

- 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.

```
# cd /var/cache/yum/x86_64/7Server/ol7_latest/packages
# ls
compat-libcap1-1.10-7.x86_64.rpm
...
oracle-rdbms-server-12cR1-preinstall-1.0-3.el7.x86_64.rpm
```

3. 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.
      - Note that `yumdownloader` and `repoquery` are included in the `yum-utils` package.

```
# rpm -ql yum-utils
...
/usr/bin/repoquery
...
/usr/bin/yumdownloader
...
```

   b. Use the `--downloadonly` option of the `downloadonly` plug-in to attempt to download the `xorg-x11-server-Xorg` program.
      - The package is not downloaded because it is already installed.

```
# yum install xorg-x11-server-Xorg --downloadonly
Loaded plugins: aliases, langpacks
Package xorg-x11-server-Xorg-1.15.0-33.el7_1.x86_64 already
installed and latest version
Nothing to do
```

   c. Use the `yumdownloader` command to download the `xorg-x11-server-Xorg` package.
      - The command downloads the package in the current directory.
      - The command does not download the dependencies for the `xorg-x11-server-Xorg` program.

```
# yumdownloader xorg-x11-server-Xorg
Loaded plugins: aliases, langpacks
xorg-x11-server-Xorg-1.15.0-33.el7_1.x86_64.rpm ...
```

d. Use the `yum deplist` command to display the dependencies for the `xorg-x11-server-Xorg` program.

- 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` and `rpm` commands 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.

```
# yum deplist xorg-x11-server-Xorg
Loaded plugins: aliases, langpacks
Finding dependencies:
package: xorg-x11-server-Xorg.x86_64 1.15.0-33.el7_1
  dependency: config(xorg-x11-server-Xorg) = 1.15.0-33.el7_1
   provider: xorg-x11-server-Xorg.x86_64 1.15.0-33.el7_1
  dependency: libGL.so.1()(64bit)
   provider: mesa-libGL.x86_64 10.2.7-5.20140910.el7
...
```

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 seconds to run.
- Compare this list with the list obtained with the `yum deplist` command in step 3d.

```
# repoquery --whatrequires xorg-x11-server-Xorg
xorg-x11-drv-ati-0:7.2.0-9.20140113git3213df1.el7.x86_64
xorg-x11-drv-ati-0:7.4.0-1.20140918git56c7fb8.el7.x86_64
xorg-x11-drv-dummy-0:0.3.6-15.el7.x86_64
...
```

Practice 8-4: Creating an RPM Package

Overview

In this practice, you prepare to build an RPM package. The steps for this preparation are:

- Create a nonprivileged 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. 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.
      - In this example, the `rpmdevtools` package is not installed.

```
# rpm -qa | grep rpmdevtools
```

   b. If necessary, use the `yum` command to install the `rpmdevtools` package.
      - 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
      - Answer `y` to "Is this ok."

```
# yum install rpmdevtools
...
Transaction Summary
================================================================
Install  1 Package (+6 Dependent packages)

Total download size: 541 k
Installed size: 1.1 M
Is this ok [y/d/N]: y
```

```
...
Complete!
```

2. Create a nonprivileged 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------. 3 rpmbuilder rpmbuilder ... /home/rpmbuilder
```

   c. Use the `passwd` command to create a password of `oracle` for the `rpmbuilder` user.
      - Ignore the BAD PASSWORD warning.

```
# passwd rpmbuilder
Changing password for user rpmbuilder.
New password: oracle
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: oracle
passwd: all authentication tokens updated successfully.
```

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
drwxrwxr-x. ... .cache
drwxrwxr-x. ... .config
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.
      - Note the new `rpmbuild` directory and the new `.rpmmacros` file.

```
$ rpmdev-setuptree
$ ls -la
...
-rw-r--r--. ... .bash_logout
-rw-r--r--. ... .bash_profile
```

```
-rw-r--r--. ... .bashrc
drwxrwxr-x. ... .cache
drwxrwxr-x. ... .config
drwxr-xr-x. ... .mozilla
drwxrwxr-x. ... rpmbuild
-rw-rw-r--. ... .rpmmacros
```

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 `rpmdev-newspec` to create a skeleton `spec` file.

```
$ rpmdev-newspec SPECS/hello.spec
SPECS/hello.spec created; type minimal, rpm version >= 4.11.
```

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 preconfigured `hello.spec` file exists on **dom0** (`192.0.2.1`) in the `/OVS/seed_pool/sfws` directory.

- You can edit the `hello.spec` file as follows by using the `vi` command, or you can use the `sftp root@192.0.2.1` command 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 6j.
  - 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 `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

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` 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
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. 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
/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.

i. Leave the `%changelog` section unchanged. Save the file and exit `vi`.

j. 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

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
install -d $RPM_BUILD_ROOT/usr/local/bin
install hello $RPM_BUILD_ROOT/usr/local/bin/hello

%files
/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.
      - 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`.

```
$ rpmbuild -bb -v SPECS/hello.spec
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
...
+ exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
...
+ exit 0
Executing(%install): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
...
```

```
Wrote: /home/rpmbuilder/rpmbuild/RPMS/x86_64/hello-debuginfo...
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp...
+ umask 022
+ cd /home/rpmbuilder/rpmbuild/BUILD
...
+ exit 0
```

c. View the new RPM package in the RPMS directory.

- The package appears with the version and release specified in the `hello.spec` file.
- Note that a `hello-debuginfo` file is also created.

```
$ cd RPMS
$ ls
x86_64
$ cd x86_64
$ ls
hello-1.0-1.el7.x86_64.rpm
hello-debuginfo-1.0-1.el7.x86_64.rpm
```

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.el7.x86_64.rpm
hello-debuginfo-1.0-1.el7.x86_64.rpm
```

c. Use the `rpm` command to install the `hello` package:

```
# rpm -ivh hello-1.0-1.el7.x86_64.rpm
Preparing...                ############################## [100%]
Updating / installing...
   1:hello-1.0-1.el7        ############################## [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
```

Practice 8-5: 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.

You also change the frequency at which the Software Update program checks for updates.

PackageKit also includes the Software graphical tool to install and remove packages, 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`.

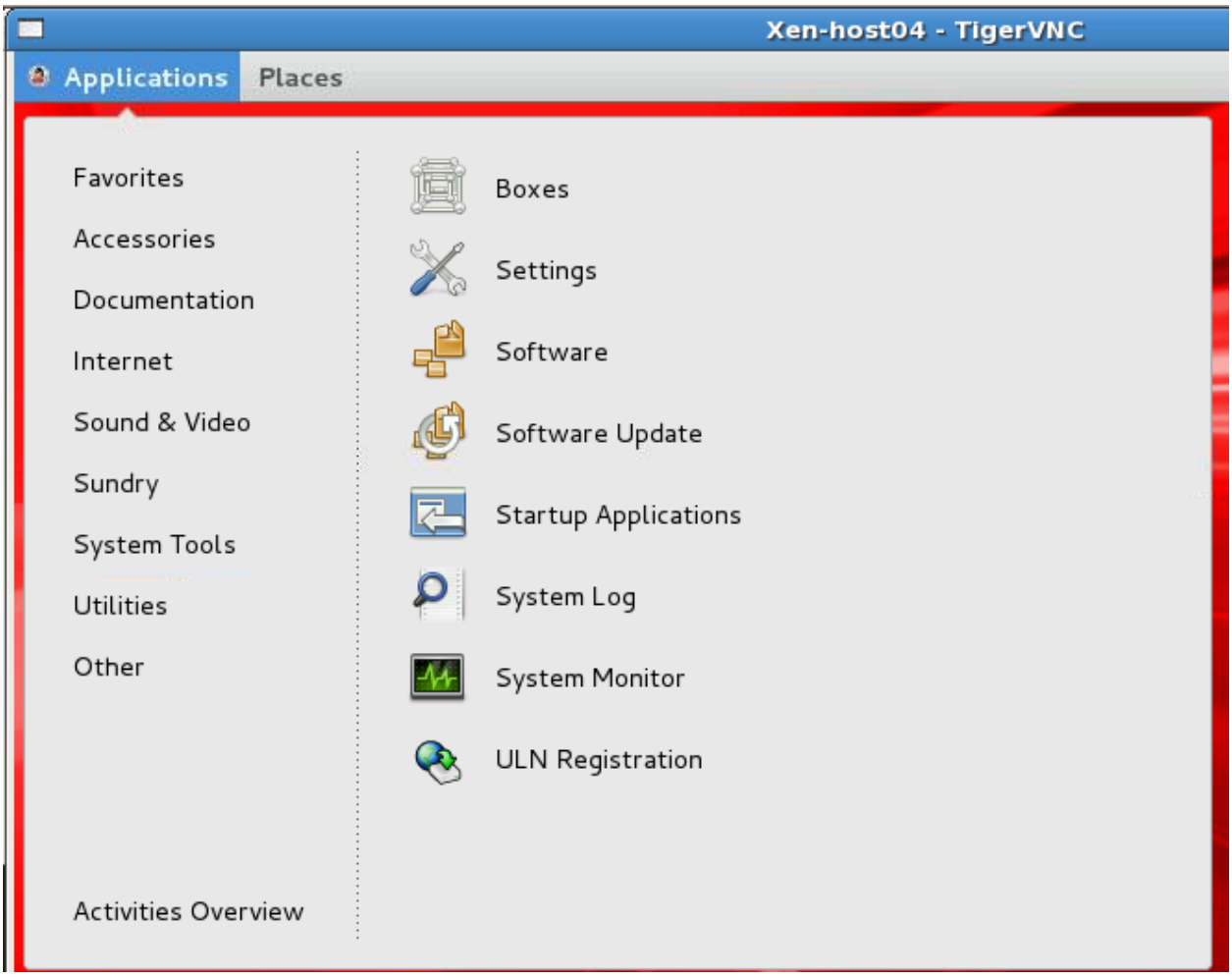
Tasks

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.

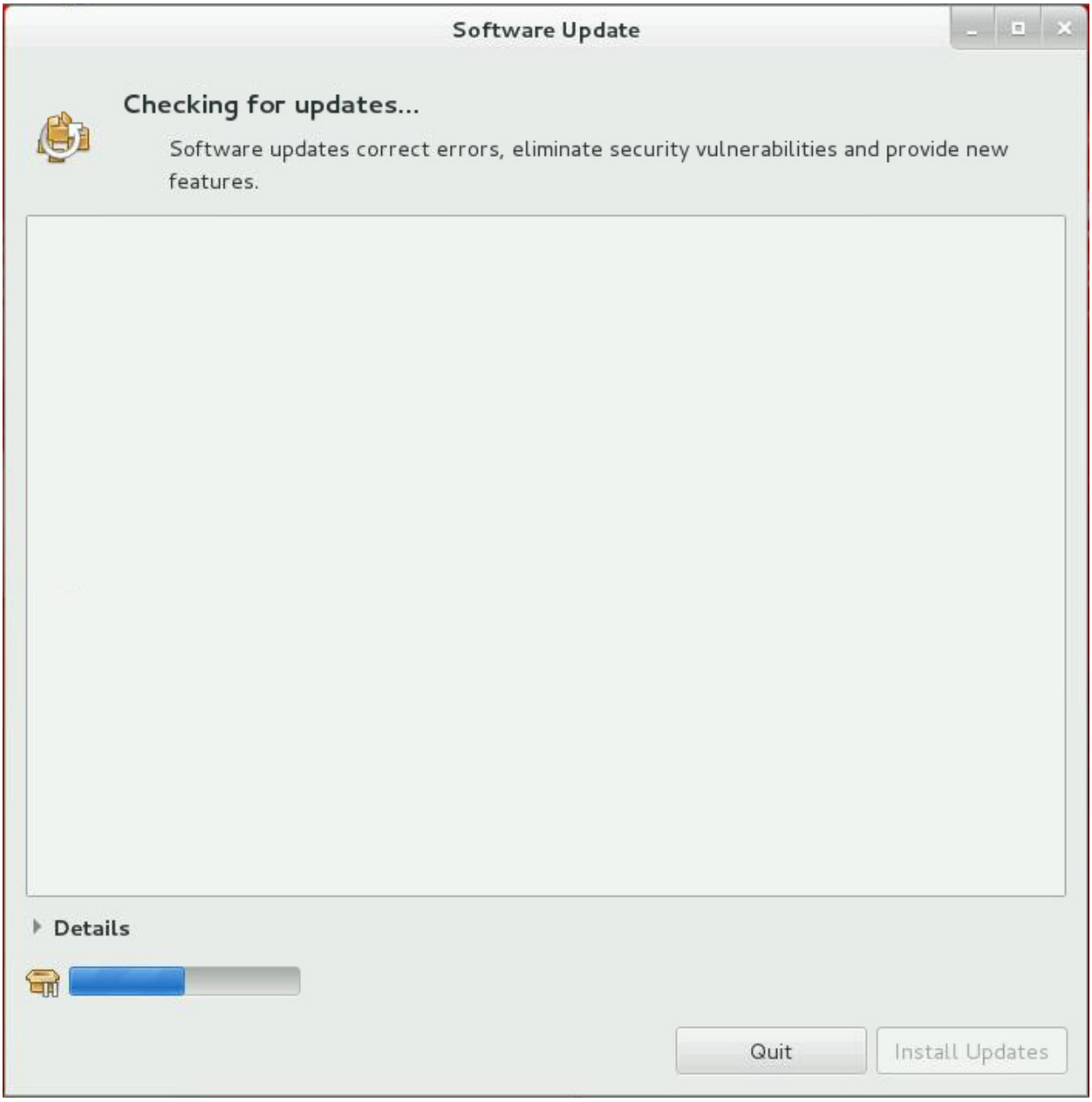
   As the `root` user on **host04**, use the `vi` editor to edit the `/etc/yum.conf` file and add the following "`proxy`" line following the "`installonly_limit=3`" line.

```
# vi /etc/yum.conf
[main]
...
installonly_limit=3
proxy=http://ges-proxy.us.oracle.com:80
```

2. Launch Software Update.
   a. In the GNOME task bar, select “Application > System Tools > Software Update.”



- The Software Update window appears.
- “Checking for updates” might take several minutes to complete.
- Continue with step 2b 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.

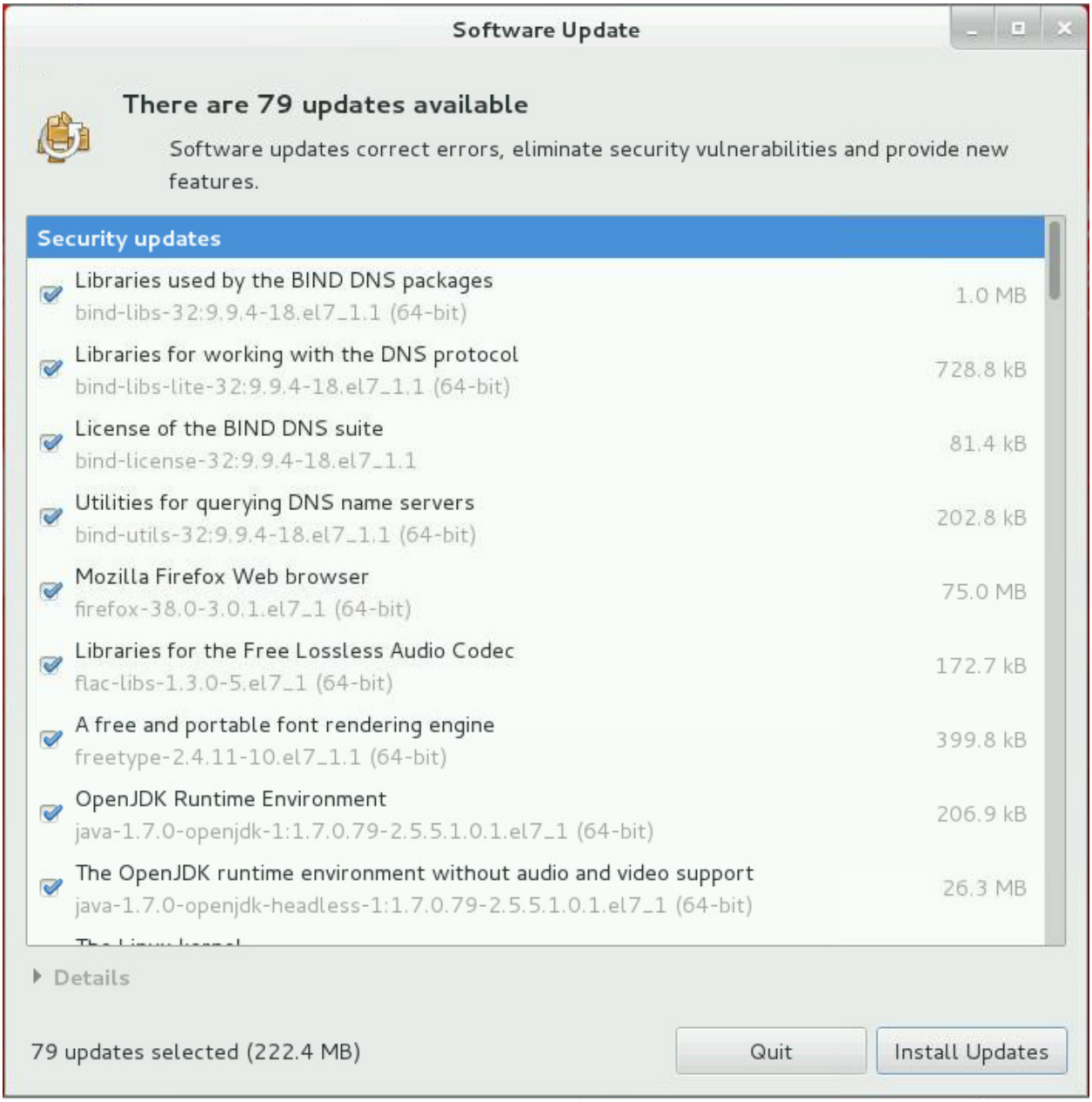
- The `yumBackend.py get-updates` is the PackageKit program that checks for updates.

```
$ ps -ef | grep yum
root  ...  /usr/bin/python
/usr/share/PackageKit/helpers/yum/yumBackend.py get-updates
newest
...
```

3. View the update(s) flagged by the Software Update program.
   - If the Software Update program fails with an error message, use the `yum clean all` command to clean all cached information and then use the `yum repolist` command to initialize the metadata. For example:

```
# yum clean all
...
# yum repolist
...
```

   a. Return to the Software Update program. In this example, the program has found 79 updates.
      - This is sample output. Your environment might be different because updates have been added since this example was captured.
      - **Do not click the "Install Updates" button** because it takes too long to install all of the updates.



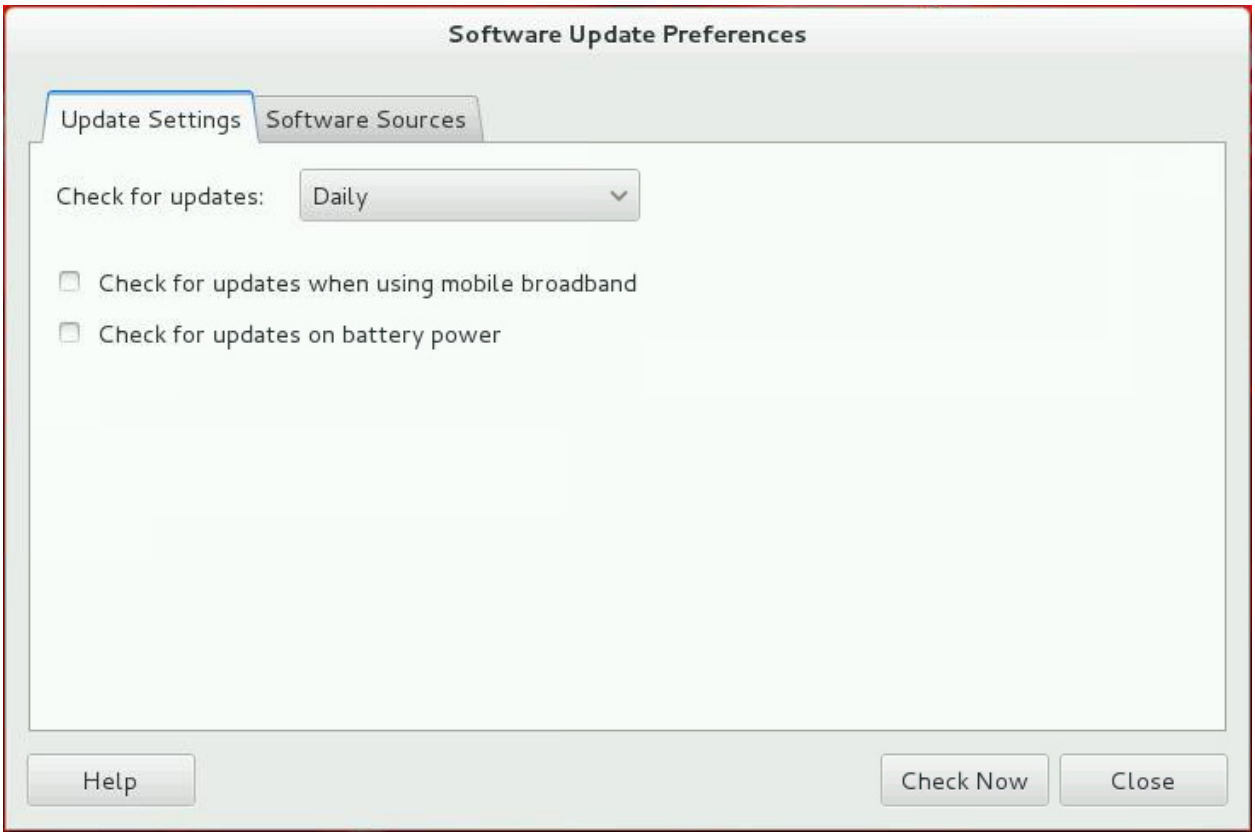
b. Scroll down through the list of updates.
   - Note that there are “Security updates,” “Bug fix updates,” and “Other updates.”

c. Click “Quit” to exit the Software Update program.

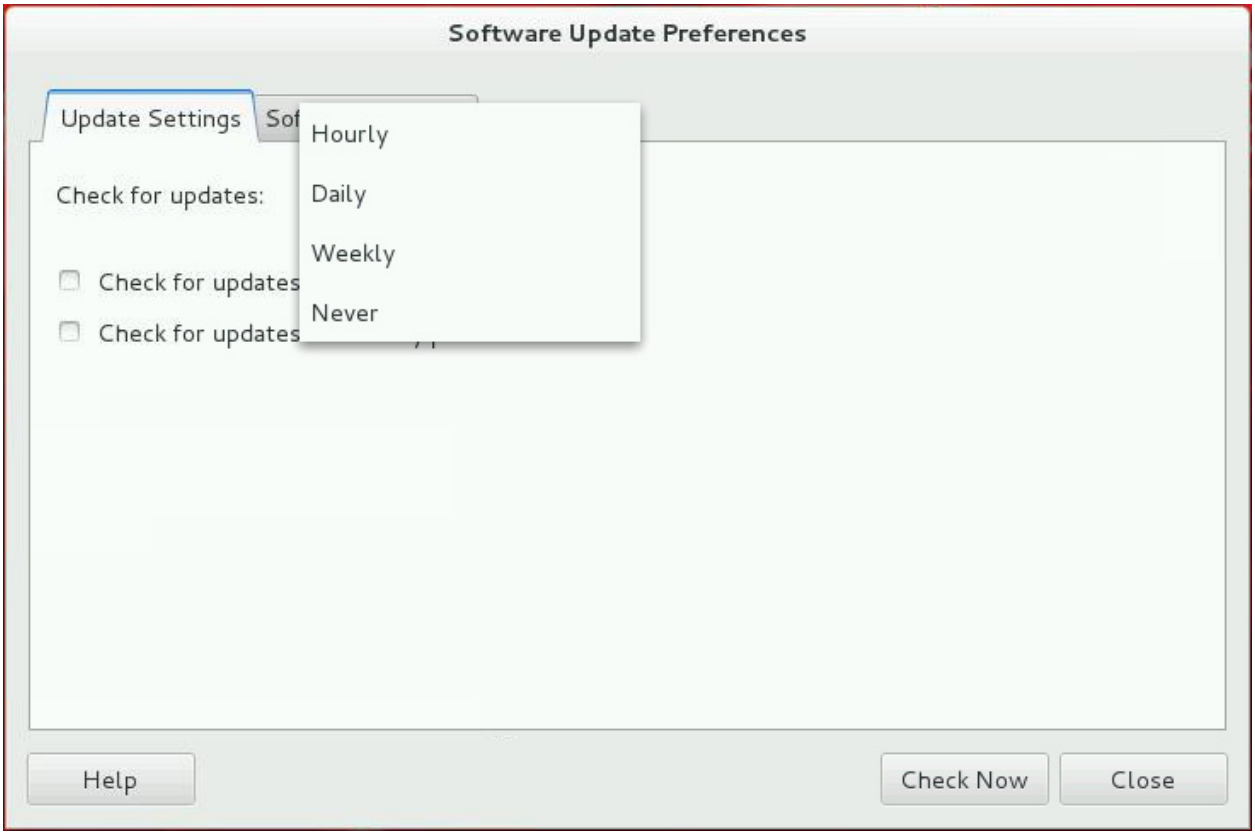
4. Change the frequency at which the Software Update program checks for updates.
   a. From a terminal window as the `root` user, run the `gpk-prefs` command to view the Software Update Preferences GUI.

```
# gpk-prefs
```

   - The “Software Update Preferences” window appears.

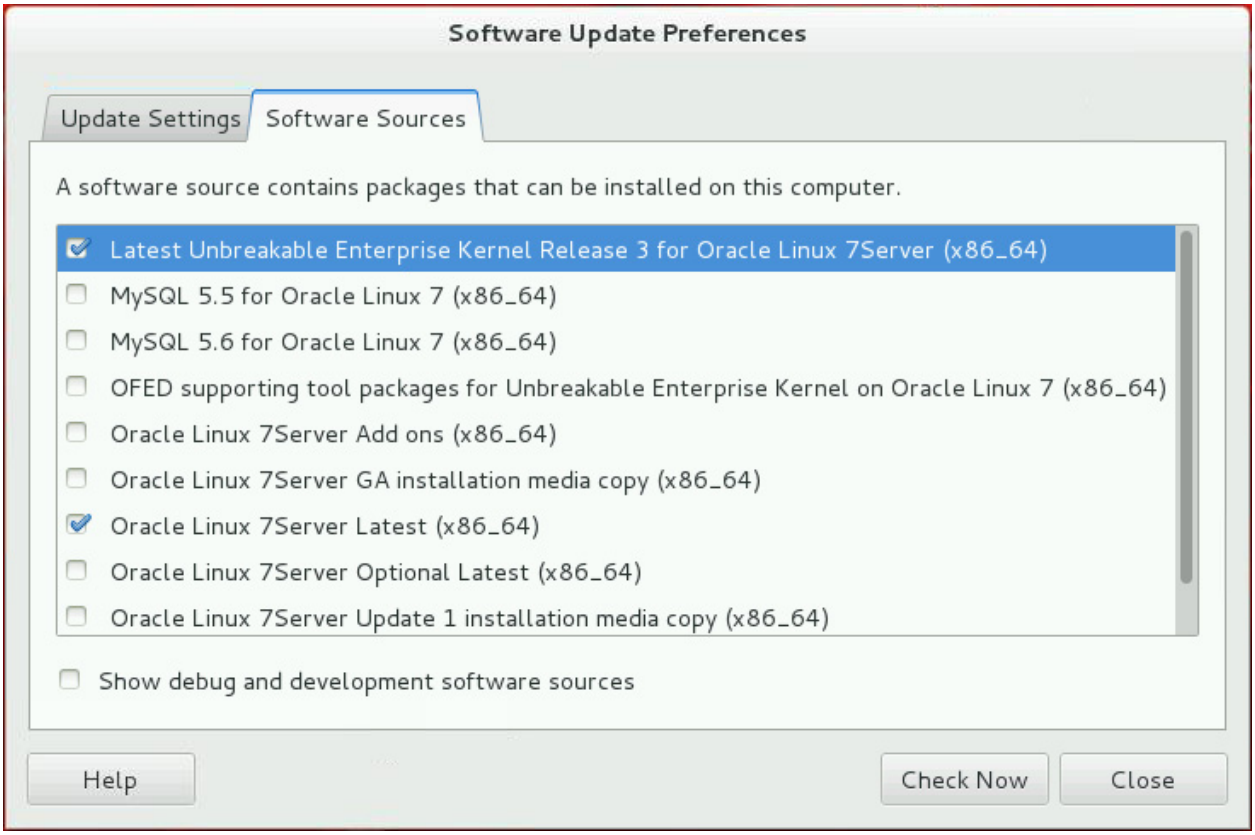


   b. Select the “Check for updates:” drop-down menu.
      - The choices are “Hourly,” “Daily,” “Weekly,” and “Never” as shown.



c. From the drop-down list, select “Hourly.”
   - Many Linux administrators prefer to manage software installation and updates by using the Yum commands directly, rather than use the graphical interfaces offered by the PackageKit programs.
   - You can select “Never” from the drop-down list to disable the Software Updates applet that checks for software updates.

d. Select the “Software Sources” tab.
   - From this window, you can enable additional Public Yum repositories.



e. Click “Close” to close the “Software Update Preferences” window.

Practice 8-6: 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.
      - Ignore the "Warning: RPMDB altered outside of yum" message. This message is caused by using `rpm` commands and can be ignored. See the following for more information: http://illiterat.livejournal.com/7834.html.

```
# yum history list
Loaded plugins: aliases, langpacks
ID     | Login user              | Date and time | Action(s)    |
Altered
-------------------------------------------------------------------------------
     6 | Oracle Student <oracle> | <date_time>   | Install      |    7
     5 | Oracle Student <oracle> | <date_time>   | Update       |    2
     4 | Oracle Student <oracle> | <date_time>   | Install      |    1
     3 | Oracle Student <oracle> | <date_time>   | Erase        |    1
     2 | Oracle Student <oracle> | <date_time>   | Install      |    1
     1 | System <unset>          | <date_time>   | Install      | 1214
Warning: RPMDB altered outside of yum.
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 `6`.

```
# yum history info 6
Loaded plugins: aliases, langpacks
ID     | Command line            | Date and time | Action(s)    |
Altered
```

```
-------------------------------------------------------------------------
     6 | install rpmdevtools     | <date_time>   | Install     |    7
history list
```

2. Install the `changelog` Yum plug-in and uninstall it by using information in the Yum history.
   - You install this plug-in package and you uninstall it in this task.

   a. Install the `yum-plugin-changelog` package by using the `yum install` command.
      - Answer `y` to "Is this ok."

```
# yum install yum-plugin-changelog
...
Transaction Summary
================================================================
Install  1 Package (+1 Dependent package)

Total download size: 114 k
Installed size: 384 k
Is this ok [y/d/N]: y
...
Complete!
```

   b. List the Yum history to display the latest transaction.
      - The most recent transaction reflects the action taken when installing the `yum-plugin-changelog` package. Two packages were installed as part of that transaction.

```
# yum history list
Loaded plugins: aliases, changelog, langpacks
ID     | Login user     | Date and time    | Action(s)|Altered
-------------------------------------------------------------------
     7 | Oracle Student <oracle>  ...      | Install  |   2
...
history list
```

   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.
      - Answer `y` to "Is this ok."

```
# yum history undo 7
...
Transaction Summary
================================================================
Remove  2 Packages

Installed size: 384 k
Is this ok [y/N]: y
```

```
...
Complete!
```

d. List the history again to examine the latest transaction information.

- The packages installed by installing the `yum-presto` package are uninstalled when you use the `yum history undo` command.

```
# yum history list
Loaded plugins: aliases, changelog, langpacks
ID     | Login user     | Date and time   | Action(s)| Altered
-------------------------------------------------------------
     8 | Oracle Student <oracle>  ...      | Erase   |    2
     7 | Oracle Student <oracle>  ...      | Install |    2
...
history list
```

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 `7Server` directory. Use the `ls –l` command to display the contents of this directory.

- In the `/var/cache/yum/x86_64/7Server` directory, there is a subdirectory for each enabled repository.

```
# ls
x86_64
# cd x86_64/
# ls
7Server
# cd 7Server/
# ls -l
drwxr-xr-x. ...  ol7_latest
drwxr-xr-x. ...  ol7_UEKR3
-rw-r--r--. ...  timedhosts
```

c. Use the `cd` command to change to the `ol7_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/OL7/latest/` repository.
- 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.

```
# cd ol7_latest
# ls -l
-rw-r--r--. ...  cachecookie
-rw-r--r--. ...  comps.xml
-rw-r--r--. ...  filelists.xml.gz
drwxr-xr-x. ...  gen
-rw-r--r--. ...  other.xml.gz
drwxr-xr-x. ...  packages
-rw-r--r--. ...  primary.xml.gz
-rw-r--r--. ...  repomd.xml
-rw-r--r--. ...  updateinfo.xml.gz
```

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

- In your environment, package caching is disabled but the packages that you downloaded are still present because you have not installed these packages. Packages are deleted after they are installed when package caching is disabled.

```
# ls -l packages
-rw-r--r--. ...  compat-libcap1-1.10-7.el7_1.x86_64.rpm
...
```

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: aliases, changelog, langpacks
Cleaning repos: ol7_UEKR3 ol7_latest
... package files removed
```

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

- The packages are no longer present.

```
# ls packages
```

c. Use the `ls -l` command to list the contents of the `gen` directory.

- This directory contains the uncompressed data from `updateinfo.xml.gz`.

```
# ls -l gen
-rw-r--r--. ...  filelists.xml
-rw-r--r--. ...  filelists.xml.sqlite
-rw-r--r--. ...  other.xml
-rw-r--r--. ...  other.xml.sqlite
-rw-r--r--. ...  primary.xml
-rw-r--r--. ...  primary.xml.sqlite
-rw-r--r--. ...  updateinfo.xml
```

d. Use the `yum clean metadata` command to clean the metadata in the Yum cache.
   - The number of files removed might differ in your environment.

```
# yum clean metadata
Loaded plugins: aliases, changelog, langpacks
Cleaning repos: ol7_UEKR3 ol7_latest
19 metadata files removed
5 sqlite files removed
0 metadata files removed
```

e. Use the `ls -l` command to list the contents of the current directory, `/var/cache/yum/x86_64/7Server/ ol7_latest` and the `gen` subdirectory, and note the effect of the `yum clean metadata` command.
   - 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.

```
# ls -l
drwxr-xr-x. ...  gen
drwxr-xr-x. ...  packages
# ls -l gen
total 0
```

5. Shut down **host04** and start **host03**.
   a. Use the `systemctl poweroff` command to shut down **host04**.
      - Your VNC window closes.

```
# systemctl poweroff
```

   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.

- Only the **host01**, **host02**, and **host03** VMs are 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                                      15    1536      1      -b----         37.3
```

Practices for Lesson 9: Advanced Storage Administration

Chapter 9

Practices for Lesson 9: Overview

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 9-1: 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** by using `vncviewer`.
   a. If necessary, refer to Practice 3-1: Configuring an OpenLDAP Server for instructions on connecting with `vncviewer`.
   b. Open a terminal window and become the `root` user on **host03**.
2. Partition a storage device using `fdisk`.
   a. As the `root` user on **host03**, use the `fdisk` command to display the partition table.
      - 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.

```
# fdisk -l | grep /dev
Disk /dev/xvda: 12.9 GB, 12884901888 bytes, 25165824 sectors
/dev/xvda1   *    2048    1026047     512000   83   Linux
/dev/xvda2     1026048   25165823   12069888   8e   Linux LVM
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/mapper/ol-root: 11.0 GB, 11022630912 bytes, ...
Disk /dev/mapper/ol-swap: 1287 MB, 1287651328 bytes, ...
```

   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
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): ENTER
Using default response p
Partition number (1-4, default 1): ENTER
First sector (2048-20971519, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-20971519, default
20971519): +1G
Partition of type Linux and of size 1 GiB is set

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, 20971520 sectors
...
    Device Boot   Start      End     Blocks     Id    System
/dev/xvdb1         2048  2099199    1048576     83    Linux
```

3. 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 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
...
Writing superblocks and filesystem accounting information: done
```

4. 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      976M  2.6M   907M    1%  /Dev
```

Practice 9-2: Implementing Access Control Lists

Overview

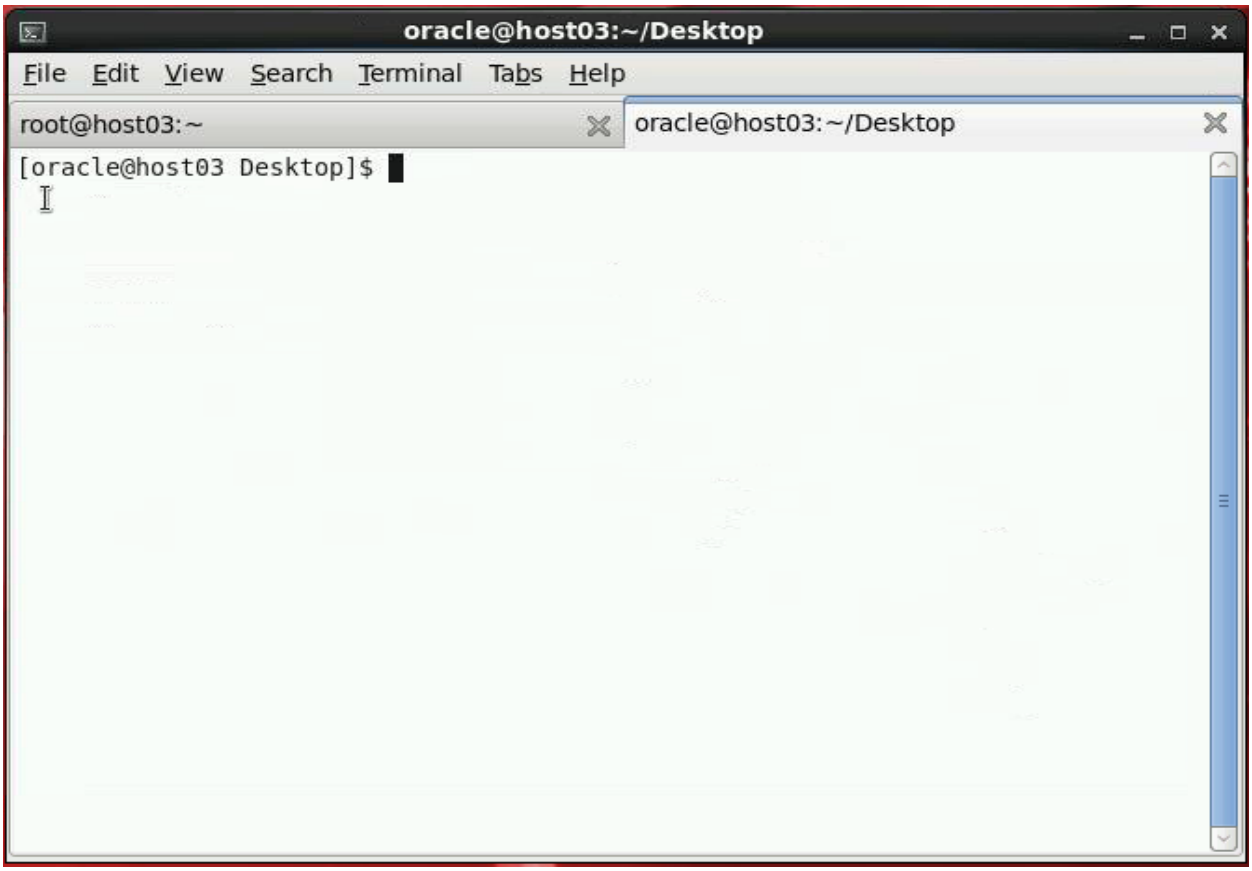
In this practice, you set ACLs on a directory.

Assumptions

- Ensure that you are using `vncviewer` to connect to **host03** and not using `ssh`.
- You are the `root` user on **host03** VM.
- You 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 looks like 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.
   - Note that you do not have permission to create files in the `/Dev` directory.

```
[oracle@host03]$ touch /Dev/test
touch: cannot touch 'Dev/test': Permission denied
```

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.
   - Note the new `user:oracle:rwx` line in the output of the `getfacl` command.

```
[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
```

6. As the `root` user, use the `ls -ld` command to display the permissions for the `/Dev` directory.
   - Note the plus sign (`+`), indicating that the directory has an ACL.

```
[root@host03]# ls -ld /Dev
drwxrwxr-x+ ... /Dev
```

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.
   - Note that the command succeeded this time.

```
 [oracle@host03]$ touch /Dev/test
```

8. As the `oracle` user, use the `ls` command to display a long listing of the `/Dev` directory.
   - Note that the `test` file is owned by the `oracle` user.

```
[oracle@host03]$ ls -l /Dev
drwx------. 2 root    root ...   lost+found
-rw-rw-r--. 1 oracle  oracle ... test
```

Practice 9-3: 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, configure disk quotas.
   a. Click the "`root@host03`" tab to enter commands as the `root` user.
   b. Use the `umount` command to unmount the file system on `/Dev`.

```
[root@host03]# umount /Dev
```

   c. Use the `mount` command with the `-o acl,usrquota,grpquota` options to remount `/dev/xvdb1` on `/Dev`.
      - These options enable disk quotas for users and groups and also enable ACL support.

```
[root@host03]# mount -t ext4 -o acl,usrquota,grpquota /dev/xvdb1
/Dev
```

   d. Use the `quotacheck` command to create disk usage tables for `/Dev`.

```
[root@host03]# quotacheck -cug /Dev
```

   e. 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
...
```

   f. Use the `quotaon` command to enable quotas on `/Dev`.

```
[root@host03]# quotaon /Dev
```

   g. 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
```

   h. 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.

i. Use the `repquota` command to report disk usage on `/Dev`.

- Note that the hard limit for the `oracle` user is now 2048.

```
[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
```

2. As the `oracle` user, verify the disk quota setting.

a. Click the "`oracle@host03`" tab to enter commands as the `oracle` user.

b. Use the `dd if=/dev/zero of=bigfile bs=1M count=4096` command to attempt to create a 4 MB file on `/Dev`.

- Note the "Disk quota exceeded" error message.

```
[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
2097152 bytes (2.1 MB) copied, ...
```

c. Use the `ls` command to display a long listing of the `/Dev` directory.

- Note that the `bigfile` is not 4 MB, but was truncated after quota limits were reached.

```
[oracle@host03]$ ls -l /Dev
...
-rw-rw-r--. 1 oracle  oracle 2097152 ... bigfile
...
```

d. 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
```

e. Use the `rm` command to delete the `bigfile` file in the `/Dev` directory.

```
[oracle@host03]$ rm bigfile
```

f. Use the `quota` command to display quota information.

- Note the difference in the number of blocks and number of files from step 13.

```
[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
```

g. Use the `rm` command to delete the `test` file in the `/Dev` directory.

```
[oracle@host03]$ rm test
```

h. Use the `cd` command to change to the oracle user's home directory.

```
[oracle@host03]$ cd
```

3. As the `root` user, reset the `/dev/xvdb1` partition for the next practice.

a. Click the "`root@host03`" tab to enter commands as the `root` user.

b. Use the `setquota oracle 0 0 0 0 /Dev` command to reset the disk quota for the `oracle` user.

```
[root@host03]# setquota oracle 0 0 0 0 /Dev
```

c. Use the `setfacl` command to remove the ACL from the `/Dev` directory.

```
[root@host03]# setfacl -b /Dev
```

d. Use the `getfacl` command to display the `/Dev` directory's ACL.

- Note that the `user:oracle:rwx` line in the output has been removed.

```
[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
```

e. Use the `ls -ld` command to display the permissions for the `/Dev` directory.

- Note that there is no plus sign (`+`), indicating that the directory has no ACL.

```
[root@host03]# ls -ld /Dev
drwxr-xr-x ... /Dev
```

f. Use the `umount` command to unmount /Dev.

```
# umount /Dev
```

g. Click the "X" on the "`oracle@host03`" tab to close the tab.

Practice 9-4: 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. Set up a cryptographic volume.
   a. Use the `cryptsetup` command with `luksFormat` to initialize the `/dev/xvdb1` volume and set an initial key of `Cvt69*@P3`.
      - The "`Cvt69*@P3`" entries are not displayed for security reasons.
      - The initial key needs to be a random sequence of characters to be accepted.

```
# cryptsetup luksFormat /dev/xvdb1

WARNING!
========
This will overwrite data on /dev/xvdb1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: Cvt69*@P3
Verify passphrase: Cvt69*@P3
```

   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: Cvt69*@P3
```

   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-xts-plain64
  keysize: 256 bits
  device:  /dev/xvdb1
  offset:  4096 sectors
  size:    2093056 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
...
crw-------. ... control
lrwxrwxrwx. ... cryptfs -> ../dm-2
lrwxrwxrwx. ... ol-root -> ../dm-0
lrwxrwxrwx. ... ol-swap -> ../dm-1
```

2. 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 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
...
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: 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
                990M  2.6M   921M    1%  /cryptfs
```

e. Use the `vi` editor to create `/etc/crypttab` and to add the following entry.

```
# vi /etc/crypttab
cryptfs /dev/xvdb1 none luks
```

3. Reboot your system and enter the passphrase to mount the encrypted file system.

a. Use the `systemctl reboot` command to reboot your system.

- After you reboot your system, your VNC session closes.

```
# systemctl reboot
```

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 `5904`, enter the following and click "Connect."

```
localhost:5904
```

d. Provide the passphrase, `Cvt69*@P3`, when prompted for the encrypted file system passphrase during reboot.

- The boot process continues after providing the correct passphrase.

```
Please enter passphrase for disk cryptfs!: Cvt69*@P3
```

4. 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. Using the `vi` editor, remove the following entry from `/etc/crypttab`.

```
# vi /etc/crypttab
cryptfs /dev/xvdb2 none luks
```

   e. Use the `cryptsetup` command with `luksOpen` to remove the device mapping.

```
# cryptsetup luksClose /dev/mapper/cryptfs
```

   f. Verify that the `cryptfs` device mapping has been removed.

```
# ls /dev/mapper
control  ol-root  ol-swap
```

Practice 9-5: 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--         737 vm.cfg
```

   c. Use the `cat` command to view the `vm.cfg` file.
      - 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`.

```
[dom0]# cat vm.cfg
name = "host03"
builder = "hvm"
memory = "1536"
boot = 'cd'
disk = [ 'file:/OVS/running_pool/host03/system.img,xvda,w',
         'file:/OVS/running_pool/host03/u01.img,xvdb,w',
         'file:/OVS/running_pool/host03/u02.img,xvdd,w',
...
```

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.
      - The output shows that the `system.img` disk image file contains two partitions.

- Sample output is shown.

```
[dom0]# kpartx -l system.img
loop8p1 : 0 1024000 /dev/loop8 2048
loop8p2 : 0 24139776 /dev/loop8 1026048
```

b. From **host03** VM, use the `fdisk` command to list the partition table for `/dev/xvda`.

- Note that `/dev/xvda` has two partitions.
- This confirms that the `system.img` file is mapped to `/dev/xvda`.

```
[host03]# fdisk -l | grep /dev/xvda
Disk /dev/xvda: 12.9 GB, 12884901888 bytes, 25165824 sectors
/dev/xvda1   *    2048    1026047     512000   83   Linux
/dev/xvda2     1026048   25165823   12069888   8e   Linux LVM
```

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.

- The output shows one partition.
- Sample output is shown.

```
[dom0]# kpartx -l u01.img
loop8p1 : 0 2097152 /dev/loop8 2048
```

b. From **host03** VM, use the `fdisk` command to list the partition table for `/dev/xvdb`.

- The output shows one partition.
- This confirms that the `u01.img` file is mapped to `/dev/xvdb`.

```
[host03]# fdisk -l | grep /dev/xvdb
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
/dev/xvdb1          2048  2099199    1048576     83    Linux
```

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.

- The output shows no partitions.
- Sample output is shown.

```
[dom0]# kpartx -l u02.img
```

b. From **host03** VM, use the `fdisk` command to list the partition table on `/dev/xvdd`.

- The output shows no partitions on `/dev/xvdd`.
- This confirms that the `u02.img` file is mapped to `/dev/xvdd`.

```
[host03]# fdisk -l | grep /dev/xvdd
Disk /dev/xvdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

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 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
...
Writing superblocks and filesystem accounting information: done
```

   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    1008M   34M   924M    4%  /Dev
```

   d. Use the cp command to copy the init* files from /boot to /Dev.
      - You view these files later in this practice to confirm the success of the kpartx command.

```
[host03]# cp /boot/init* /Dev
[host03]# ls /Dev
initramfs-0-rescue-...img                   initrd-plymouth.img
initramfs-3.10.0-229.el7.x86_64.img         lost+found
initramfs-3.8.13-55.1.6.el7uek.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.
      - Before adding the device files, a listing of /dev/mapper shows only the control file.

```
[dom0]# ls /dev/mapper
control
```

   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.
      - Sample output is shown.
      - This confirms that there is one partition on /dev/xvdb.

```
[dom0]# kpartx -l u01.img
loop9p1 : 0 2097152 /dev/loop9 2048
```

c. Use the `kpartx -a` command to add the device mappings for the detected partitions.
   - To save time in this practice, you do not need to shut down the **host03** VM before using the `kpartx -a` command.
   - It would be best practice 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.
   - Sample output is shown.
   - Note that a file was created for the partition on `/dev/xvdb`.

```
[dom0]# ls /dev/mapper
control   loop9p1
```

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/loop9p1` on `/mnt/map1`.
      - Substitute the device name from step 6d.

```
[dom0]# mount /dev/mapper/loop9p1 /mnt/map1
```

   c. Use the `ls` command to view the files on `/mnt/map1`.
      - Note that these are the same files that you copied to `/Dev` in step 5d.

```
[dom0]# ls /mnt/map1
initramfs-0-rescue-...img                    initrd-plymouth.img
initramfs-3.10.0-229.el7.x86_64.img          lost+found
initramfs-3.8.13-55.1.6.el7uek.x86_64.img
```

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/loop9
```

   d. Use the `ls` command to list the contents of `/dev/mapper`.
      - Note that the device mapping no longer exists in `/dev/mapper`.

```
[domo]# ls /dev/mapper
control
```

Practice 9-6: Exploring and Configuring Udev

Overview

In this practice, you:

- Explore Udev files and directories
- Query the Udev database
- Create a Udev rule to create a symbolic link to a device

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Explore Udev.
   - Udev is now part of `systemd`.
   a. Use the `rpm -ql` command to view the "`udev`" files included with the `systemd` RPM package.

```
# rpm -ql systemd | grep udev
/etc/udev
/etc/udev/hwdb.bin
/etc/udev/rules.d
/etc/udev/udev.conf
/usr/bin/udevadm
...
```

   b. 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
100-balloon.rules       75-probe_mtd.rules
10-dm.rules             75-tty-description.rules
11-dm-lvm.rules         77-mm-ericsson-mbm.rules
...
# ls /etc/udev/rules.d
70-persistent-ipoib.rules
```

   c. 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.
      - Note the operators:
        - `==`: Compare for equality
        - `=`: Assign a value to a key
        - `+=`: Add the value to the current values for the key

```
# less /lib/udev/rules.d/50-udev-default.rules
# do not edit this file, it will be overwritten on update
```

```
SUBSYSTEM=="virtio-ports", KERNEL=="vport", ATTR{name}=="?*",...

# select "system RTC" or just use the first one
SUBSYSTEM=="rtc", ATTR{hctosys}=="1", SYMLINK+="rtc"
SUBSYSTEM=="rtc", KERNEL=="rtc0", SYMLINK+="rtc", OPTIONS+=...

SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", IMPORT{builtin}...
SUBSYSTEM=="input", ENV{ID_INPUT}=="", IMPORT{builtin}="input...
...
```

2. Query the Udev database.
   - Sample output is shown.
   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
E: DEVNAME=/dev/xvdd
E: DEVPATH=/devices/vbd-5696/block/xvdd
E: DEVTYPE=disk
E: MAJOR=202
E: MINOR=48
E: MPATH_SBIN_PATH=/sbin
E: SUBSYSTEM=block
E: TAGS=:systemd:
E: USEC_INITIALIZED=12403
```

   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
Udevadm info starts with the device specified by the devpath and
then walks up the chain of parent devices. It prints for every
device found, all possible attributes in the udev rules key
format. A rule to match, can be composed by the attributes of
the device and the attributes from one single parent device.

  looking at device '/devices/vbd-5696/block/xvdd':
    KERNEL=="xvdd"
    SUBSYSTEM=="block"
    DRIVER==""
```

```
    ATTR{ro}=="0"
...
  looking at parent device '/devices/vbd-5696':
    KERNELS=="vbd-5696"
    SUBSYSTEMS=="xen"
    DRIVERS=="vbd"
    ATTR{devtype}=="vbd"
    ATTR{nodename}=="device/vbd/5696"
```

3. Create a symbolic link to a device node.
   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.
      - The `SYMLINK` directive names the new symlink for the device.

```
# vi /etc/udev/rules.d/10-local.rules
KERNEL=="xvdd", SUBSYSTEM=="block", SYMLINK="my_disk"
```

   b. Run the `udevadm trigger` command to manually force Udev to trigger rules.

```
# udevadm trigger
```

   c. Use the `ls -l` command to list the `/dev/my*` devices.
      - Note that `/dev/my_disk` is a symlink to `/dev/xvdd`.

```
# ls -l /dev/my*
lrwxrwxrwx. ... /dev/my_disk -> xvdd
```

   d. Use the `udevadm info` command to query the Udev database for the symlinks for `/dev/xvdd`.

```
# udevadm info --query=symlink --name=/dev/xvdd
my_disk
```

4. Remove the `/dev/my_disk` symlink.
   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
```

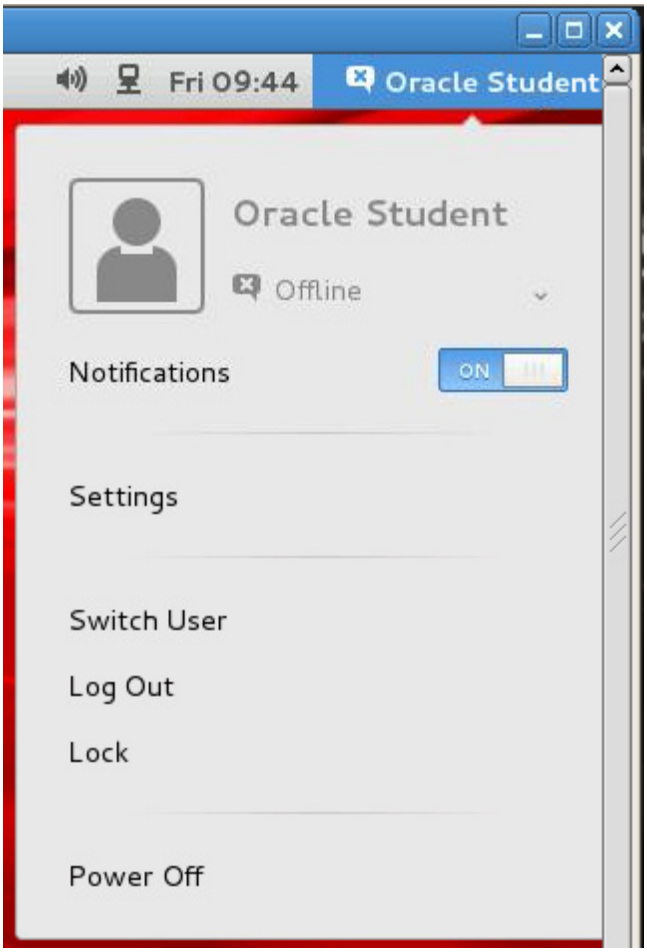
   b. Run the `udevadm trigger` command to manually force Udev to trigger rules.

```
# udevadm trigger
```

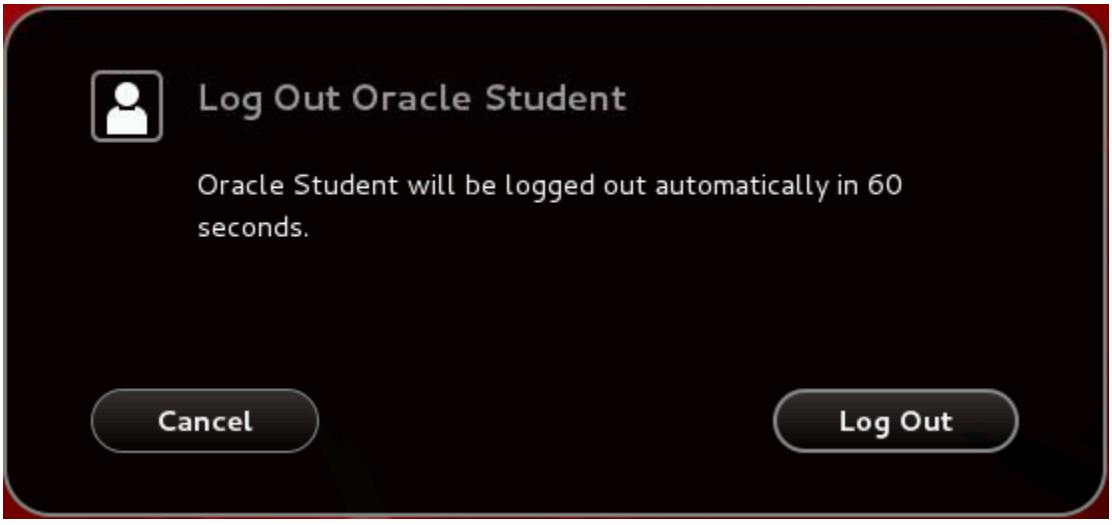
   c. Use the `ls` command to list the `/dev/my*` devices.
      - Note that `/dev/my_disk` no longer exists.

```
# ls /dev/my*
ls: cannot access /dev/my*: No such file or directory
```

5. Log off **host03**.
   a. Click the “Oracle Student” in the top-right corner of the GNOME desktop to display the drop-down menu.



   b. Click “Log Out” from the menu.
      - The following window appears.



   c. Click “Log Out.”
   d. Close the VNC window by clicking the “X” in the top-right corner of the window.

Practices for Lesson 10: Advanced Networking

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

In these practices, you do the following:

- Configure network bonding by using the GUI and the command line
- Explore network bonding interface configuration
- Configure 802.1q VLAN tagging interfaces
- Explore 802.1q VLAN tagging interface configuration
- Configure a site-to-site VPN

Practice 10-1: Configuring Network Bonding by Using the GUI

Overview

In this practice, you:

- View the network configuration on **dom0**
- Log in to **host02** by using `vncviewer`
- View the network configuration on **host02**
- Configure network bonding on **host02** by using the Network Settings GUI

Assumptions

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

Tasks

1. View the network configuration on **dom0**.

   Use the `ifconfig` command to view the network configuration.

   - The IP address of `bond0` is different on your system.
   - Note that the `virbr2` bridge is on the `192.168.2` subnet.
   - The bonded interfaces you create in this practice are also on the `192.168.2` subnet.

```
[dom0]# ifconfig
...
bond0     Link encap:Ethernet ...
          inet addr:10.150.30.83 ...
...
eth0      Link encap:Ethernet ...
...
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 ...
...
virbr2    Link encap:Ethernet ...
          inet addr:192.168.2.1 ...
...
virbr3    Link encap:Ethernet ...
          inet addr:192.168.3.1 ...
...
```

2. Log in to **host02** by using `vncviewer`.

a. From **dom0**, determine the VNC port number for **host02** by running the `xm list -l host02 | grep location` command.

```
[dom0]# xm list -l host02 | 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.

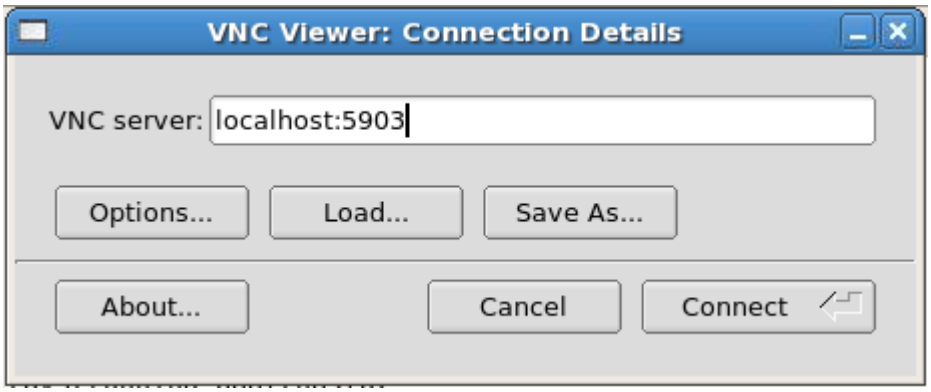
b. From **dom0**, run the `vncviewer&` command.

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

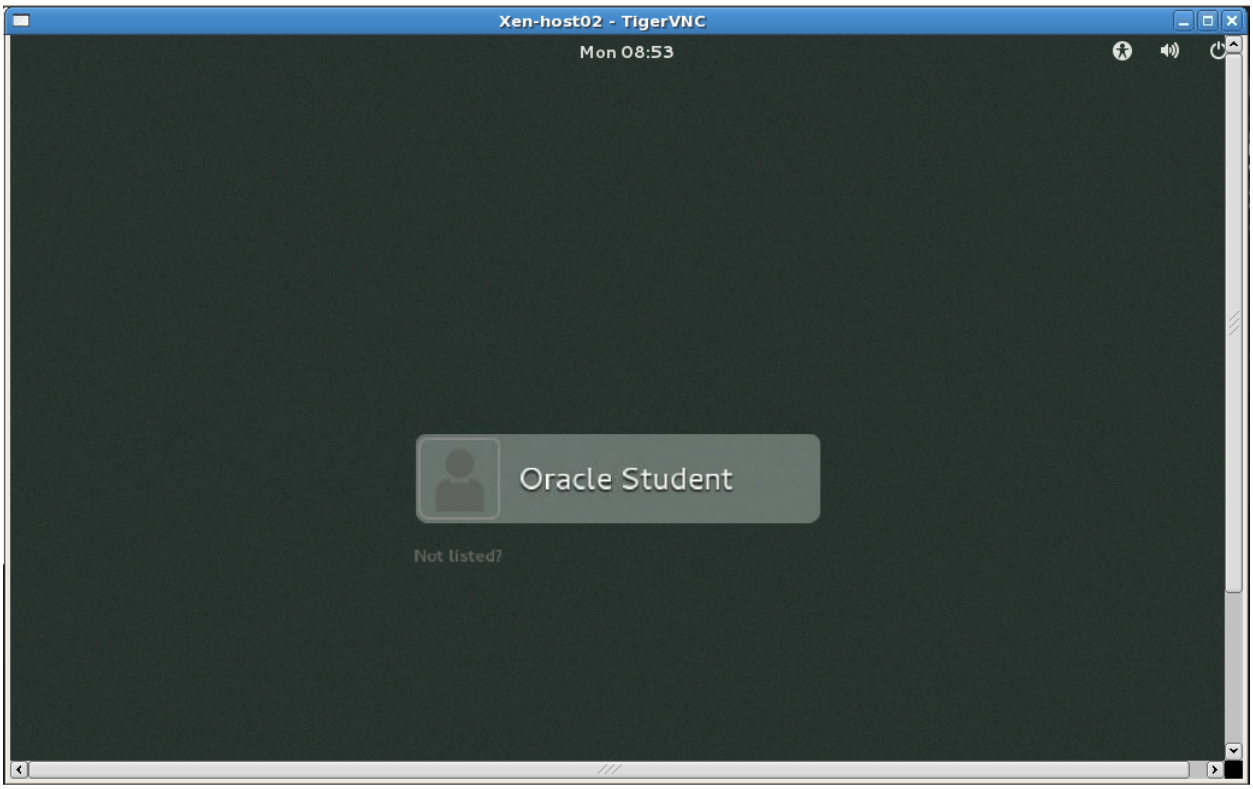
- The "VNC Viewer: Connection Details" dialog box appears.

c. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list -l host02 | grep location` command.

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



- The GNOME login screen appears.

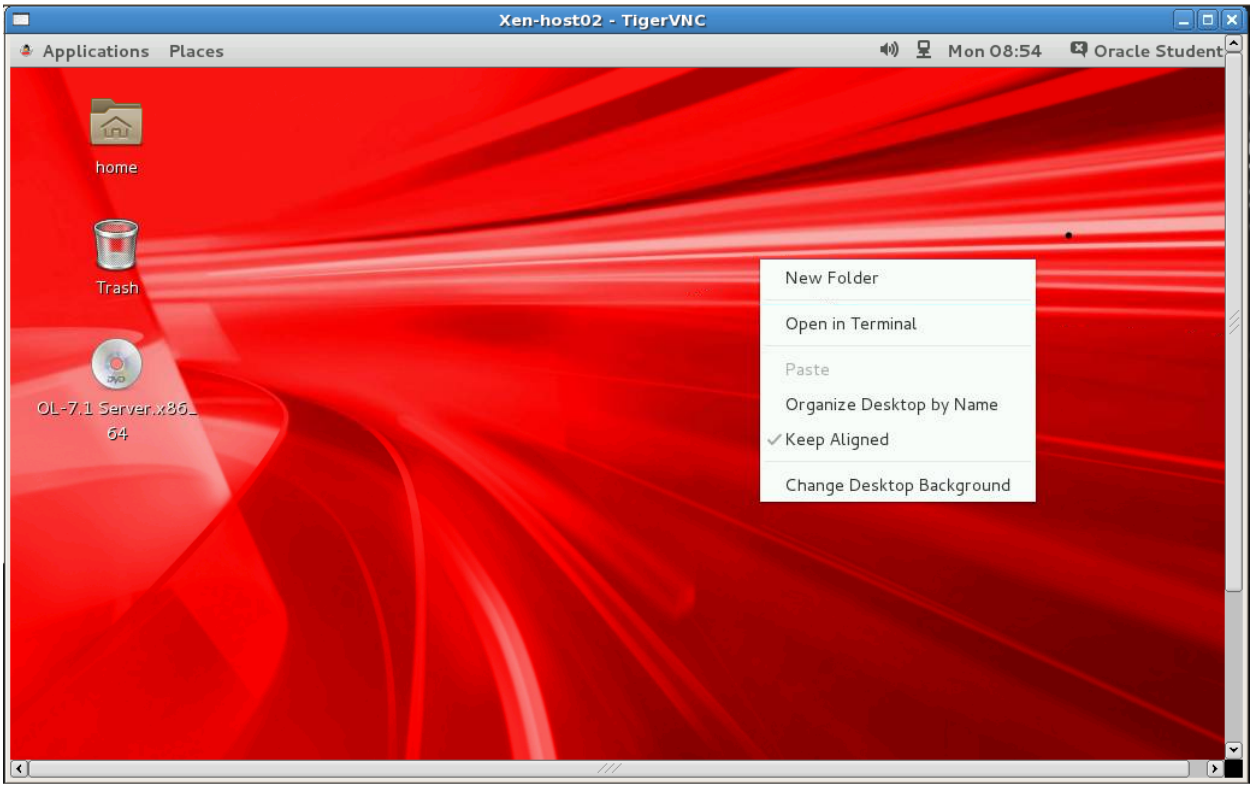


d. Click “Oracle Student” in the list of users. You are prompted for the password.

e. Enter `oracle` for the Password and click “Sign In.”

- The GNOME desktop appears.

f. Right-click 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 network interfaces on **host02**.

   a. Use the `ip addr` command to view the network interfaces.
      - Note that the `eth2` and `eth3` interfaces do not have IP addresses.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:02 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.102/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.102/24 brd 192.168.1.255 scope global eth1
```

```
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:04:02 brd ff:ff:ff:ff:ff:ff
```

b. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

- Note that the `eth0`, `eth1`, `eth2` and `eth3` Ethernet network interfaces have configuration files

```
# ls /etc/sysconfig/network-scripts
ifcfg-eth0  ...
ifcfg-eth1  ...
ifcfg-eth2  ...
ifcfg-eth3  ...
...
```

c. Use the `nmcli con` command to view the network connections.

- Note that the connections correspond to the existing network interface configuration files in the `/etc/sysconfig/network-scripts` directory.
- Note that the `eth2` and `eth3` connections are not associated with a device.

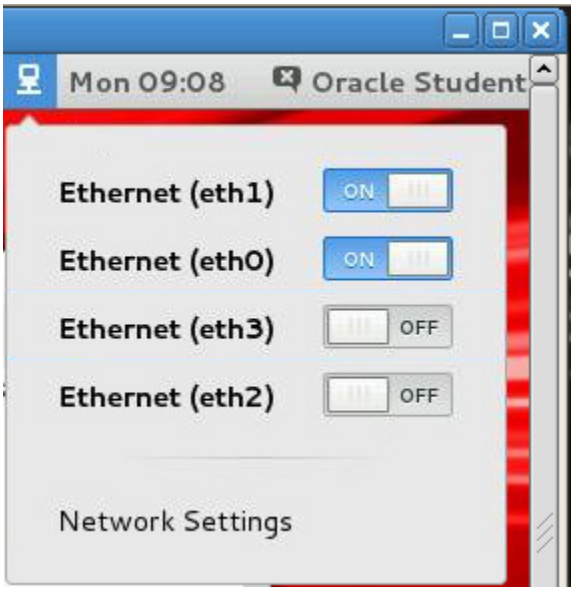
```
# nmcli con
NAME        UUID    TYPE              DEVICE
eth1        ...     802-3-ethernet    eth1
eth2        ...     802-3-ethernet    --
eth3        ...     802-3-ethernet    --
eth0        ...     802-3-ethernet    eth0
```

4. Use the “Network Settings Editor” to configure network bonding.

a. Click the network icon from the GNOME desktop notification area.

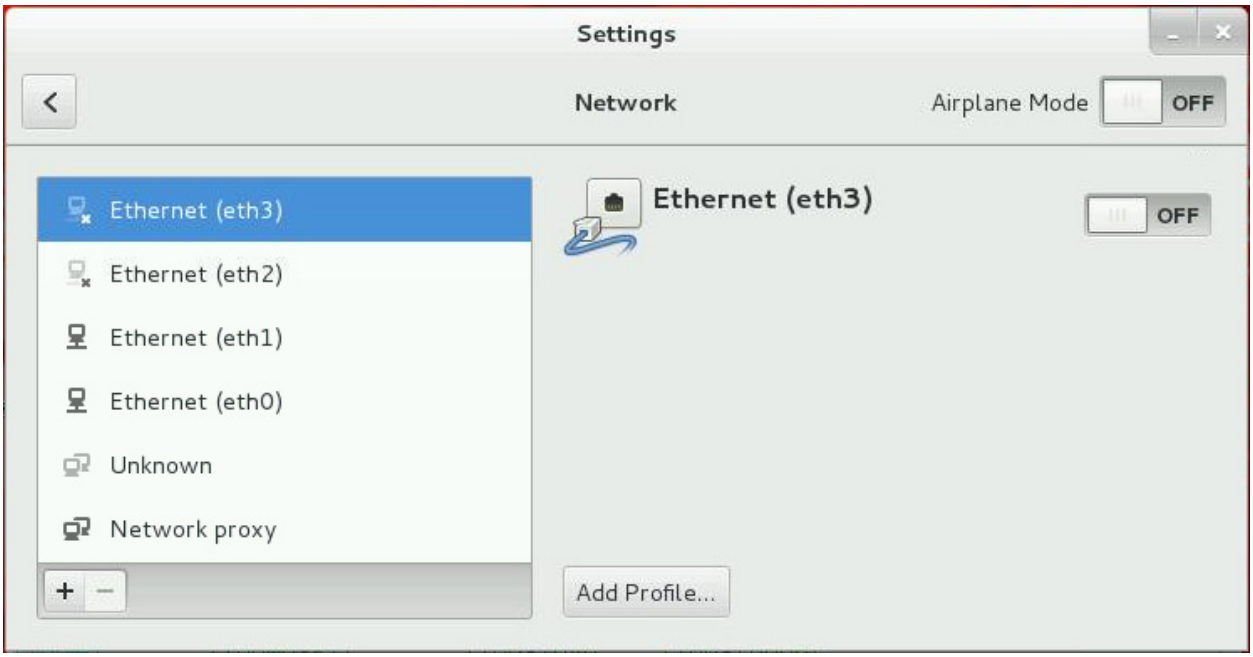
- The drop-down menu includes four Ethernet interfaces and the “Network Settings” option.

- Note that `eth2` and `eth3` are OFF.



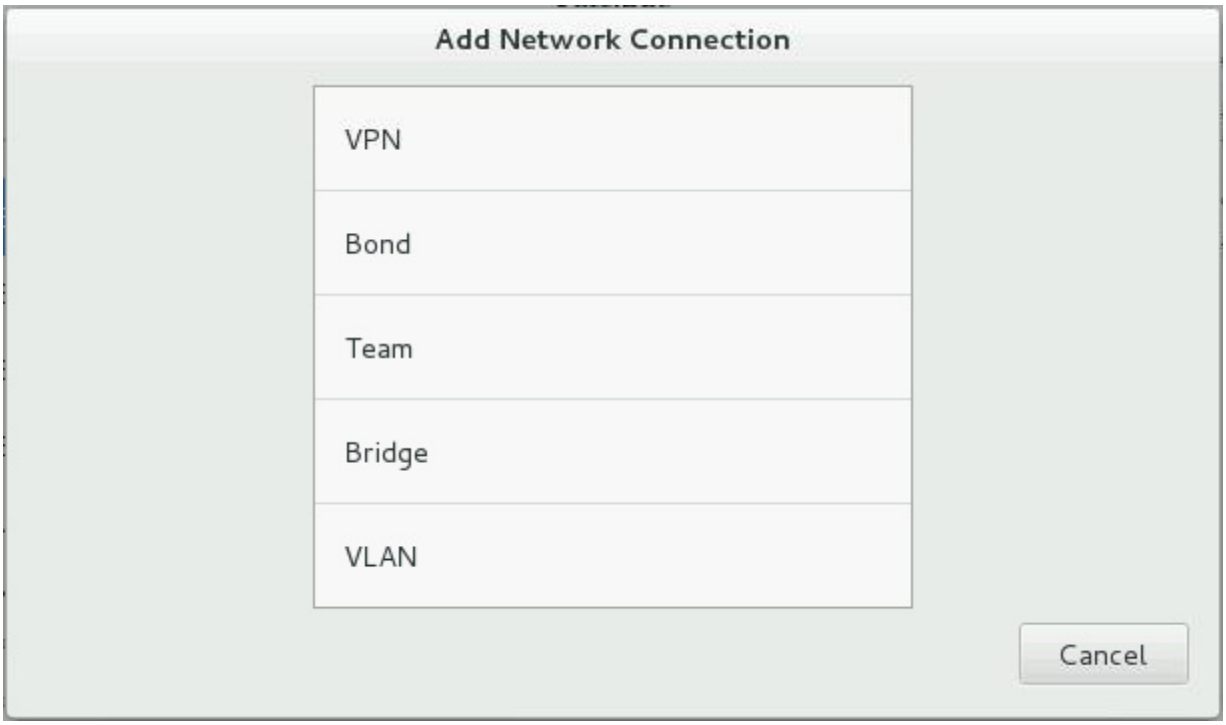
b. Click the “Network Settings” option from the drop-down menu.

- The “Network Settings Editor” appears.



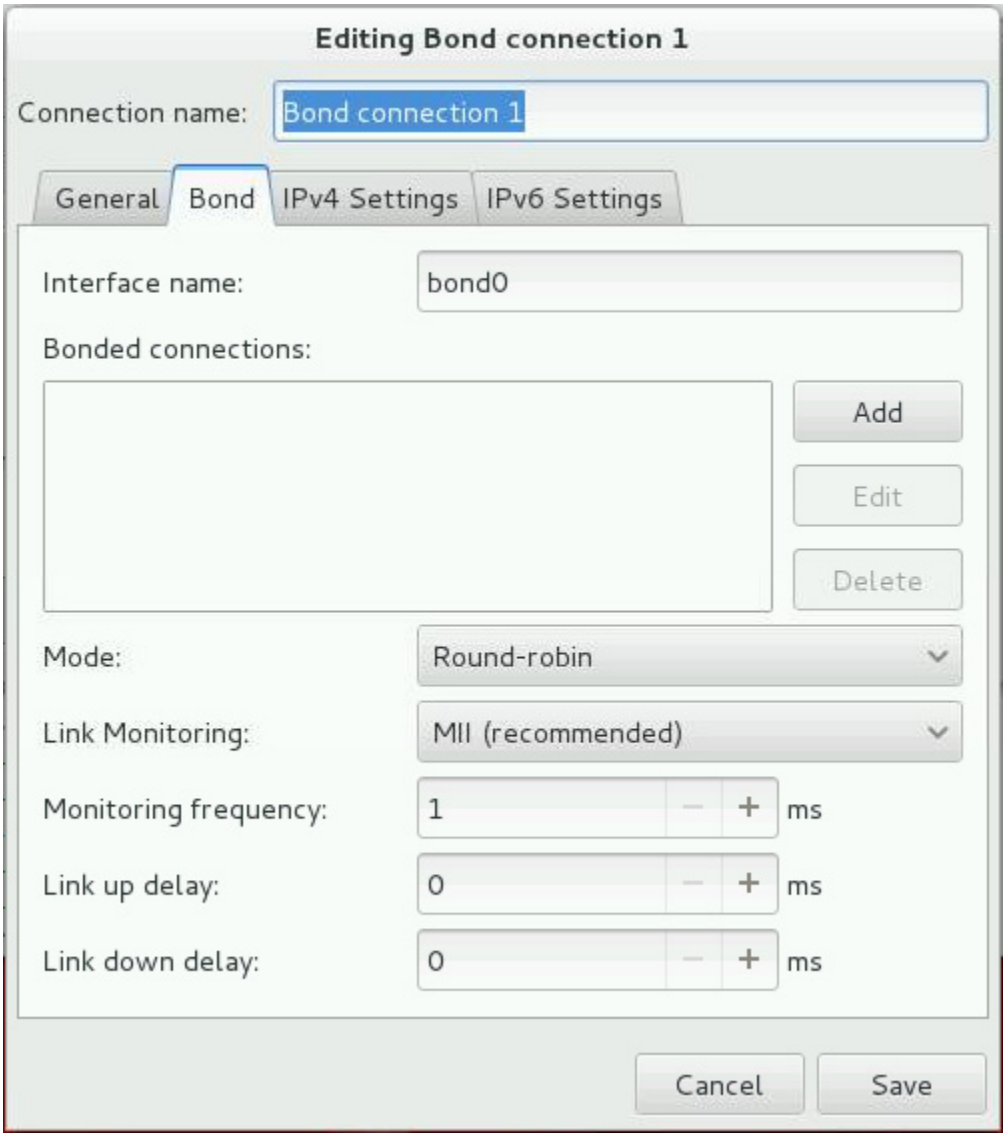
c. Click the "+" button to add a new connection type.

- The "Add Network Connection" window appears.



d. Click "Bond" to add a bonded interface.

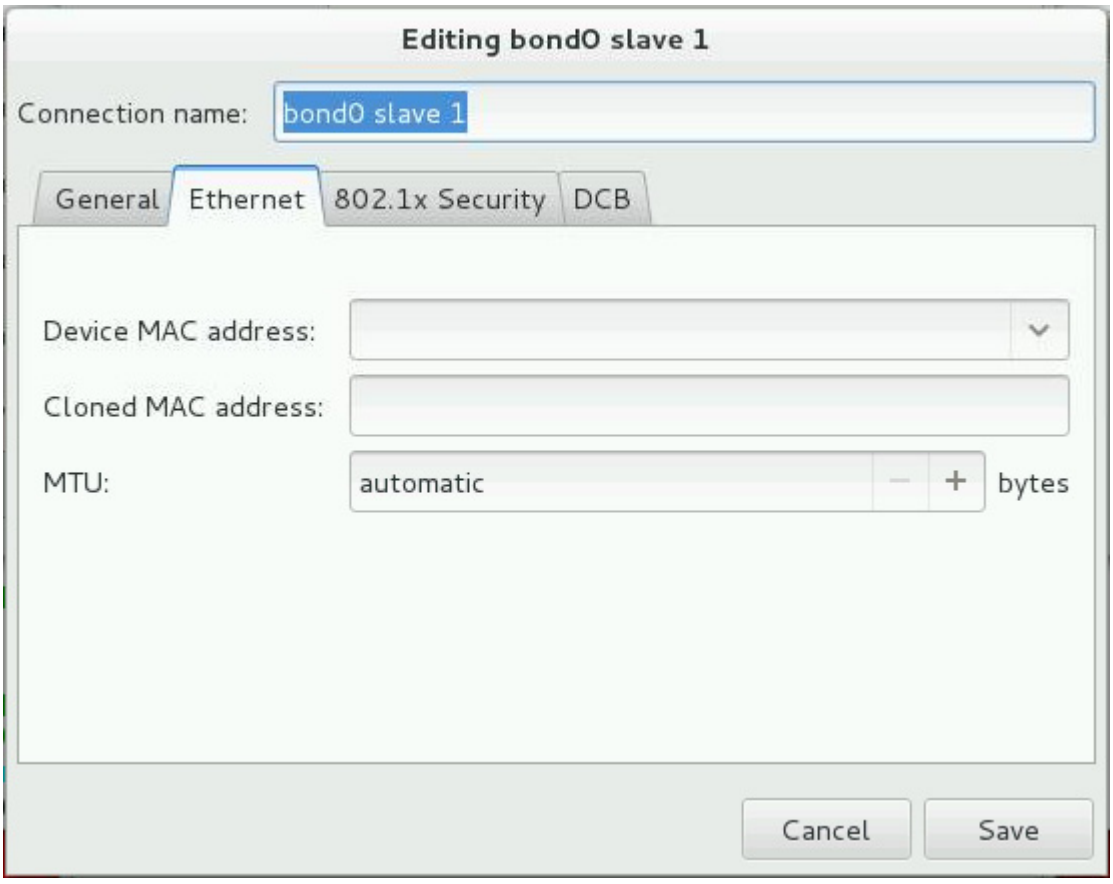
- The following window appears.
- The default Connection name is "Bond connection 1."
- The default interface name is `bond0`.



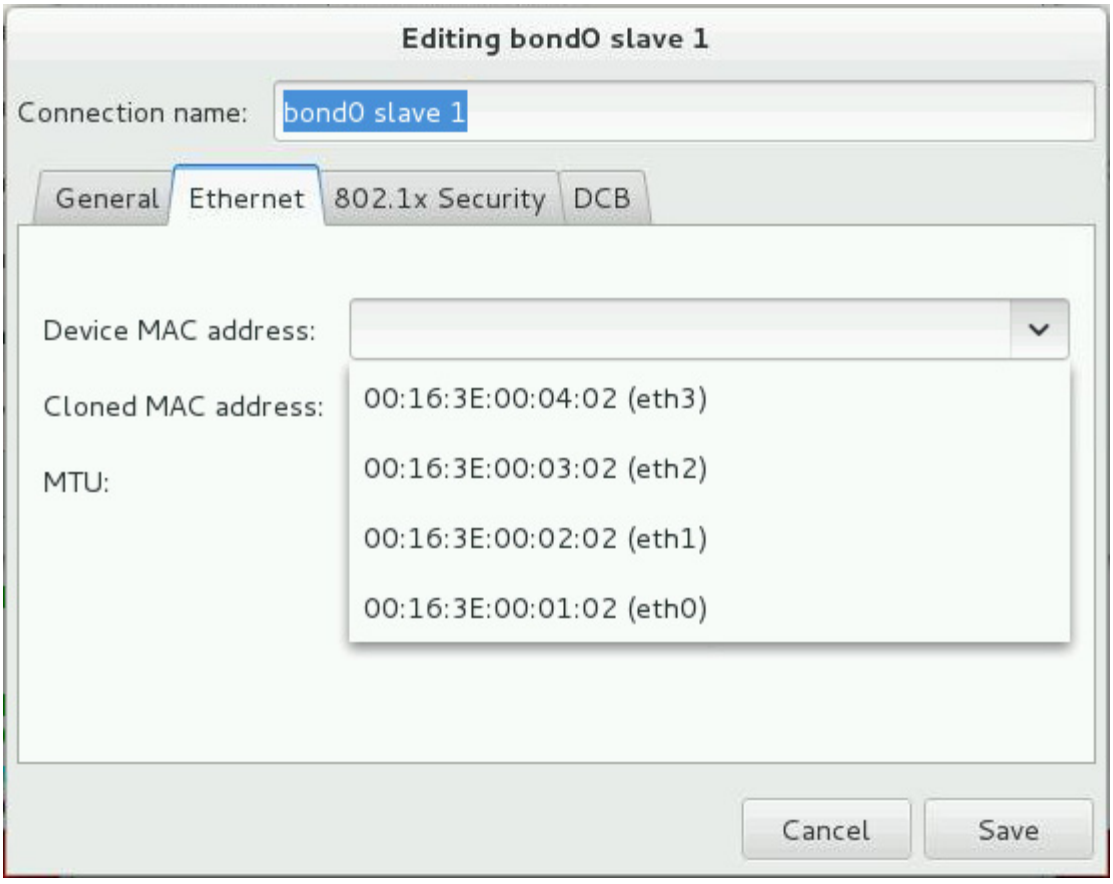
e. Click the “Add” button to add the slave interfaces to the bond.

- The following window appears.

f. Accept the default “Ethernet” selection. Click “Create” to display the following window.



g. Click the down arrow on the “Device MAC address” prompt to display the available Ethernet devices.

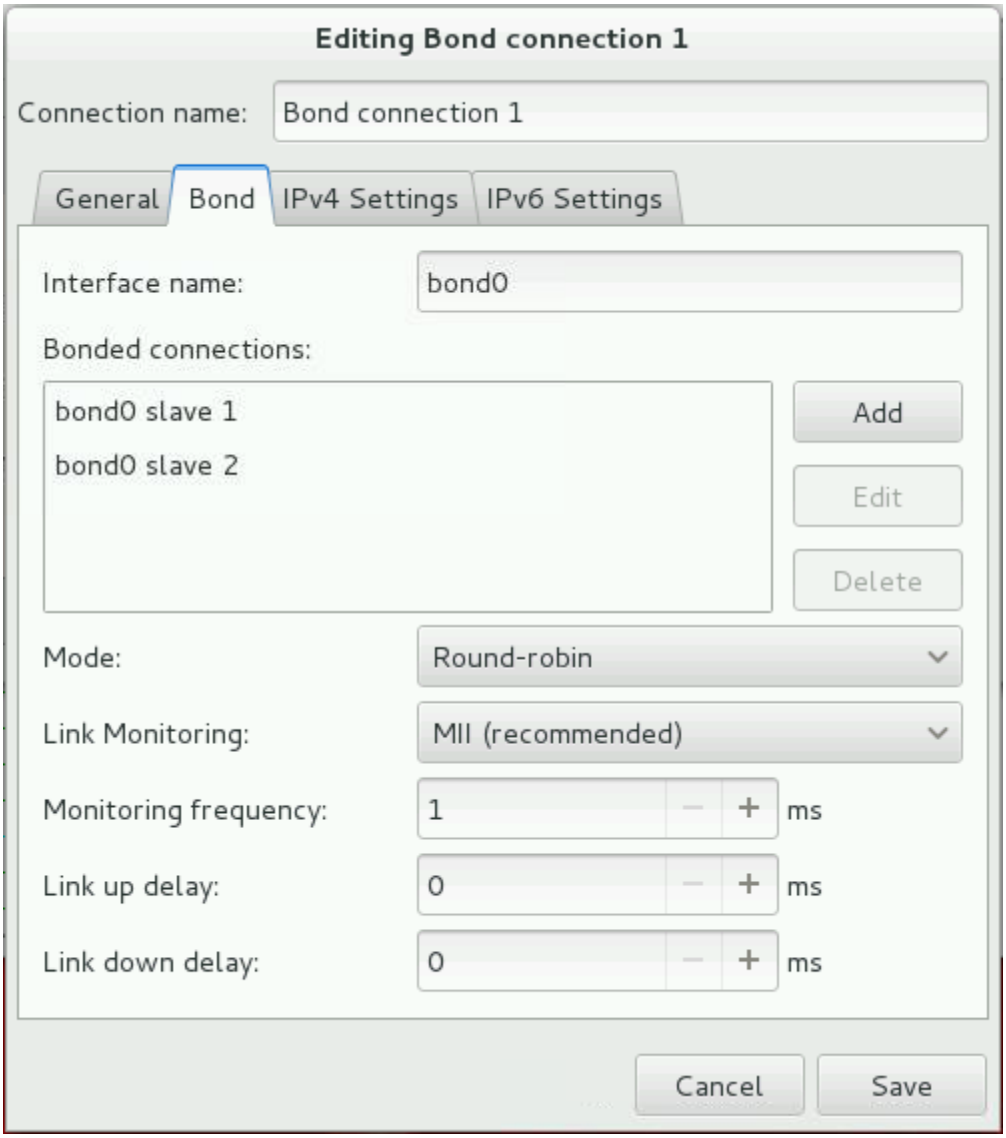


h. Select the `eth3` device from the drop-down list.

- Click the “Save” button.

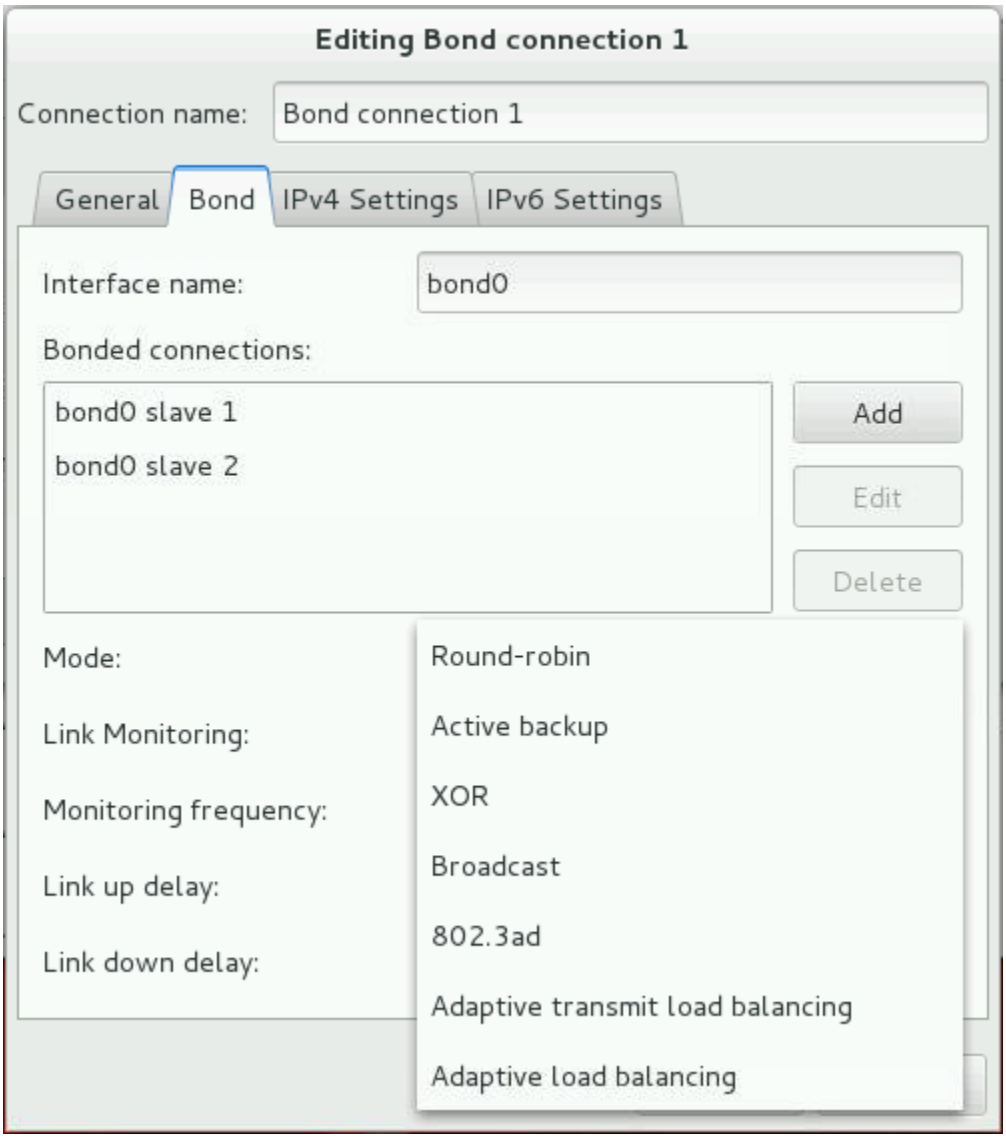
i. Repeat steps 5e (click “Add”), 5f (click “Create”), and 5g (click the down arrow) and add the `eth2` slave interface. Click “Save” and the following window appears.

- Note that two “Bonded connections” have been added.



j. Click the down arrow on the “Mode” prompt to display the available modes.

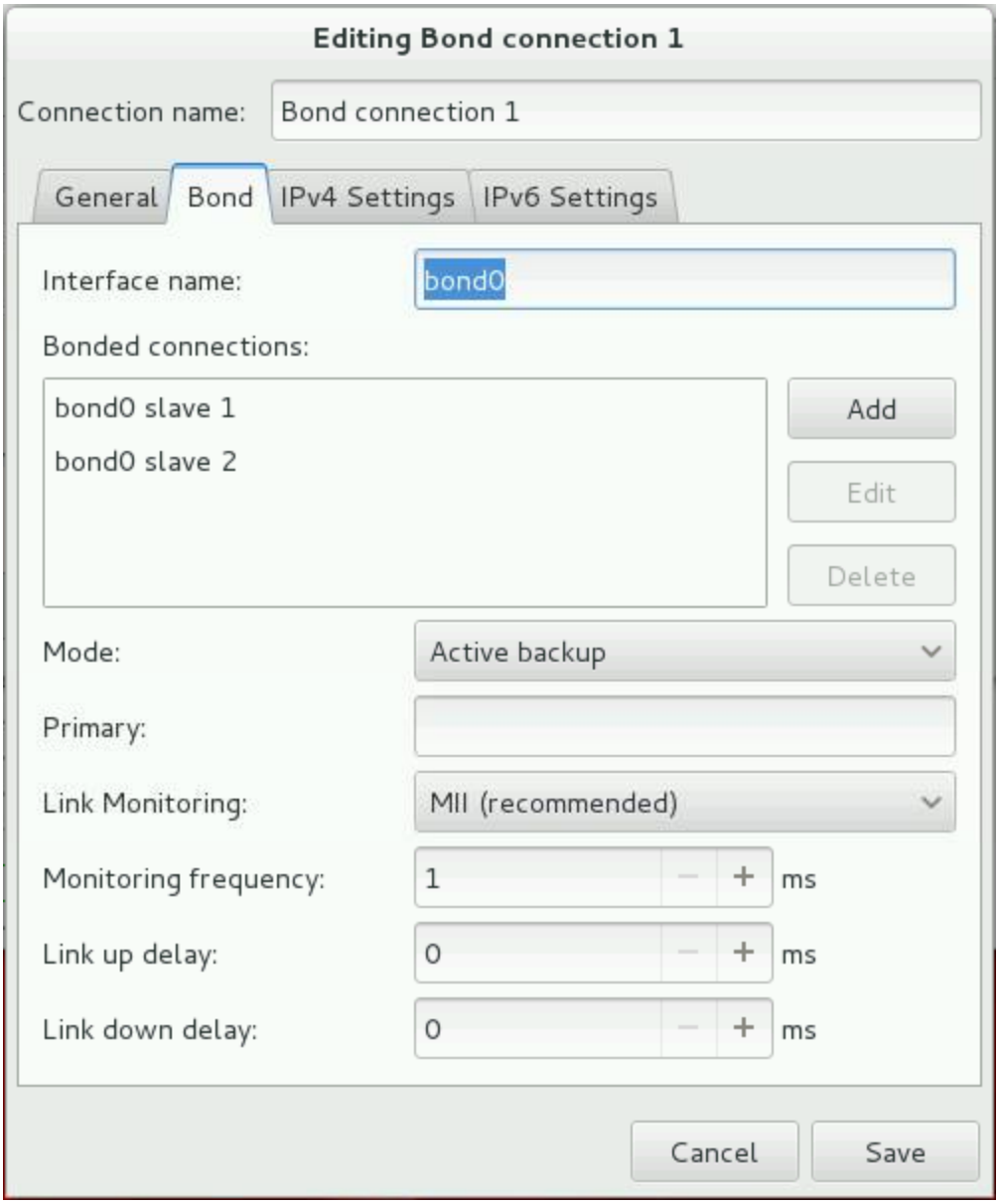
- The list of modes appears.



k. Select the “Active backup” option from the drop-down list.

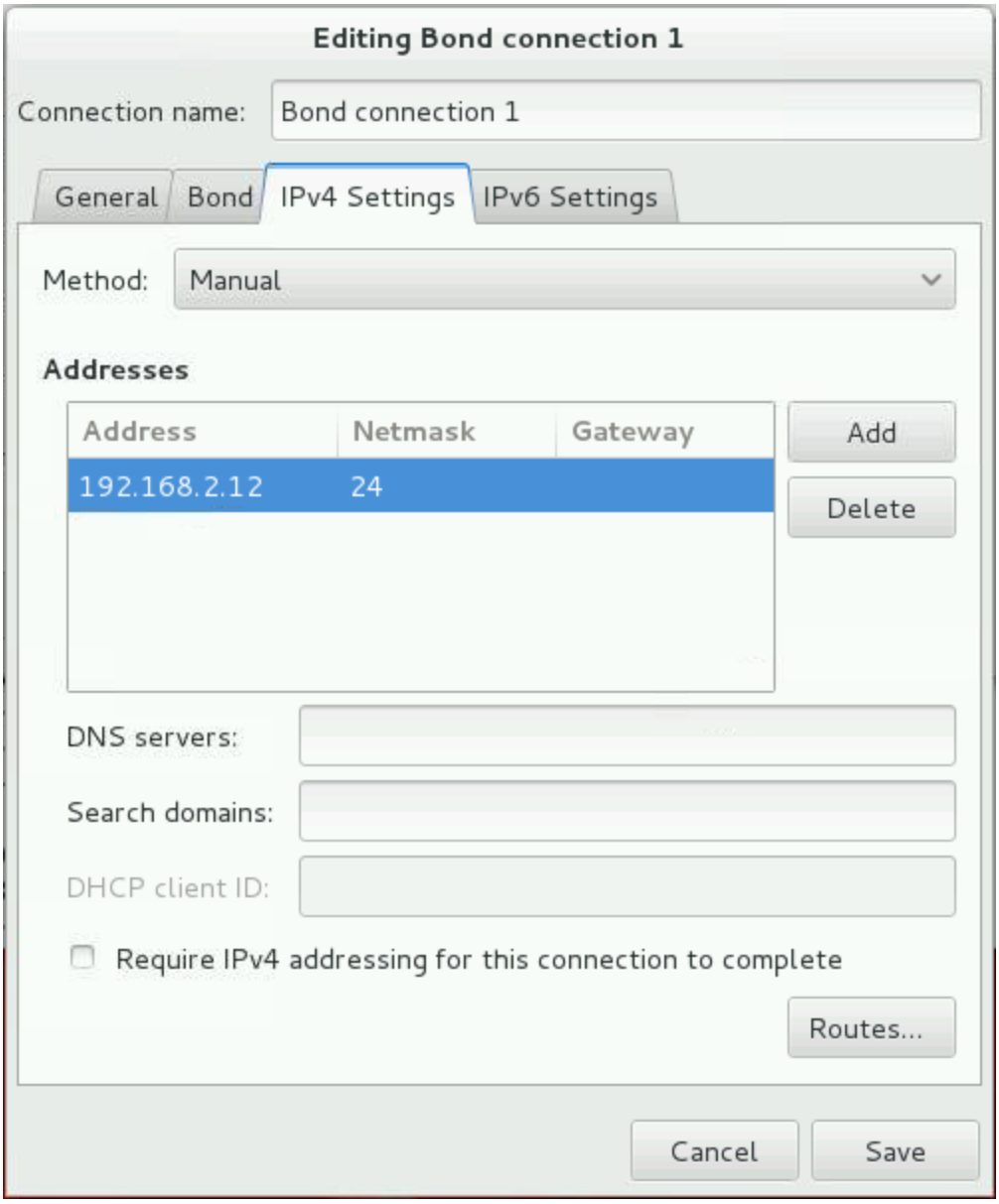
- The following window appears.
- Note that a “Primary” prompt appears when “Active backup” is the selected Mode.
- You can designate an interface as “Primary” to make it the active slave when it is available.

- Do not specify a “Primary” interface for this exercise.



l. Click the “IPv4 Settings” tab to assign an IPv4 address to the bonded interface.
   - The following window appears.
   - Change the “Method” to “Manual.”
   - Click “Add” to add the following Address information:
     - Address: `192.168.2.12`
     - Netmask: `24`

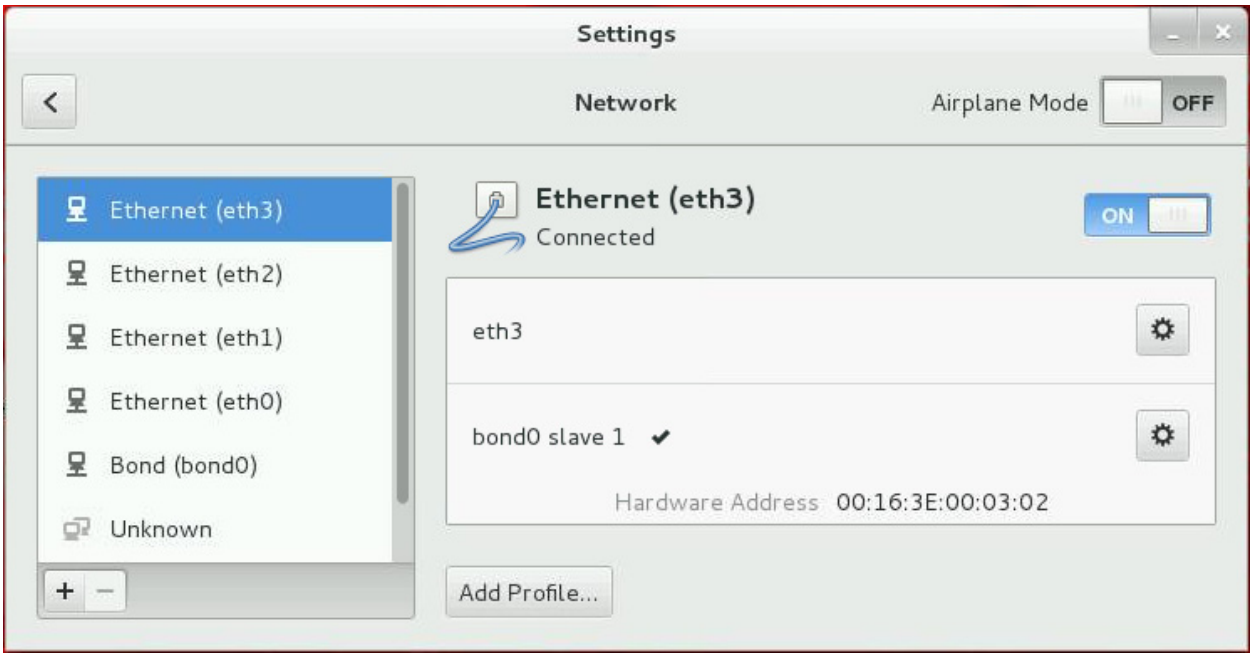
- Gateway: `<empty>`



m. Click “Save” to complete configuring network bonding.

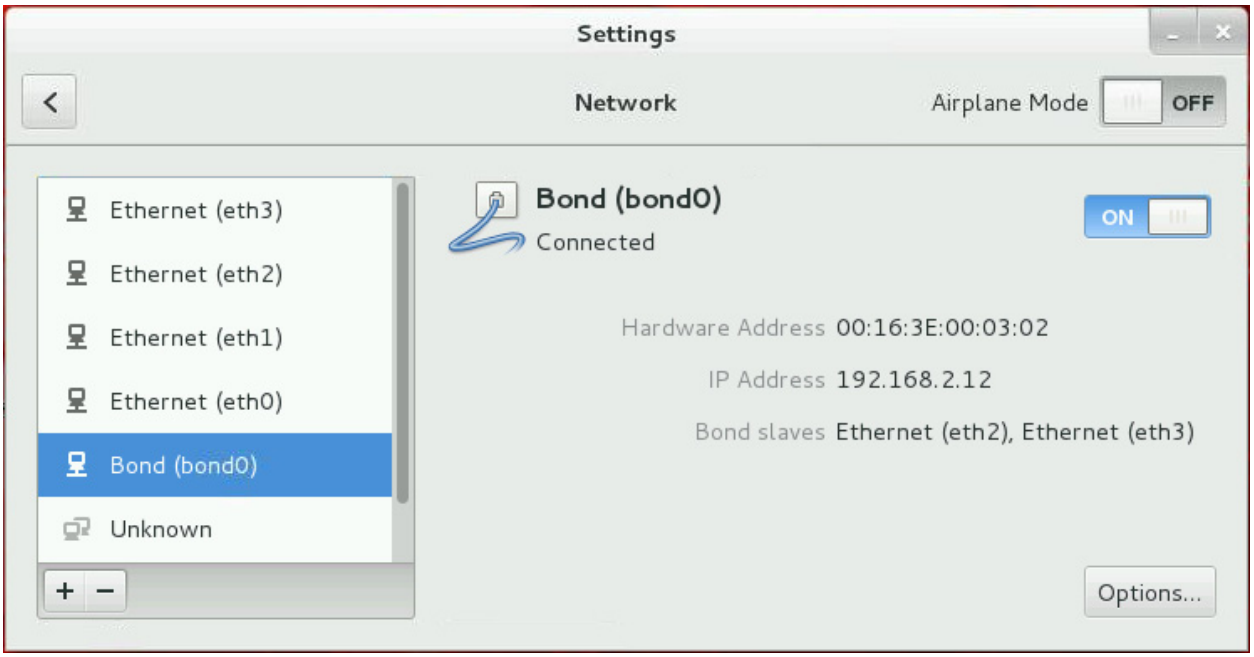
- You need to click in the “Gateway” field before “Save” becomes selectable.

- The “Bond (bond0)” interface now appears in the “Network Settings” window.



n. Select the “Bond (bond0)” option to display the following window.

- Note that the Hardware Address, IP Address, and Bond slaves are shown.



o. Click the “`X`” in the top-right corner to close the window.

5. View the network interfaces on **host02**.

a. Use the `ip addr` command to view the network interfaces.

- Note that the new `bond0` interface is listed and includes “`MASTER`” and “`state UNKNOWN`”.
- Note that `eth2` and `eth3` now include “`SLAVE`” and “`master bond0`”.
- Note that `eth2`, `eth3`, and `bond0` all have the same MAC address.

```
# ip addr
...
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 ...
master bond0 state UP ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 ...
master bond0 state UP ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 ...
state UNKNOWN
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.12/24 brd 192.168.2.255 scope global bond0
...
```

b. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.
   - Note that there is a network configuration file for the bonded interface, `ifcfg-Bond_connection_1`.
   - Note that there are network configuration files for the two slave interfaces, `ifcfg-bond0_slave_1` and `ifcfg-bond0_slave_2`.

```
# ls /etc/sysconfig/network-scripts
ifcfg-bond0_slave_1  ...
ifcfg-bond0_slave_2  ...
ifcfg-Bond_connection_1  ...
ifcfg-eth0  ...
ifcfg-eth1  ...
...
```

c. Use the `cat` command to view the contents of the `ifcfg-Bond_connection_1` file.
   - Note that the "BONDING_OPTS" setting has "mode=active-backup."
   - Note that the "BONDING_OPTS" setting also sets the Link Monitoring method to MII by default.
   - The Link monitoring frequency is 1 millisecond, and Link up delay and Link down delay are set to 0 by default.

```
# cat /etc/sysconfig/network-scripts/ifcfg-Bond_connection_1
DEVICE=bond0
BONDING_OPTS="miimon=1 updelay=0 downdelay=0 mode=active-backup"
TYPE=Bond
BONDING_MASTER=yes
BOOTPROTO=none
IPADDR=192.168.2.12
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
```

```
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
IPV6_PRIVACY=no
NAME=”Bond connection 1”
UUID=...
ONBOOT=yes
```

d. Use the `cat` command to view the contents of the `ifcfg-bond0_slave_1` file.

- Note that “MASTER” is set to the UUID value in the `ifcfg-Bond_connection_1` file.

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0_slave_1
HWADDR=00:16:3E:00:04:02
TYPE=Ethernet
NAME=”bond0 slave 1”
UUID=...
ONBOOT=yes
MASTER=...
SLAVE=yes
```

e. Use the `cat` command to view the contents of the `ifcfg-bond0_slave_2` file.

- Note that “MASTER” is set to the UUID value in the `ifcfg-Bond_connection_1` file.

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0_slave_2
HWADDR=00:16:3E:00:03:02
TYPE=Ethernet
NAME=”bond0 slave 2”
UUID=...
ONBOOT=yes
MASTER=...
SLAVE=yes
```

f. Use the `nmcli con` command to view the network connections.

- Note that the bond and slave connections are now shown.

```
# nmcli con
NAME                UUID    TYPE              DEVICE
eth1                ...     802-3-ethernet    eth1
eth2                ...     802-3-ethernet    --
eth3                ...     802-3-ethernet    --
Bond connection 1   ...     bond              bond0
bond0 slave 2       ...     802-3-ethernet    eth2
bond0 slave 1       ...     802-3-ethernet    eth3
```

```
eth0                ...     802-3-ethernet    eth0
```

g. Use the `nmcli` utility to bring up the “Bond connection 1” connection.

```
# nmcli con up “Bond connection 1”
Connection successfully activated (D-BUS active path:...)
```

h. Use the `ip addr` command to view the network interfaces.

- Note that the `bond0` interface is now “UP”.

```
# ip addr
...
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 ...
state UP
...
```

- Do not log off **host02**. You use it again in subsequent practices.

i. If necessary, open a new terminal window on **dom0**.

- Use the `su -` command to become the `root` user in this new terminal window.
- The `root` password is `oracle`.

```
$ su –
Password: oracle
```

Practice 10-2: Configuring Network Bonding from the Command Line

Overview

In this practice, you:

- Log in to **host01** by using `ssh`
- View the network configuration on **host01**
- Configure network bonding on **host01** by using the command line

Assumptions

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

Tasks

1. From **dom0**, use the `ssh` command to connect to **host01**.
   - The `root` password is `oracle`.

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

2. View the network interfaces on **host01**.
   a. Use the `ip addr` command to view the network interfaces.
      - The IP address for `eth1` was obtained by using DHCP. Yours might be different than the example shown.
      - Note that the `eth2` and `eth3` interfaces do not have IP addresses.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue ...
    link/loopback 00:00:00:00:00:00: brd 00:00:00:00:00:00
    inet addr:127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.101/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.168.1.255 scope global eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:03:01 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:04:01 brd ff:ff:ff:ff:ff:ff
```

b. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

```
# ls /etc/sysconfig/network-scripts
ifcfg-eth0  ...
ifcfg-eth1  ...
ifcfg-eth2  ...
ifcfg-eth3  ...
...
```

c. Use the `nmcli con` command to view the network connections.

- Note that the `eth2` and `eth3` connections are not associated with a device.

```
# nmcli con
NAME                UUID    TYPE              DEVICE
eth1                ...     802-3-ethernet    eth1
eth2                ...     802-3-ethernet    --
eth3                ...     802-3-ethernet    --
eth0                ...     802-3-ethernet    eth0
```

3. Use the `nmcli` utility to configure network bonding.

a. Use the `nmcli con add` command to add a "bond" connection type.

- Use the "`type bond`" argument to specify a bonded interface.
- Use the "`con-name bond0`" argument to specify the name of the new bond connection.
- Use the "`ifname bond0`" argument to specify the interface to bind the connection to.
- Use the "`mode active-backup`" argument to specify the bonding mode.
- Use the "`ip4 192.168.2.11/24`" argument to specify IPv4 address to assign to the interface.

```
# nmcli con add type bond con-name bond0 ifname bond0 mode
active-backup ip4 192.168.2.11/24
Connection 'bond0' (...) successfully added.
```

b. Use the `nmcli con add` command to add `eth2` as a "bond-slave" connection type.

- The bond-slave interface is `eth2`.
- The bond master is `bond0`.

```
# nmcli con add type bond-slave ifname eth2 master bond0
Connection 'bond-slave-eth2' (...) successfully added.
```

c. Use the `nmcli con add` command to add `eth3` as a "bond-slave" connection type.

- The bond-slave interface is `eth3`.
- The bond master is `bond0`.

```
# nmcli con add type bond-slave ifname eth3 master bond0
Connection 'bond-slave-eth3' (...) successfully added.
```

d. Use the `nmcli con` command to view the network connections.
   - Note that a new `bond-slave-eth2` connection exists for device `eth2`.
   - Note that a new `bond-slave-eth3` connection exists for device `eth3`.
   - Note that a new `bond0` connection exists which is a type bond.

```
# nmcli con
NAME               UUID    TYPE              DEVICE
eth1               ...     802-3-ethernet    eth1
eth2               ...     802-3-ethernet    --
eth3               ...     802-3-ethernet    --
bond-slave-eth3    ...     802-3-ethernet    eth3
bond-slave-eth2    ...     802-3-ethernet    eth2
bond0              ...     bond              bond0
eth0               ...     802-3-ethernet    eth0
```

e. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.
   - Note that a new `ifcfg-bond0` file exists.
   - Note that a new `ifcfg-bond-slave-eth2` file exists.
   - Note that a new `ifcfg-bond-slave-eth3` file exists.

```
# ls /etc/sysconfig/network-scripts
ifcfg-bond0           ...
ifcfg-bond-slave-eth2 ...
ifcfg-bond-slave-eth3 ...
...
```

f. Use the `cat` command to view the contents of the `ifcfg-bond0` file.

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
BONDING_OPTS=mode=active-backup
TYPE=Bond
BONDING_MASTER=yes
BOOTPROTO=none
IPADDR=192.168.2.11
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=bond0
UUID=...
```

```
ONBOOT=yes
```

g. Use the `cat` command to view the contents of the `ifcfg-bond-slave-eth2` file.

- Note that "MASTER" is set to `bond0`.

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond-slave-eth2
TYPE=Ethernet
NAME=bond-slave-eth2
UUID=...
DEVICE=eth2
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

h. Use the `cat` command to view the contents of the `ifcfg-bond-slave-eth3` file.

- Note that "MASTER" is set to `bond0`.

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond-slave-eth3
TYPE=Ethernet
NAME=bond-slave-eth3
UUID=...
DEVICE=eth3
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

i. Use the `ip addr` command to view the network interfaces.

- Note that the `eth2` interface now includes "`SLAVE`" and "`master bond0`".
- Note that the `eth3` interface now includes "`SLAVE`" and "`master bond0`".
- Note that the new `bond0` interface is listed and includes "`MASTER`" and "`state UNKNOWN`"

```
# ip addr
...
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 ...
master bond0 state UP ...
    link/ether 00:16:3e:00:03:01 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 ...
master bond0 state UP ...
    link/ether 00:16:3e:00:03:01 brd ff:ff:ff:ff:ff:ff
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 ...
state UNKNOWN
    link/ether 00:16:3e:00:03:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.11/24 brd 192.168.2.255 scope global bond0
...
```

j. Use the `nmcli` utility to bring up `bond0`.

```
# nmcli con up bond0
Connection successfully activated (D-BUS active path:...)
```

k. Use the `ip addr` command to ensure that the `bond0` interface is UP.

```
# ip addr
...
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 ...
state UP
...
```

- Do not log off **host01**. You use it again in subsequent practices.

Practice 10-3: Working with Bonded Interfaces

Overview

In this practice, you:

- Test connectivity between the bonded interfaces on **host01** and **host02**
- Explore the `/sys/class/net/bond0` directory
- Change the MII monitoring frequency on **host01**
- Test for slave failover on **host01**
- Remove bond and slave connections on **host02** by using the GUI
- Remove bond and slave connections on **host01** by using the command line

Assumptions

- You are the `root` user on **host01** and **host02**.
- The bonded interface on **host01** has an IP address of `192.168.2.11`.
- The bonded interface on **host02** has an IP address of `192.168.2.12`.

Tasks

1. Test connectivity between the bonded interfaces on **host01** and **host02**.
   a. From **host02**, use the `ping` command to communicate to the bonded interface on **host01**.
      - The IP address of the bonded interface on **host01** is `192.168.2.11`.
      - Press CTRL-C to exit after a few lines of output.

```
[host02]# ping 192.168.2.11
PING 192.168.2.11 (192.168.2.11) 56(84) bytes of data.
64 bytes from 192.168.2.11: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.11: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.11: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.11: icmp_seq=1 ttl=64 time=...
^C
...
```

   b. From **host02**, use the `netstat -r` command to view the route table.
      - Note that the route to `192.168.2.0` is through the `bond0` interface.

```
[host02]# netstat -r
Kernel IP routing table
Destination    Gateway ...  Iface
Default        ...     ...  eth0
192.0.2.0      ...     ...  eth0
192.168.1.0    ...     ...  eth1
192.168.2.0    ...     ...  bond0
```

   c. From **host01**, use the `ping` command to communicate to the bonded interface on **host02**.
      - The IP address of the bonded interface on **host02** is `192.168.2.12`.

- Press CTRL-C to exit after a few lines of output.

```
[host01]# ping 192.168.2.12
PING 192.168.2.12 (192.168.2.12) 56(84) bytes of data.
64 bytes from 192.168.2.12: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.12: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.12: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.12: icmp_seq=1 ttl=64 time=...
^C
...
```

d. From **host01**, use the `netstat -r` command to view the route table.

- Note that the route to `192.168.2.0` is through the `bond0` interface.
- If the `netstat` command is not found, use the `yum` command to install the `net-tools` package. Answer `y` to "Is this ok."

```
[host01]# netstat -r
-bash: netstat: command not found
[host01]# yum install net-tools
...
Is this ok [y/d/N]: y
...
Complete!
[host01]# netstat -r
Kernel IP routing table
Destination     ...  Iface
...
192.168.2.0     ...  bond0
```

2. View the contents of `/sys/class/net/bond0/`.

- Each network interface contains a directory in `/sys/class/net`.

a. From **host01**, use the `cd` command to change to the `/sys/class/net` directory.

- Use the `ls` command to display the contents of the directory.
- Note that `bonding_masters` is a regular file.

```
[host01]# cd /sys/class/net
[host01]# ls
bond0  bonding_masters  eth0  eth1  eth2  eth3  lo
```

b. Use the `cat` command to view the `bonding_masters` file.

```
[host01]# cat bonding_masters
bond0
```

c. Use the `cd` command to change to the `bond0` directory.

- Use the `ls` command to display the contents of the directory.

```
[host01]# cd bond0
[host01]# ls
addr_assign_type  carrier  ifalias    netdev_group  slave_eth3
address           dev_id   ifindex    operstate     speed
addr_len          dormant  iflink     power         statistics
bonding           duplex   link_mode  queues        subsystem
broadcast         flags    mtu        slave_eth2    tx_queue_len
```

d. Use the `cat` command to view the `operstate` file.

```
[host01]# cat operstate
up
```

e. Use the `cat` command to view the `address` file.

```
[host01]# cat address
00:16:3e:00:03:01
```

f. Use the `cat` command to view the `uevent` file.

```
[host01]# cat uevent
INTERFACE=BOND0
IFINDEX=6
```

g. Use the `cd` command to change to the `bonding` directory.

- Use the `ls` command to display the contents of the directory.

```
[host01]# cd bonding
[host01]# ls
active_slave    all_slaves_active  miimon     primary_reselect
ad_actor_key    arp_interval       mii_status    queue_id
ad_aggregator   arp_ip_target      min_links     resend_igmp
ad_num_ports    arp_validate       mode          slaves
ad_partner_key  downdelay          num_grat_arp  updelay
ad_partner_mac  fail_over_mac      num_unsol_na  use_carrier
ad_select       lacp_rate          primary      xmit_hash_policy
```

h. Use the `cat` command to view the `active_slave` file.

```
[host01]# cat active_slave
eth2
```

i. Use the `cat` command to view the `mode` file.

```
[host01]# cat mode
active-backup 1
```

j. Use the `cat` command to view the `slaves` file.

```
[host01]# cat slaves
eth2 eth3
```

k. Use the `cat` command to view the `miimon` file.

- This specifies the MII link monitoring frequency in milliseconds.

```
[host01]# cat miimon
100
```

l. Use the `cat` command to view the `mii_status` file.

```
[host01]# cat mii_status
up
```

3. Change the MII monitoring frequency.

a. On **host01**, use the `vi` command to edit the `/etc/sysconfig/network-scripts/ifcfg-bond0` file.

- Change the BONDING_OPTS setting as follows to set `miimon` to `120`.

```
[host01]# vi /etc/sysconfig/network-scripts/ifcfg-bond0
...
BONDING_OPTS=mode=active-backup                    (old value)
BONDING_OPTS="mode=active-backup miimon=120"       (new value)
```

b. Use the `nmcli` command to reload all connection files from disk.

- NetworkManager does not monitor changes to connection files by default.
- You need to use this command to tell NetworkManager to reread the connection profiles from disk whenever making a change.

```
[host01]# nmcli con reload
```

c. Use the `nmcli` command to bring down the `bond0` connection.

- Stopping the master interface also stops the slave interfaces.

```
[host01]# nmcli con down bond0
Connection 'bond0' successfully deactivated (d-Bus active ...)
```

d. Use the `nmcli` command to bring up the `bond-slave-eth2` connection.

```
[host01]# nmcli con up bond-slave-eth2
Connection successfully activated (d-Bus active ...)
```

e. Use the `nmcli` command to bring up the `bond-slave-eth3` connection.

```
[host01]# nmcli con up bond-slave-eth3
Connection successfully activated (d-Bus active ...)
```

f. Use the `nmcli` command to bring up the `bond0` connection.

```
[host01]# nmcli con up bond0
Connection successfully activated (d-Bus active ...)
```

g. Use the `cat` command to view the `miimon` file.

- Note the value is now `120`, instead of `100`.

```
[host01]# cat /sys/class/net/bond0/bonding/miimon
120
```

4. Test for slave failover.

- The active slave in "Active backup" mode is stored in the `active_slave` file.
- You can also determine the active slave by viewing the `/proc/net/bonding/bond0` file.

a. On **host01**, use the `cat` command to view the `active_slave` file, which is located in the `/sys/class/net/bond0/bonding` directory.

- The active slave in this example is `eth2`.

```
[host01]# cat /sys/class/net/bond0/bonding/active_slave
eth2
```

b. Use the `cd` command to change to the `/proc/net/bonding` directory.

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

```
[host01]# cd /proc/net/bonding
[host01]# ls
bond0
```

c. Use the `cat` command to view the contents of the `bond0` file.

- Note that "Currently Active Slave" is `eth2`.

```
[host01]# cat bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth2
MII Status: up
MII Polling Interval (ms): 120
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth2
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 00:16:3e:00:03:01
Slave queue ID: 0

Slave Interface: eth3
MII Status: up
```

```
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 00:16:3e:00:03:01
Slave queue ID: 0
```

d. Use the `ip link` command to bring down `eth2`.

```
[host01]# ip link set dev eth2 down
```

e. Use the `ip link` command to view the `eth2` link.

- Note that the "state" is `DOWN` for `eth2`.

```
[host01]# ip link show eth2
4: eth2: <BROADCAST,MULTICAST,SLAVE. mtu 1500 ... state DOWN ...
...
```

f. Use the `cat` command to view the `/var/log/messages` file.

- Note the "bonding" messages; `eth2` is disabled and `eth3` is active.

```
[host01]# cat /var/log/messages
<date_time> host01 NetworkManager[9730]: <info> (eth2): link
disconnected (deferring action for 4 seconds)
<date_time> host01 kernel: bonding: bond0: link status
definitely down for interface eth2, disabling it
<date_time> host01 kernel: bonding: bond0: making interface eth3
the new active one.
<date_time> host01 NetworkManager[9730]: <info> (eth2): link
disconnected (calling deferred action)
```

g. Use the `cat` command to view the contents of the `bond0` file.

- Note that now the "Currently Active Slave" is `eth3`.
- Also note that `eth2` is "down."

```
[host01]# cat bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth3
MII Status: up
MII Polling Interval (ms): 120
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth2
MII Status: down
Speed: Unknown
Duplex: Unknown
```

```
Link Failure Count: 0
Permanent HW addr: 00:16:3e:00:03:01
Slave queue ID: 0
...
```

h. Use the `cat` command to view the `/sys/class/net/bond0/bonding/active_slave` file.

- This file also indicates that `eth3` is the active slave.

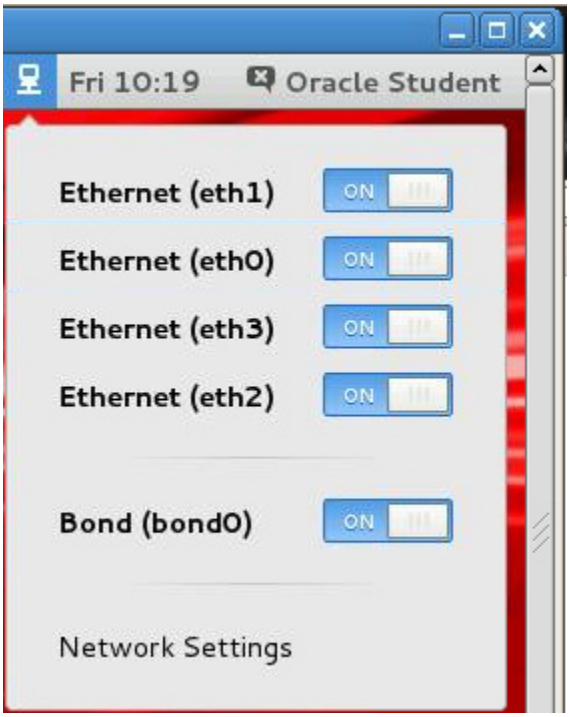
```
[host01]# cat /sys/class/net/bond0/bonding/active_slave
eth3
```

5. Remove bond and slave connections on **host02**.

- Begin by using the Network Settings GUI.
- Complete removal of the connections by using the command line.

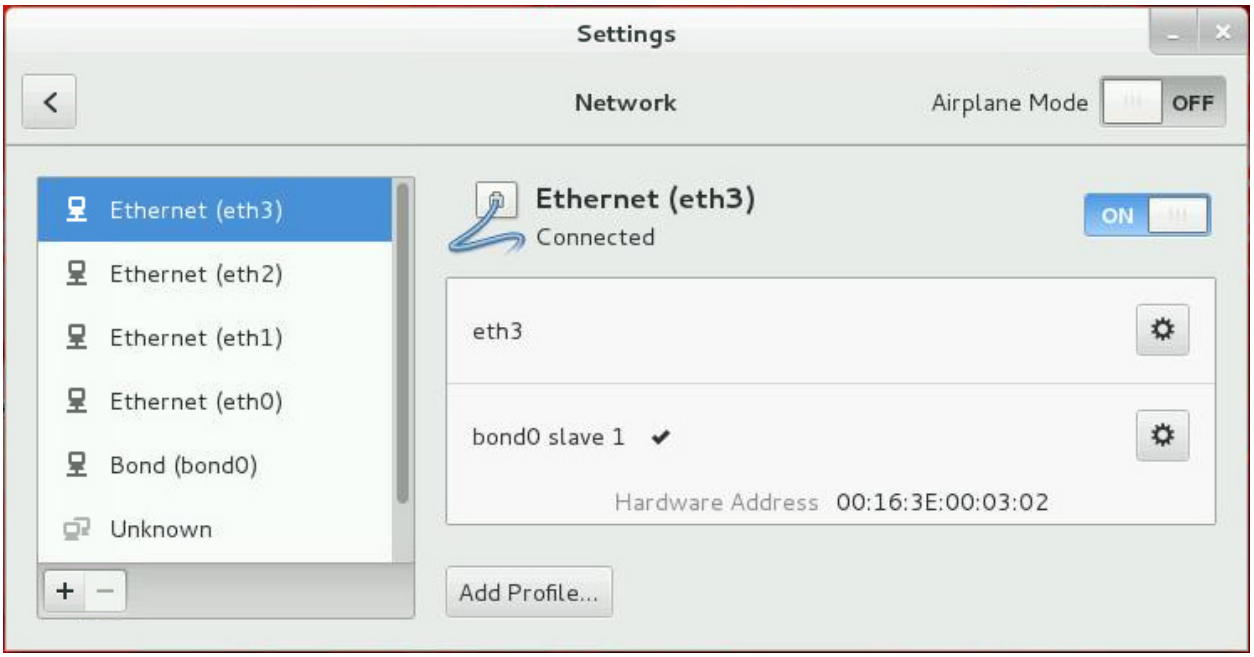
a. On **host02**, click the network icon from the GNOME desktop notification area.

- The drop-down menu includes “Ethernet” interfaces, the “Bond (bond0)” interface, and the “Network Settings” option.



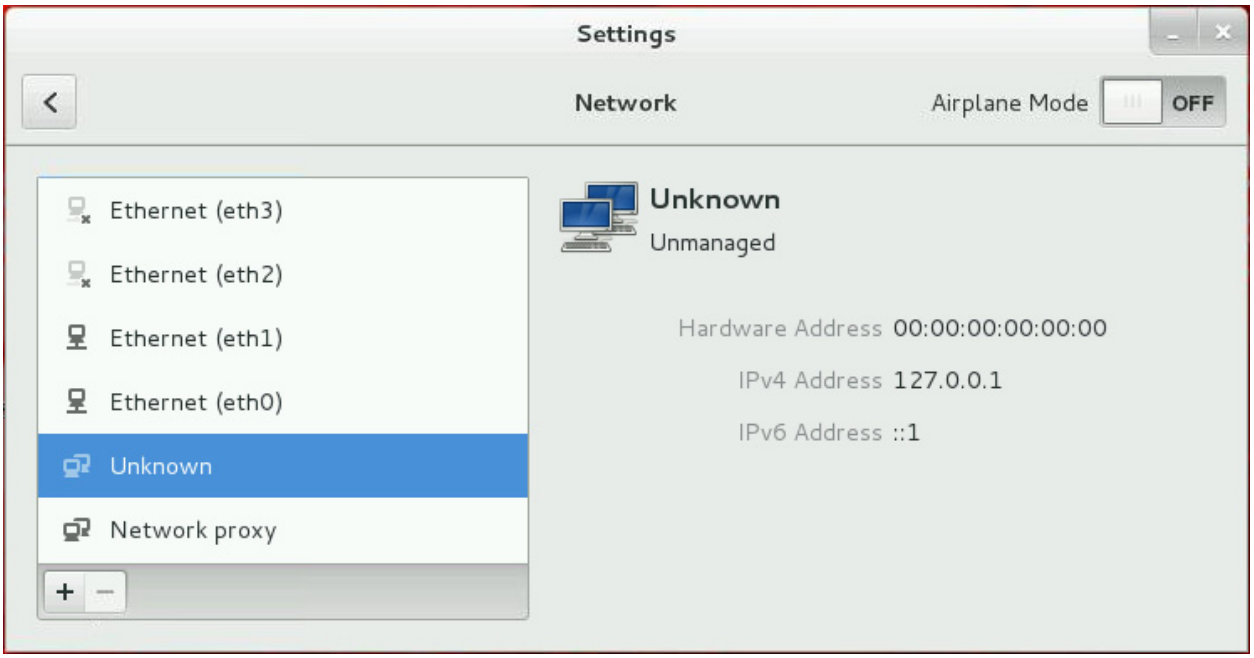
b. Click the “Network Settings” option from the menu.

- The “Network Settings Editor” appears.



c. Select the “Bond (bond0)” interface and click the “-” button to remove the connection.

- The window is shown as follows after clicking the “-” button.



d. Click the “X” in the top-right corner to close the window.

e. Use the `ls` command to view the contents of the `/sys/class/net` directory.

- Note that the `bond0` directory no longer exists.

```
[host02]# ls /sys/class/net
bonding_masters  eth0  eth1  eth2  eth3  lo
```

f. Use the `cat` command to view the `/sys/class/net/bonding_masters` file.

- Note that the file is empty.

```
[host02]# cat /sys/class/net/bonding_masters
```

g. Use the `ls` command to view the contents of the `/proc/net/bonding` directory.

- Note that the directory is empty.

```
[host02]# ls /proc/net/bonding
```

h. Use the `nmcli con` command to view the network connections.

- Note that the bond connection no longer exists.
- Note that the slave connections still exist but are no longer associated with a device.

```
[host02]# nmcli con
NAME               UUID    TYPE              DEVICE
eth1               ...     802-3-ethernet    eth1
eth2               ...     802-3-ethernet    --
eth3               ...     802-3-ethernet    --
bond0 slave 1      ...     802-3-ethernet    --
bond0 slave 2      ...     802-3-ethernet    --
eth0               ...     802-3-ethernet    eth0
```

i. Use the `ip link` command to view the links.

- Note that the `bond0` entry no longer exists.
- Note that the `eth2` and `eth3` entries no longer include "`SLAVE`" or "`master bond0`" in their description.
- Note that the `eth2` and `eth3` entries have their original MAC addresses.

```
[host02]# ip link
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
    link/ether 00:16:3e:00:04:02 brd ff:ff:ff:ff:ff:ff
```

j. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

- Note that the network configuration file for the bonded interface no longer exists.
- Note that the network configuration files for the slaves still exist.

```
[host02]# ls /etc/sysconfig/network-scripts
ifcfg-bond0_slave_1  ...
ifcfg-bond0_slave_2  ...
ifcfg-eth0  ...
ifcfg-eth1  ...
...
```

k. Use the `nmcli con delete` command to delete the slave connections.

```
[host02]# nmcli con delete "bond0 slave 1"
[host02]# nmcli con delete "bond0 slave 2"
```

l. Use the `nmcli con` command to view the network connections.

- Note that the slave connections no longer exist.

```
[host02]# nmcli con
NAME                UUID    TYPE              DEVICE
eth1                ...     802-3-ethernet    eth1
eth2                ...     802-3-ethernet    --
eth3                ...     802-3-ethernet    --
eth0                ...     802-3-ethernet    eth0
```

m. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

- Note that the network configuration files for the slaves no longer exist.

```
[host02]# ls /etc/sysconfig/network-scripts
ifcfg-eth0  ...
ifcfg-eth1  ...
...
```

6. Remove bond and slave connections on **host01**.

- Use the command line to remove the connections.

a. On **host01**, use the `nmcli con` command to view the network connections.

```
[host01]# nmcli con
NAME                UUID    TYPE              DEVICE
eth1                ...     802-3-ethernet    --
eth2                ...     802-3-ethernet    --
eth3                ...     802-3-ethernet    --
bond-slave-eth3     ...     802-3-ethernet    eth3
bond-slave-eth2     ...     802-3-ethernet    eth2
bond0               ...     bond              bond0
eth0                ...     802-3-ethernet    eth0
```

b. Use the `nmcli con delete` command to delete the bond and the slave connections.

```
[host01]# nmcli con delete bond0
[host01]# nmcli con delete bond-slave-eth2
[host01]# nmcli con delete bond-slave-eth3
```

c. Use the `nmcli con` command to view the network connections.

- Note that the bond and slave connections no longer exist.

```
[host01]# nmcli con
NAME                UUID    TYPE              DEVICE
eth1                ...     802-3-ethernet    eth1
eth2                ...     802-3-ethernet    --
eth3                ...     802-3-ethernet    --
```

```
eth0                ...      802-3-ethernet     eth0
```

d. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

- Note that the network configuration files for the bond and slaves no longer exist.

```
[host01]# ls /etc/sysconfig/network-scripts
ifcfg-eth0  ...
ifcfg-eth1  ...
ifcfg-eth2  ...
ifcfg-eth3  ...
...
```

e. Use the `ip link` command to view the links.

- Note that the `bond0` entry no longer exists.

```
[host01]# ip link
...
```

f. Use the `ls` command to view the contents of the `/sys/class/net` directory.

- Note that the `bond0` directory no longer exists.

```
[host01]# ls /sys/class/net
bonding_masters  eth0  eth1  eth2  eth3  lo
```

g. Use the `cat` command to view the `/sys/class/net/bonding_masters` file.

- Note that the file is empty.

```
[host01]# cat /sys/class/net/bonding_masters
```

h. Use the `ls` command to view the contents of the `/proc/net/bonding` directory.

- Note that the directory is empty.

```
[host01]# ls /proc/net/bonding
```

Practice 10-4: Configuring 802.1Q VLAN Tagging by Using the GUI

Overview

In this practice, you:

- Ensure that the VLAN (`8021q`) kernel module is loaded on **host02**
- Use the "Network Settings Editor" to configure VLAN tagging on **host02**
- Review network configuration on **host02**

Assumptions

- You are the `root` user on **host02**.

Tasks

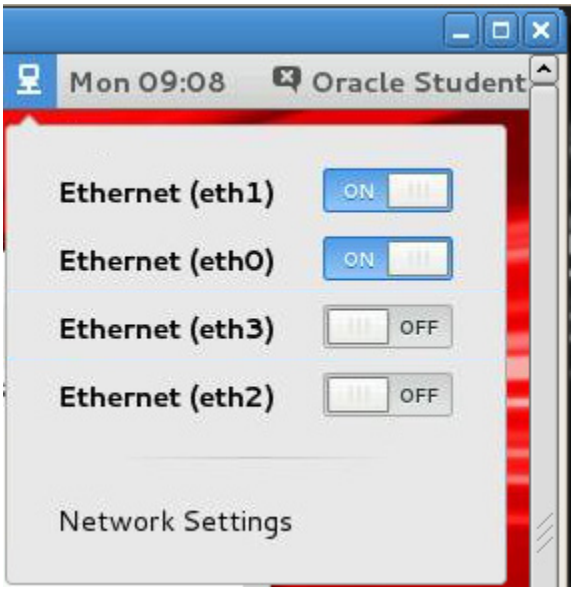
1. On **host02**, load the VLAN (`8021q`) kernel module if necessary.
   a. Use the `lsmod` command to view the loaded kernel modules.
      - Pipe the output to `grep` and search for "`8021q`".
      - In this example, the kernel module is not loaded.

```
# lsmod | grep 8021q
```

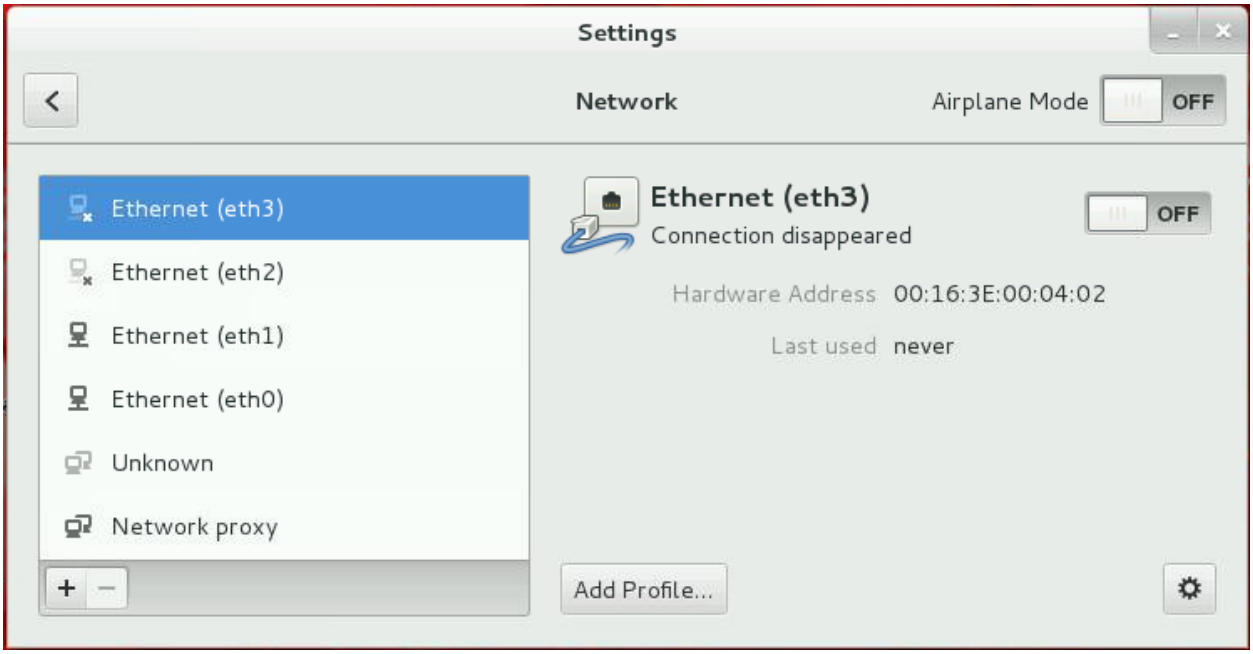
   b. If the kernel module is not loaded, use the `modprobe` command to load the `8021q` kernel module.
      - Use the `lsmod` command to ensure `8021q` is loaded.

```
# modprobe 8021q
# lsmod | grep 8021q
8021q     20082    0
...
```

2. Use the “Network Settings Editor” to configure VLAN tagging.
   a. Click the network icon from the GNOME desktop notification area.
      - The drop-down menu includes four Ethernet interfaces and the “Network Settings” option.

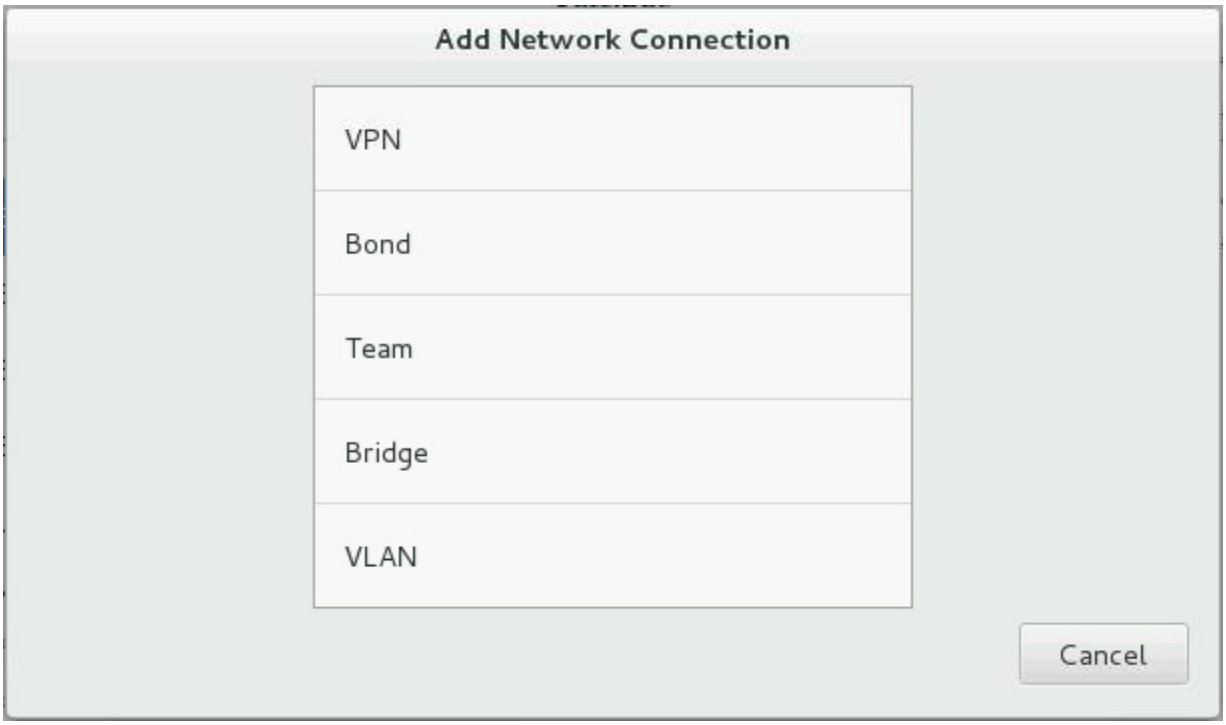


   b. Click the “Network Settings” option from the drop-down menu.
      - The “Network Settings Editor” appears.



c. Click the "+" button to add a new connection type.

- The "Add Network Connection" window appears.



d. Click "VLAN" to add a VLAN connection.

- The following window appears.
- The default Connection name is "VLAN connection 1."

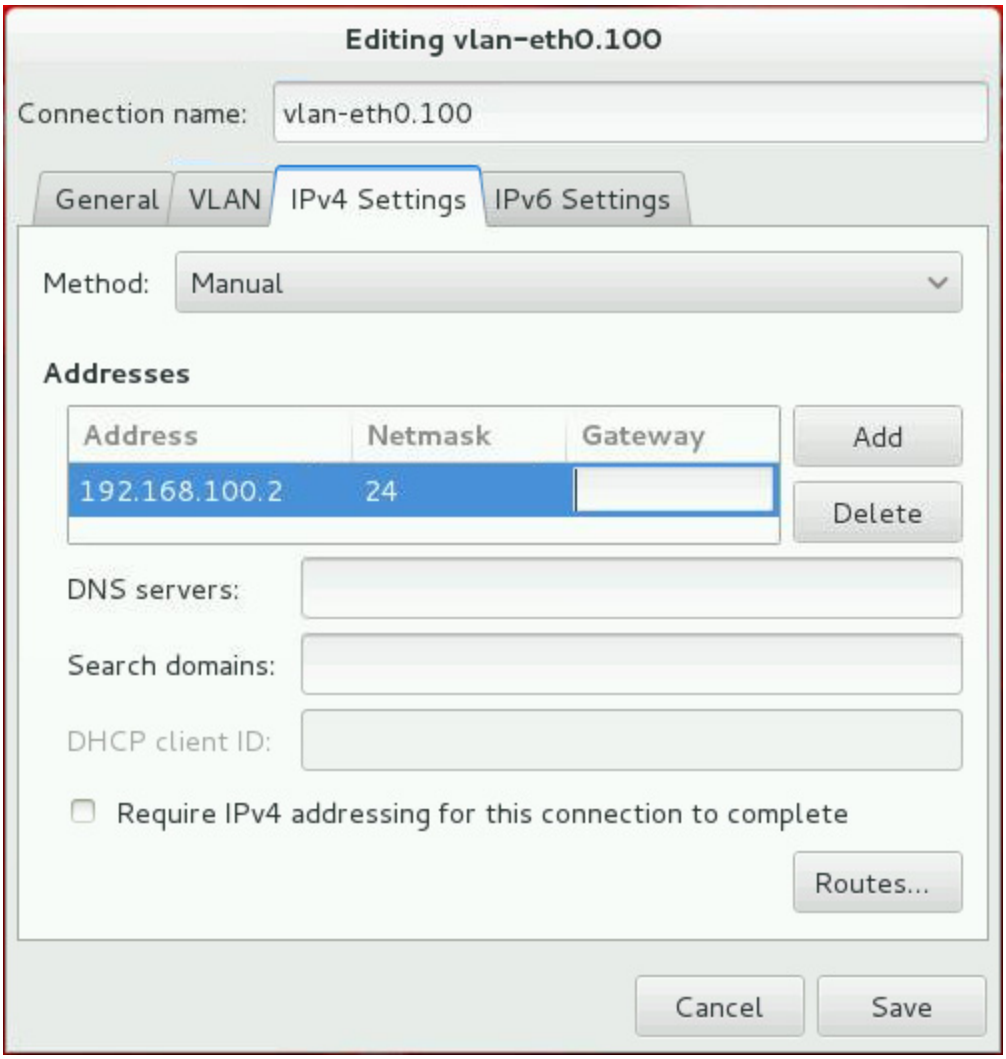
e. Update the screen as follows.

- Change “Connection name:” to `vlan-eth0.100`.
- Click the “Parent interface:” down arrow and select `eth0 (00:16:3E:00:01:02)`.
- Change “VLAN id:” to `100`.
- Change “VLAN interface name:” to `eth0.100`.

- The window appears as follows.

f. Click the “IPv4 Settings” tab to assign an IPv4 address to the VLAN interface.
   - Change the “Method” to “Manual.”
   - Click “Add” to add the following Address information:
     - Address: `192.168.100.2`
     - Netmask: `24`
     - Gateway: `<empty>`

- The window appears as follows.

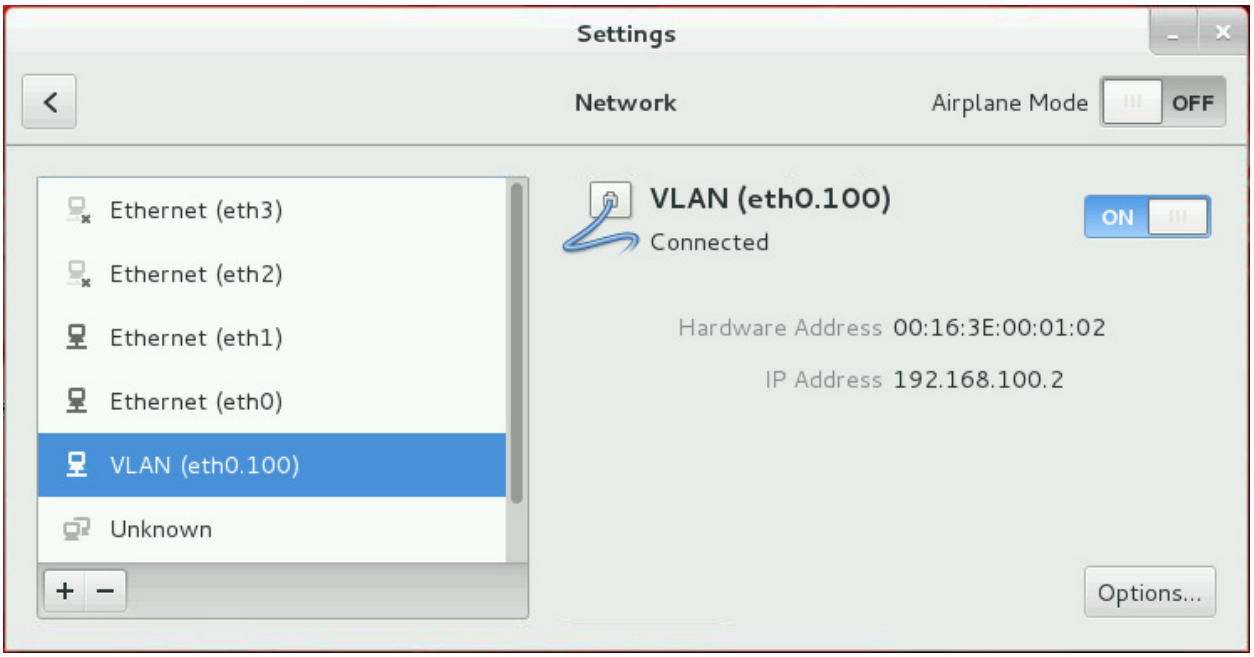


g. Click “Save” to complete configuring VLAN tagging.

- The “VLAN (eth0.100)” interface now appears in the “Network Settings” window.

h. Select the “VLAN (eth0.100)” entry to display the following window.

- Note that the Addresses are shown.



i. Click the “X” in the top-right corner to close the window.

3. View the network interfaces on **host02**.

a. Use the `ip addr` command to view the protocol addresses for the network devices.

- Note that the `eth0.100` device exists.
- Note that the `eth0.100` MAC address is the same as the `eth0` MAC address.
- Note that the `192.168.100.2/24` IPv4 address is assigned to the `eth0.100` device.

```
# ip addr
...
2: eth0: ...
    link/ether 00:16:3e:00:01:02 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.102/24 brd 192.0.2.255 scope global eth0
...
7: eth0.100@eth0: ...
    link/ether 00:16:3e:00:01:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.2/24 brd 192.168.100.255 scope ... eth0.100
...
```

b. Use the `nmcli dev` command to view the network devices.
   - Note that the `eth0.100` device is associated with the `vlan-eth0.100` connection.

```
# nmcli dev
NAME            TYPE           STATE           CONNECTION
eth0            ethernet       connected       eth0
eth1            ethernet       connected       eth1
eth0.100        vlan           connected       vlan-eth0.100
...
```

c. Use the `nmcli con` command to view the network connections.
   - Note that the `vlan-eth0.100` connection is listed.

```
# nmcli con
NAME               UUID    TYPE              DEVICE
vlan-eth0.100      ...     vlan              eth0.100
eth0               ...     802-3-ethernet    eth0
eth1               ...     802-3-ethernet    eth1
```

d. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.
   - Note that there is a network configuration file for the VLAN interface, `ifcfg-vlan-eth0.100`.

```
# ls /etc/sysconfig/network-scripts
ifcfg-eth0           ...
ifcfg-eth1           ...
ifcfg-lo             ...
ifcfg-vlan-eth0.100  ...
...
```

e. Use the `cat` command to view the contents of the `ifcfg-vlan-eth0.100` file.
   - Note that the "DEVICE" setting is `eth0.100`.
   - Note that the "PHYSDEV" setting is `eth0`.

```
# cat /etc/sysconfig/network-scripts/ifcfg-vlan-eth0.100
VLAN=yes
TYPE=Vlan
DEVICE=eth0.100
PHYSDEV=eth0
VLAN_ID=100
REORDER_HDR=0
BOOTPROTO=none
IPADDR=192.168.100.2
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
```

```
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=vlan-eth0.100
UUID=...
ONBOOT=yes
```

Practice 10-5: Configuring 802.1Q VLAN Tagging from the Command Line

Overview

In this practice, you:

- Ensure that the VLAN (`8021q`) kernel module is loaded on **host01**
- Create an 802.1Q VLAN interface on **host01**

Assumptions

- You are the `root` user on **host01**.

Tasks

1. On **host01**, load the VLAN (`8021q`) kernel module if necessary.
   a. Use the `lsmod` command to view the loaded kernel modules.
      - Pipe the output to `grep` and search for "`8021q`".
      - In this example, the kernel module is not loaded.

```
# lsmod | grep 8021q
```

   b. If the kernel module is not loaded, use the `modprobe` command to load the `8021q` kernel module.
      - Use the `lsmod` command to ensure `8021q` is loaded.

```
# modprobe 8021q
# lsmod | grep 8021q
8021q     20082    0
...
```

2. On **host01**, create an 802.1Q VLAN interface and view the results.
   a. Use the `nmcli con add` command to create the VLAN interface.
      - Use the "`type vlan`" argument to specify an 802.1q tagged virtual LAN interface.
      - Use the "`con-name vlan-eth0.100`" argument to specify the name of the new VLAN connection.
      - Use the "`ifname eth0.100`" argument to specify the interface to bind the connection to.
      - Use the "`dev eth0`" argument to specify the parent device this VLAN is on.
      - Use the "`id 100`" argument to specify the VLAN ID.
      - Use the "`ip4 192.168.100.1/24`" argument to specify IPv4 address to assign to the interface.

```
# nmcli con add type vlan con-name vlan-eth0.100 ifname eth0.100
dev eth0 id 100 ip4 192.168.100.1/24
Connection 'vlan-eth0.100' (<UUID>) successfully added.
```

   b. Use the `ip addr` command to view the protocol addresses for the network devices.
      - Note that the `eth0.100` device exists.
      - Note that the `eth0.100` MAC address is the same as the `eth0` MAC address.

- Note that the `192.168.100.1/24` IPv4 address is assigned to the `eth0.100` device.

```
# ip addr
...
2: eth0: ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.101/24 brd 192.0.2.255 scope global eth0
...
8: eth0.100@eth0: ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.1/24 brd 192.168.100.255 scope ... eth0.100
...
```

c. Use the `nmcli dev` command to view the network devices.

- Note that the `eth0.100` device is associated with the `vlan-eth0.100` connection.

```
# nmcli dev
NAME            TYPE            STATE           CONNECTION
eth0            ethernet        connected       eth0
eth0.100        vlan            connected       vlan-eth0.100
...
```

d. Use the `nmcli con` command to view the network connections.

- Note that the `vlan-eth0.100` connection is listed.

```
# nmcli con
NAME               UUID    TYPE              DEVICE
vlan-eth0.100      ...     vlan              eth0.100
eth0               ...     802-3-ethernet    eth0
...
```

e. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

- Note that there is a network configuration file for the VLAN interface, `ifcfg-vlan-eth0.100`.

```
# ls /etc/sysconfig/network-scripts
ifcfg-eth0           ...
ifcfg-eth1           ...
...
ifcfg-vlan-eth0.100  ...
...
```

f. Use the `cat` command to view the contents of the `ifcfg-vlan-eth0.100` file.

- Note that the "DEVICE" setting is `eth0.100`.

- Note that the “PHYSDEV” setting is `eth0`.

```
# cat /etc/sysconfig/network-scripts/ifcfg-vlan-eth0.100
VLAN=yes
TYPE=Vlan
DEVICE=eth0.100
PHYSDEV=eth0
VLAN_ID=100
REORDER_HDR=0
BOOTPROTO=none
IPADDR=192.168.100.1
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=vlan-eth0.100
UUID=...
ONBOOT=yes
```

Practice 10-6: Working with VLAN Interfaces

Overview

In this practice, you:

- Test connectivity between the VLAN interfaces on **host01** and **host02**
- Use `tcpdump` to view tagged and untagged packets
- Explore the contents of the `/sys/class/net/eth0.100` directory
- Explore the contents of the `/proc/net/vlan` directory
- Remove the VLAN interfaces on **host01** and **host02**

Assumptions

- You are the `root` user on **host01**.
- You are the `root` user on **host02**.

Tasks

1. Test connectivity between the VLAN interfaces on **host01** and **host02**.
   a. From **host02**, use the `ping` command to communicate to the VLAN interface on **host01**.
      - The IP address of the VLAN interface on **host01** is `192.168.100.1`.
      - Press CTRL-C to exit after a few lines of output.

```
[host02]# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
^C
...
```

   b. From **host02**, use the `netstat -r` command to view the route table.
      - Note that the route to `192.168.100.0` is through the `eth0.100` interface.

```
[host02]# netstat -r
Kernel IP routing table
Destination     ...  Iface
Default         ...  eth0
192.0.2.0       ...  eth0
192.168.1.0     ...  eth1
192.168.100.0   ...  eth0.100
```

   c. From **host01**, use the `ping` command to communicate to the VLAN interface on **host02**.
      - The IP address of the VLAN interface on **host02** is `192.168.100.2`.
      - Press CTRL-C to exit after a few lines of output.

```
[host01]# ping 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=...
^C
...
```

d. From **host01**, use the `netstat -r` command to view the route table.

- Note that the route to `192.168.100.0` is through the `eth0.100` interface.

```
[host01]# netstat -r
Kernel IP routing table
Destination     ...  Iface
...
192.168.100.0   ...  eth0.100
```

2. Use `tcpdump` to view tagged and untagged packets.

- You first observe traffic on the VLAN interface, `eth0.100`, where you do not see VLAN tags.
- You next observe traffic on the parent interface, `eth0`, where you do see VLAN tags.

a. On **host02**, open a second terminal window.

b. Use the `su -` command to become the `root` user in this second terminal.

- The `root` password is `oracle`.

```
[host02]$ su -
Password: oracle
[host02]#
```

c. In this second terminal window, enter the following `tcpdump` command.

- Use the `-e` option to view the Ethernet header, which includes the 802.1Q tags.
- Use the `-i eth0.100` to sniff on the VLAN interface.

```
[host02]# tcpdump -e -i eth0.100
tcpdump: verbose output suppressed, use -v or -vv for full ...
listening on eth0.100, link-type EN10MB (Ethernet), capture ...
```

d. On **host02**, in the first terminal window, use the `ping` command to communicate to the VLAN interface on **host01**.

- The IP address of the VLAN interface on **host01** is `192.168.100.1`.
- Press CTRL-C to exit after a few lines of output.

```
[host02]# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
^C
```

```
...
```

e. In the second terminal window on **host02**, view the output of the `tcpdump` command.

- Note that you see normal traffic without VLAN tags.

```
... 00:16:3e:00:01:02 (oui Unknown) > Broadcast, ethertype ARP
(0x0806), length 42: Request who-has 192.168.100.1 tell
192.168.100.2, length 28
... 00:16:3e:00:01:01 (oui Unknown) > 00:16:3e:00:01:02 (oui
Unknown), ethertype ARP (0x0806), length 42: Reply 192.168.100.1
is at 00:16:3e:00:01:01 (oui Unknown), length 28
... 00:16:3e:00:01:01 (oui Unknown) > 00:16:3e:00:01:02 (oui
Unknown), ethertype IPv4 (0x0800), length 98: 192.168.100.1 >
192.168.100.2: ICMP echo reply, id 15342, seq 1, length 64
...
```

f. In the second terminal window on **host02**, press CTRL-C to exit the `tcpdump` command.

```
... 00:16:3e:00:01:01 (oui Unknown) > 00:16:3e:00:01:02 (oui
Unknown), ethertype IPv4 (0x0800), length 98: 192.168.100.1 >
192.168.100.2: ICMP echo reply, id 15342, seq 1, length 64
...
^C
... packets captured
... packets received by filter
... packets dropped by kernel
```

g. In the second terminal window on **host02**, enter the following `tcpdump` command.

- Use the `-e` option to view the Ethernet header, which includes the 802.1Q tags.
- Use the `-i eth0` to sniff on the physical interface.
- Optionally, use the `clear` command to clear the screen before running `tcpdump`.

```
[host02]# clear
[host02]# tcpdump -e -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full ...
listening on eth0, link-type EN10MB (Ethernet), capture size ...
```

h. On **host02**, in the first terminal window, use the `ping` command to communicate to the VLAN interface on **host01**.

- The IP address of the VLAN interface on **host01** is `192.168.100.1`.

- Press CTRL-C to exit after a few lines of output.

```
[host02]# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=...
^C
...
```

i. In the second terminal window on **host02**, view the output of the `tcpdump` command.

- Note that you see the tagged 802.1Q packets (**vlan 100** is in bold font in the sample output).

```
... 00:16:3e:00:01:02 (oui Unknown) > Broadcast, ethertype
802.1Q (0x8100), length 46: vlan 100, p 0, ethertype ARP,
Request who-has 192.168.100.1 tell 192.168.100.2, length 28
... 00:16:3e:00:01:01 (oui Unknown) > 00:16:3e:00:01:02 (oui
Unknown), ethertype 802.1Q (0x8100), length 46: vlan 100, p 0,
ethertype ARP, Reply 192.168.100.1 is at 00:16:3e:00:01:01 (oui
Unknown), length 28
...
```

j. In the second terminal window on **host02**, press CTRL-C to exit the `tcpdump` command.

```
... 00:16:3e:00:01:01 (oui Unknown) > 00:16:3e:00:01:02 (oui
Unknown), ethertype 802.1Q (0x8100), length 46: vlan 100, p 0,
ethertype ARP, Reply 192.168.100.1 is at 00:16:3e:00:01:01 (oui
Unknown), length 28
...
^C
... packets captured
... packets received by filter
... packets dropped by kernel
```

k. Click the "X" in the upper-right corner of the second terminal window to close the window.

- Click "Close Terminal" if prompted.

3. View the contents of `/sys/class/net/eth0.100/`.

- Each network interface contains a directory in `/sys/class/net`.

a. From **host01**, use the `cd` command to change to the `/sys/class/net` directory.

- Use the `ls` command to display the contents of the directory.
- Note that `eth0.100` is a directory.

```
[host01]# cd /sys/class/net
[host01]# ls
bonding_masters  eth0  eth0.100  eth1  eth2  eth3  lo
```

b. Use the `cd` command to change to the `eth0.100` directory.

- Use the `ls` command to display the contents of the directory.

```
[host01]# cd eth0.100
[host01]# ls
addr_assign_type  carrier  flags    link_mode     power       ...
address           dev_id   ifalias  mtu           queues      ...
addr_len          dormant  ifindex  netdev_group  speed       ...
broadcast         duplex   iflink   operstate     statistics  ...
```

c. Use the `cat` command to view the `operstate` file.

```
[host01]# cat operstate
up
```

d. Use the `cat` command to view the `address` file.

```
[host01]# cat address
00:16:3e:00:01:01
```

e. Use the `cat` command to view the `uevent` file.

- Sample output is shown. The “IFINDEX” value might be different.

```
[host01]# cat uevent
DEVTYPE=vlan
INTERFACE=eth0.100
IFINDEX=8
```

4. View the `/proc/net/vlan` directory.

a. From **host01**, use the `cd` command to change to the `/proc/net/vlan` directory.

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

```
[host01]# cd /proc/net/vlan
[host01]# ls
config  eth0.100
```

b. Use the `cat` command to view the `config` file.

```
[host01]# cat config
VLAN Dev name    | VLAN ID
Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VID_NO_PAD
eth0.100       | 100  | eth0
```

c. Use the `cat` command to view the `eth0.100` file.

- Sample output is shown.
- Egress traffic begins inside of a network and proceeds through its routers to a destination somewhere outside of the network.

- Ingress traffic originates from outside of the network's routers and proceeds toward a destination inside of the network.

```
[host01]# cat eth0.100
eth0.100  VID: 100       REORDER_HDR: 1  dev->priv_flags: 1
         total frames received           11
          total bytes received          700
      Broadcast.Multicast Rcvd            0

      total frames transmitted           19
       total bytes transmitted         1382
Device: eth0
INGRESS priority mappings: 0:0  1:0  2:00  3:0  4:0  5:0  ...
 EGRESS priority mappings
```

5. Remove VLAN interface on **host01**.
   a. Use the `nmcli con` command to view the network connections.

```
[host01]# nmcli con
NAME               UUID    TYPE              DEVICE
eth0               ...     802-3-ethernet    eth0
...
vlan-eth0.100      ...     vlan              eth0.100
```

   b. Use the `nmcli con delete` command to delete the `vlan-eth0.100` connection.

```
[host01]# nmcli con delete vlan-eth0.100
```

   c. Use the `nmcli con` command to view the network connections.
      - Note that the VLAN connection no longer exists.

```
[host01]# nmcli con
NAME               UUID    TYPE              DEVICE
eth0               ...     802-3-ethernet    eth0
...
```

   d. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.
      - Note that the network configuration file for the VLAN interface no longer exists.

```
[host01]# ls /etc/sysconfig/network-scripts
ifcfg-eth0  ...
...
```

   e. Use the `ip link` command to view the links.
      - Note that the `eth0.100` device no longer exists.

```
[host01]# ip link
...
```

f. Use the `ls` command to view the contents of the `/sys/class/net` directory.

- Note that the `eth0.100` directory no longer exists.

```
[host01]# ls /sys/class/net
bonding_masters  eth0  eth1  eth2  eth3  lo
```

g. Use the `ls` command to view the contents of the `/proc/net/vlan` directory.

- Note that the `eth0.100` file no longer exists.

```
[host01]# ls /proc/net/vlan
config
```

h. Use the `cat` command to view the `config` file.

- Note that the file only contains header information.

```
[host01]# cat /proc/net/vlan/config
VLAN Dev name    | VLAN ID
Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VID_NO_PAD
```

6. Remove VLAN interface on **host02**.

- On **host02**, use "Network Settings Editor" to view the VLAN interface details.

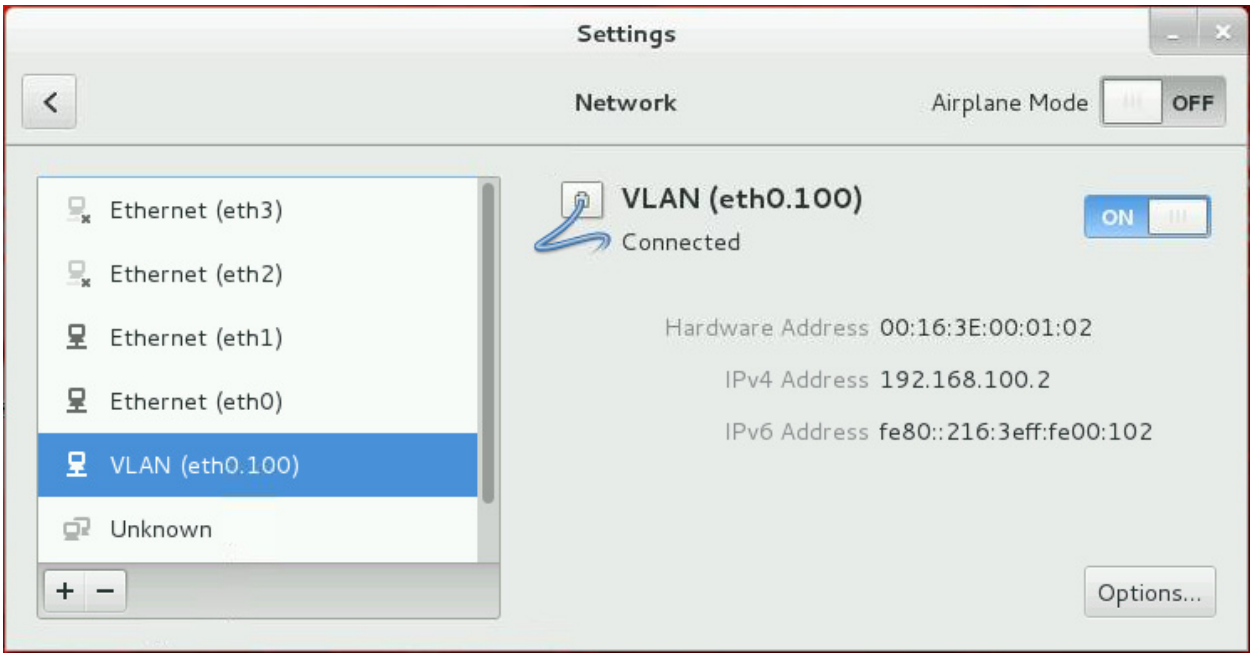
a. Click the network icon from the GNOME desktop notification area.

- The drop-down menu includes the "Network Settings" option.

b. Click the "Network Settings" option from the menu.

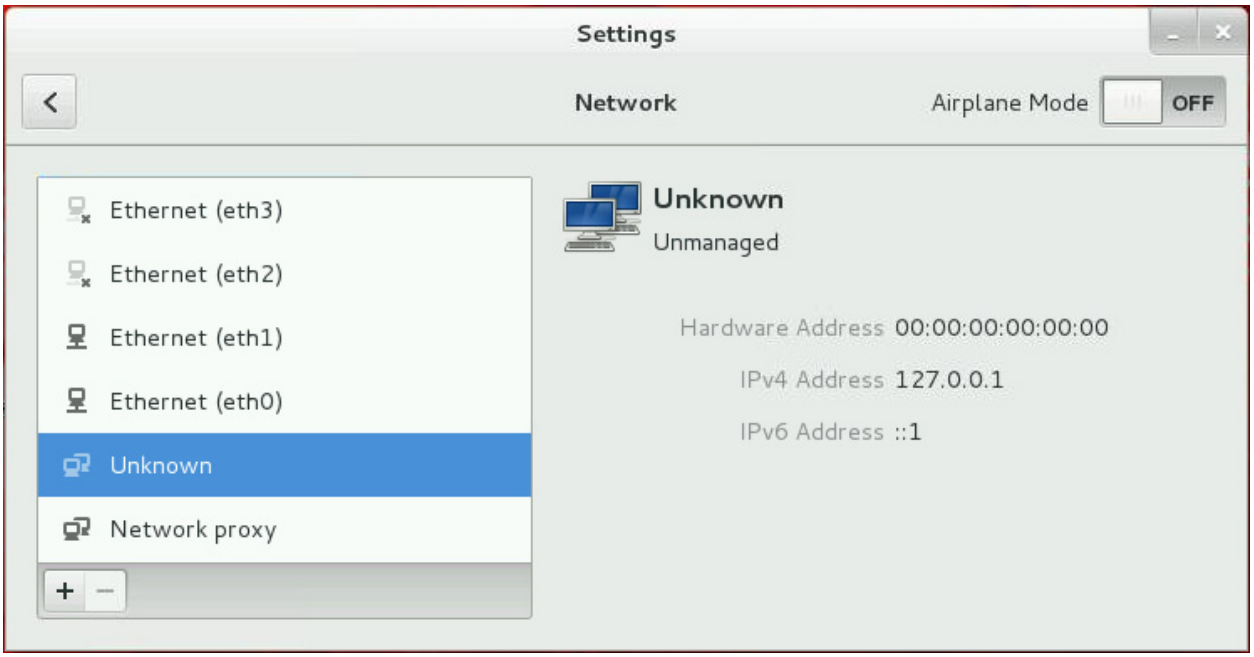
- The "Network Settings Editor" appears, which includes the "VLAN (eth0.100)" interface.

c. Click the "VLAN (eth0.100)" entry.



d. Click the "-" button to remove the VLAN interface.

- The window appears as follows.



e. Click the "X" in the top-right corner to close the window.

f. Use the `nmcli con` command to view the network connections.

- Note that the VLAN connection no longer exists.

```
[host02]# nmcli con
NAME                UUID     TYPE              DEVICE
eth0                ...      802-3-ethernet    eth0
...
```

g. Use the `ls` command to view the `/etc/sysconfig/network-scripts/` directory.

- Note that the network configuration file for the VLAN interface no longer exists.

```
[host02]# ls /etc/sysconfig/network-scripts
ifcfg-eth0   ...
...
```

h. Use the `ip link` command to view the links.

- Note that the `eth0.100` device no longer exists.

```
[host02]# ip link
...
```

i. Use the `ls` command to view the contents of the `/sys/class/net` directory.

- Note that the `eth0.100` directory no longer exists.

```
[host02]# ls /sys/class/net
bonding_masters  eth0  eth1  eth2  eth3  lo
```

j. Use the `ls` command to view the contents of the `/proc/net/vlan` directory.
   - Note that the `eth0.100` file no longer exists.

```
[host02]# ls /proc/net/vlan
config
```

k. Use the `cat` command to view the `config` file.
   - Note that the file only contains header information.

```
[host02]# cat /proc/net/vlan/config
VLAN Dev name    | VLAN ID
Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VID_NO_PAD
```

7. In preparation for the next practice, power off **host01**, **host02**, and **host03**.
   a. From **host01**, use the `systemctl` command to power off **host01**.

```
[host01]# systemctl poweroff
```

   b. From **host02**, use the `systemctl` command to power off **host02**.

```
[host02]# systemctl poweroff
```

   c. From **dom0**, use the `xm shutdown -w` command to power off **host03**.

```
[dom0]# xm shutdown –w host03
```

   - If the `xm shutdown` command takes more than a few seconds, use the `xm destroy` command to power off **host03**.

```
[dom0]# xm destroy host03
```

   d. From **dom0**, use the `xm list` command.
   - Note that **host01**, **host02** and **host03** are no longer active.

```
[dom0]# xm list
Name                  ID     Mem  VCPUs       State       Time(s)
Domain-0               0    2048      2      r-----        ...
```

Practice 10-7: Configuring a Site-to-Site VPN

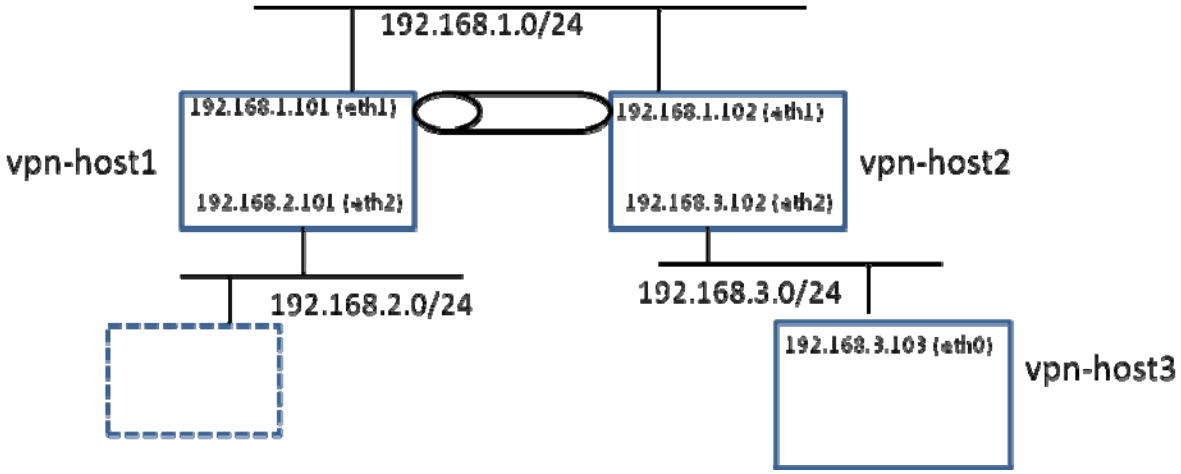
Overview

In this practice, you:

- Explore and start the **vpn-host1** and **vpn-host2** virtual machines
- Generate RSA authentication keys for **vpn-host1** and **vpn-host2**
- Update the `/etc/ipsec.conf` file for **vpn-host1** and **vpn-host2**
- Stop the `firewalld` service on **vpn-host1** and **vpn-host2**
- Start the `ipsec` service on **vpn-host1** and **vpn-host2**
- Verify connectivity between **vpn-host1** and **vpn-host2**
- Shut down the **vpn-host1** and **vpn-host2** virtual machines
- Start the **host01**, **host02**, and **host03** virtual machines

Assumptions

- You are the `root` user on **dom0**.
- The **host01**, **host02**, and **host03** virtual machines are shut down.
- The **vpn-host1** and **vpn-host2** virtual machines exist on your system.
- The following describes the **vpn*** virtual machines network configuration.



- You create a VPN tunnel from **vpn-host1** to **vpn-host2**.

Tasks

1. Explore and start the **vpn-host1** virtual machine configuration file.
   a. From **dom0**, use the `cd` command to change to the `/OVS/running_pool/vpn-host1` directory.

```
[dom0]# cd /OVS/running_pool/vpn-host1
```

   b. Use the `cat` command to view the `vm.cfg` file for **vpn-host1**.
      - Note that there are three virtual network interfaces:
        – The interface on the `virbr0` bridge is `eth0` with IP address `192.0.2.111`. This interface provides access to the Yum repository on **dom0**.
        – The interface on the `virbr1` bridge is `eth1` with IP address `192.168.1.101`.

- The interface on the `virbr2` bridge is `eth2` with IP address `192.168.2.101`.

```
[dom0]# cat vm.cfg
name = 'vpn-host1'
builder = 'hvm'
memory = 1536
boot = 'cd'
disk = [ 'file:/OVS/running_pool/vpn-host1/system.img,hda,w',
         'file:/OVS/seed_pool/OracleLinux-R7-U1-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',
        'mac=00:16:3e:00:03:01, bridge=virbr2']
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'
```

c. Use the `xm create` command to start the **vpn-host1** virtual machine.

```
[dom0]# xm create vm.cfg
```

2. Explore and start the **vpn-host2** virtual machine configuration file.

a. From **dom0**, use the `cd` command to change to the `/OVS/running_pool/vpn-host2` directory.

```
[dom0]# cd /OVS/running_pool/vpn-host2
```

b. Use the `cat` command to view the `vm.cfg` file for **vpn-host2**.

- Note that there are three virtual network interfaces:
  - The interface on the `virbr0` bridge is `eth0` with IP address `192.0.2.112`. This interface provides access to the Yum repository on **dom0**.
  - The interface on the `virbr1` bridge is `eth1` with IP address `192.168.1.102`.
  - The interface on the `virbr3` bridge is `eth2` with IP address `192.168.3.102`. Note that `eth2` on **vpn-host2** is on a different bridge and different subnet than `eth2` on **vpn-host1**.

```
[dom0]# cat vm.cfg
name = 'vpn-host2'
```

```
builder = 'hvm'
memory = 1536
boot = 'cd'
disk = [ 'file:/OVS/running_pool/vpn-host2/system.img,hda,w',
         'file:/OVS/seed_pool/OracleLinux-R7-U1-Server-x86_64-
dvd.iso,hdc:cdrom,r']
vif = [ 'mac=00:16:3e:00:01:02, bridge=virbr0',
        'mac=00:16:3e:00:02:02, bridge=virbr1',
        'mac=00:16:3e:00:04:02, bridge=virbr3']
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'
```

c. Use the `xm create` command to start the **vpn-host2** virtual machine.

```
[dom0]# xm create vm.cfg
```

3. Log in to **vpn-host1** by using `vncviewer`.

a. From **dom0**, determine the VNC port number for **vpn-host1** by running the following `xm list` command.

```
[dom0]# xm list –l vpn-host1 | grep location
               (location 0.0.0.0:5902)
               (location 3)
```

- The sample shown indicates that the port number is `5902`. This might not be true in your case.

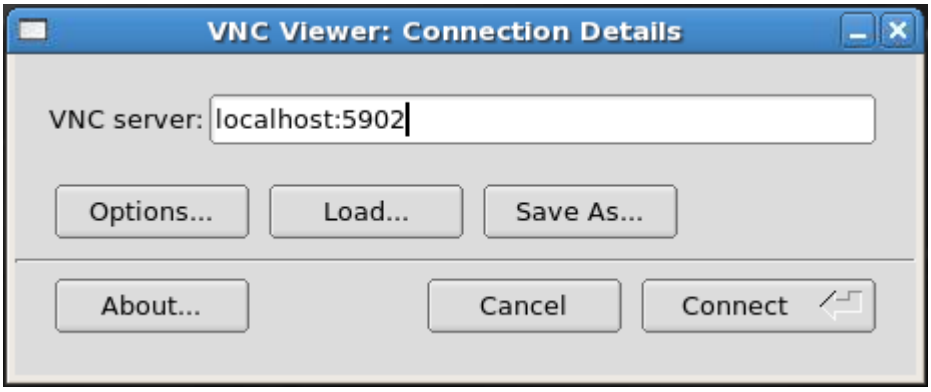
b. From **dom0**, run the `vncviewer&` command.

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

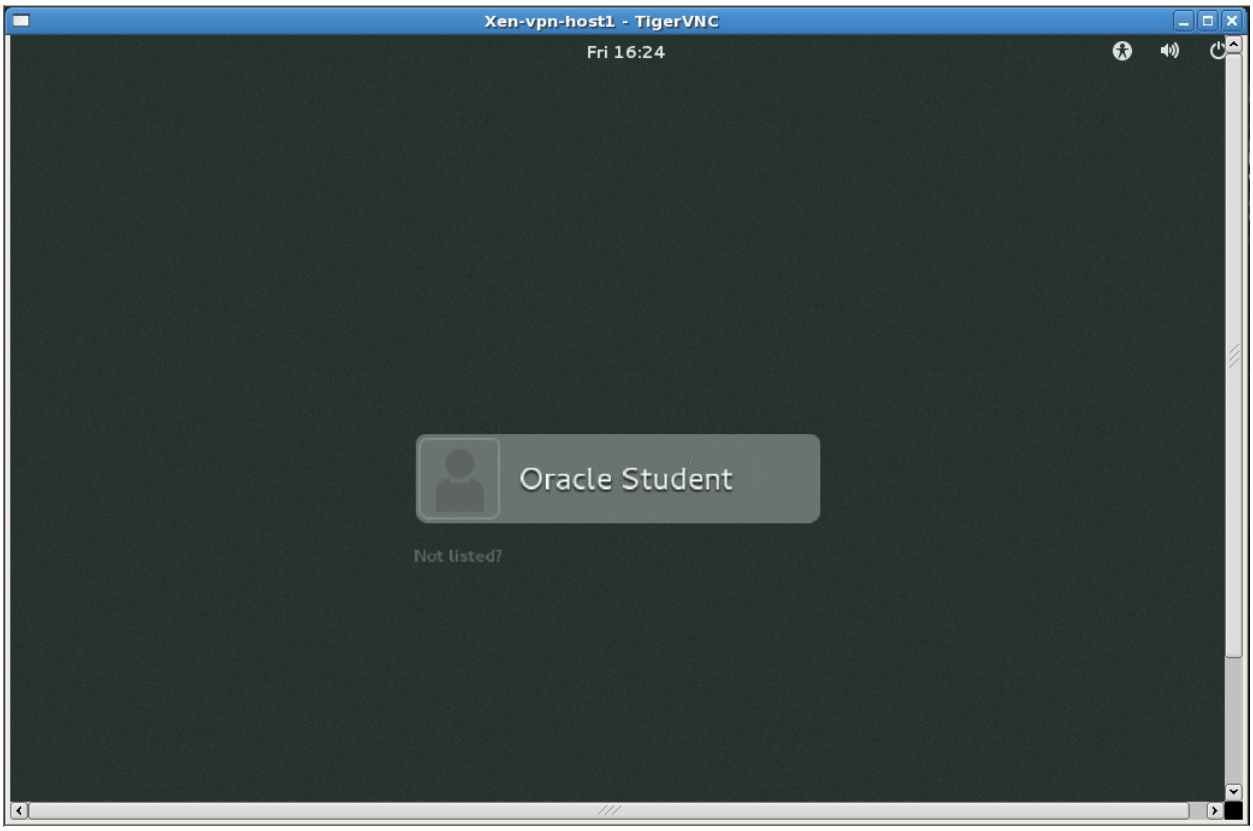
- The "VNC Viewer: Connection Details" dialog box appears.

c. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list –l vpn-host1 | grep location` command.

- For example, if the port number is `5902`, enter `localhost:5902` and click "Connect."



- The GNOME login screen appears. You might need to press ENTER to display the login screen.

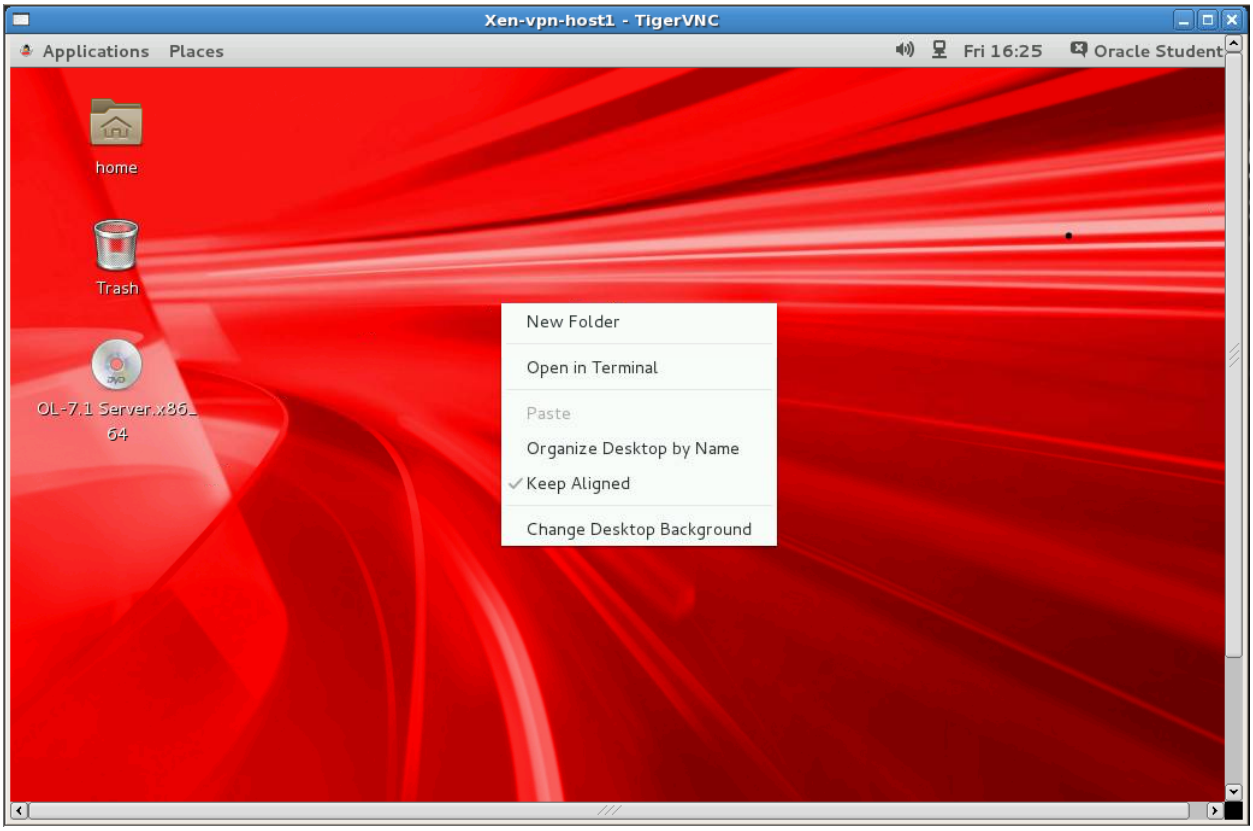


d. Click “Oracle Student” in the list of users. You are prompted for the password.

e. Enter `oracle` for the Password and click “Sign In.”

- The GNOME desktop appears.

f. Right-click 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`.

```
[vpn-host1]$ su -
Password: oracle
[vpn-host1]#
```

4. Generate a new RSA authentication key for **vpn-host1**.

   Use the `ipsec newhostkey` command to generate the key.
   - The `--configdir` option specifies the Network Security Services (NSS) configuration directory where the certificate key and databases reside.
   - The `--output` option is mandatory.
   - This command might take a couple minutes to complete.

```
[vpn-host1]# ipsec newhostkey --configdir /etc/ipsec.d --output
/etc/ipsec.d/www.example.com.secrets
Generated RSA key pair using the NSS database
```

5. Log in to **vpn-host2** by using `vncviewer`.
   a. From **dom0**, open a second terminal window by clicking the Terminal icon on the desktop.
   b. In the second terminal window on **dom0**, use the `su -` command to become the `root` user.
      - The `root` password is `oracle`.

```
[dom0]$ su -
Password: oracle
[dom0]#
```

c. Determine the VNC port number for **vpn-host2** by running the following `xm list` command.

```
[dom0]# xm list -l vpn-host2 | 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.

d. Run the `vncviewer&` command.

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

- The "VNC Viewer: Connection Details" dialog box appears.

e. Enter `localhost:<port_number>`, substituting the port number displayed from the previous `xm list -l vpn-host2 | grep location` command.

- For example, if the port number is `5903`, enter `localhost:5903` and click "Connect."
- The GNOME login screen appears. You might need to press ENTER to display the login screen.

f. Click "Oracle Student" in the list of users. You are prompted for the password.

g. Enter `oracle` for the Password and click "Sign In."

- The GNOME desktop appears.

h. Right-click the desktop to display the pop-up menu.

i. From the pop-up menu, click "Open in Terminal."

- A terminal window appears.

j. In the terminal window, use the `su -` command to become the `root` user.

- The `root` password is `oracle`.

```
[vpn-host2]$ su -
Password: oracle
[vpn-host2]#
```

6. Generate a new RSA authentication key for **vpn-host2**.

   Use the `ipsec newhostkey` command to generate the key.

   - This command might take a couple minutes to complete.

```
[vpn-host2]# ipsec newhostkey --configdir /etc/ipsec.d --output
/etc/ipsec.d/www.example.com.secrets
Generated RSA key pair using the NSS database
```

7. Put the appropriate RSA keys in the `/etc/ipsec.conf` file.

   a. On **vpn-host1**, use the `cd` command to change to the `/etc` directory.

```
[vpn-host1] cd /etc
```

   b. Use `ipsec showhostkey --left` to display host key on left host, **vpn-host1**.

   - The sample output is shown.

```
[vpn-host1]# ipsec showhostkey --left
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"
ipsec showhostkey loading secrets from
"/etc/ipsec.d/www.example.com.secrets"
ipsec showhostkey loaded private key for keyid:
PPK_RSA:AQOuaErmq
        # rsakey AQOuaErmq
leftrsasigkey=0sAQOuaErmqqXZqWP/5tXPI2xXqR/qq8TPyGUnoUQ+rCkHy+WK
q14MrCcmPaHDVZfMIoRAN4Mot2k2535sHnc+SkWxaDyjueGKczTndALmck0eXXWa
WgcfNS94rH9wtleQuZXmTlnSQvW8kiHO1N1o22NrCRYZF8zrpQTNFC1WNAiO2qxW
ZSgdJn2q9iW6MFq0804AsNKI9QrrpC1n7xXyDrWhi+v5B73C0ly4/uYeNIotyK9C
ImM713QK3MUpZOSNnRiACIQYw8aX+YEKSgjPU3+nEHp243QeUVraIf5LE0cKtTQu
S3Ur1cgZfQZCFX1rGyHqD/ZtUyzL9Fvo5j04kjnZgJTywr4f0Tmw7a+2QJPIQQ52
iOv1jnV5WzbKB2zpDICsCzRZ7yVaK7MXrDxvbNss8gjXjK5BXgFLcVlFh/eJgcji
/AUK0S1vqXdYiJjWtZpjznRTDyE7+jqgLsSi0jY5y7i4dYhD+I0RujzTuv6z7ObD
+yLYpa/DoXQFMrFjB3kz9L+uqz7TtmwCthNdCJVJjnKL0jBIZ7IfVqBvIJoS5nra
WYbF/thUq7C6ziHML8AL2tUcx5wIne28ijJOT2LfjeU=
```

c. Select `leftrsasigkey=<string>` and copy it into the buffer.

- Highlight the string as shown.
- With the string highlighted, select “Edit > Copy” from the terminal window menu.

```
[vpn-host1]# ipsec showhostkey --left
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"
ipsec showhostkey loading secrets from
"/etc/ipsec.d/www.example.com.secrets"
ipsec showhostkey loaded private key for keyid:
PPK_RSA:AQOuaErmq
        # rsakey AQOuaErmq
leftrsasigkey=0sAQOuaErmqqXZqWP/5tXPI2xXqR/qq8TPyGUnoUQ+rCkHy+WK
q14MrCcmPaHDVZfMIoRAN4Mot2k2535sHnc+SkWxaDyjueGKczTndALmck0eXXWa
WgcfNS94rH9wtleQuZXmTlnSQvW8kiHO1N1o22NrCRYZF8zrpQTNFC1WNAiO2qxW
ZSgdJn2q9iW6MFq0804AsNKI9QrrpC1n7xXyDrWhi+v5B73C0ly4/uYeNIotyK9C
ImM713QK3MUpZOSNnRiACIQYw8aX+YEKSgjPU3+nEHp243QeUVraIf5LE0cKtTQu
S3Ur1cgZfQZCFX1rGyHqD/ZtUyzL9Fvo5j04kjnZgJTywr4f0Tmw7a+2QJPIQQ52
iOv1jnV5WzbKB2zpDICsCzRZ7yVaK7MXrDxvbNss8gjXjK5BXgFLcVlFh/eJgcji
/AUK0S1vqXdYiJjWtZpjznRTDyE7+jqgLsSi0jY5y7i4dYhD+I0RujzTuv6z7ObD
+yLYpa/DoXQFMrFjB3kz9L+uqz7TtmwCthNdCJVJjnKL0jBIZ7IfVqBvIJoS5nra
WYbF/thUq7C6ziHML8AL2tUcx5wIne28ijJOT2LfjeU=
```

d. Use the `vi` editor to edit the `ipsec.conf` file.

```
[vpn-host1]# vi ipsec.conf
```

e. Paste the contents of the buffer at the end of the file.

- Position you cursor on the last line of the file.
- Press the lowercase letter `o` key to get into insert mode and open a blank line at the end of the file.
- Select “Edit > Paste” from the terminal window menu to past the contents of the buffer into the file.
- Press “Esc” to exit insert mode.

- Save and close the `ipsec.conf` file.

f. On **vpn-host2**, use the `ipsec showhostkey` to display host key on right host, **vpn-host2**.

- The sample output is shown.

```
[vpn-host2]# ipsec showhostkey --right
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"
ipsec showhostkey loading secrets from
"/etc/ipsec.d/www.example.com.secrets"
ipsec showhostkey loaded private key for keyid:
PPK_RSA:AQPXXwWB4
        # rsakey AQPXXwWB4
rightrsasigkey=0sAQPXXwWB4r62JUqcItOtIps5GIkOxOe0n51jZ/09Sra5Qth
hlc0WaapVjycZIgDj3tVE4h/UCpBGZbE1MZ7u8DRZjrcv3aXF2CSESJcW8w0hoOD
9SUh3ZvDt1OE5bBWtM7moeJ2iY9rM0OqigRfIMeMKw0ZFdglxGGmuvfWtJrD886c
GYUFTP3K3+1zblg9vlcoOGdfb5jy03jAHgBC2waC1YYAZFQOcHp9XBGVzPq8VkXZ
AnECA8VtPuyExBXt/GBGUgJOdrLjG/HHtweLlqgB3hmy5NZhYiyS8UVpC7RBLpWG
OotjmM2dupw+voGP38bWy8K51T8wfRQbfsbUd84Ga6R7676ZKSZXBSMyDsLrsWl6
e1tf9sShJ9E6YZ3ZqSt1FsR8zMlArQhE2gfp+InlQAp1Q7v8TUODy0z1bih407o0
nsYGFXwB9izXGNGrvxoKgvzgleRj7ROP6DAls/8aXdir0N0que975Rc01YM2o0sj
nWwQq124YvenLn1RCbH5fq5NF6V29U7+B5q/2afL6hCvfmQ==
```

g. Select the `rightrsasigkey=<string>` and copy it into the buffer.

- Highlight the string as shown.
- With the string highlighted, select “Edit > Copy” from the terminal window menu.

```
[vpn-host2]# ipsec showhostkey --right
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"
ipsec showhostkey loading secrets from
"/etc/ipsec.d/www.example.com.secrets"
ipsec showhostkey loaded private key for keyid:
PPK_RSA:AQPXXwWB4
        # rsakey AQPXXwWB4
rightrsasigkey=0sAQPXXwWB4r62JUqcItOtIps5GIkOxOe0n51jZ/09Sra5Qth
hlc0WaapVjycZIgDj3tVE4h/UCpBGZbE1MZ7u8DRZjrcv3aXF2CSESJcW8w0hoOD
9SUh3ZvDt1OE5bBWtM7moeJ2iY9rM0OqigRfIMeMKw0ZFdglxGGmuvfWtJrD886c
GYUFTP3K3+1zblg9vlcoOGdfb5jy03jAHgBC2waC1YYAZFQOcHp9XBGVzPq8VkXZ
AnECA8VtPuyExBXt/GBGUgJOdrLjG/HHtweLlqgB3hmy5NZhYiyS8UVpC7RBLpWG
OotjmM2dupw+voGP38bWy8K51T8wfRQbfsbUd84Ga6R7676ZKSZXBSMyDsLrsWl6
e1tf9sShJ9E6YZ3ZqSt1FsR8zMlArQhE2gfp+InlQAp1Q7v8TUODy0z1bih407o0
nsYGFXwB9izXGNGrvxoKgvzgleRj7ROP6DAls/8aXdir0N0que975Rc01YM2o0sj
nWwQq124YvenLn1RCbH5fq5NF6V29U7+B5q/2afL6hCvfmQ==
```

h. Use the `vi` editor to create a temporary file named `right`.

- You are going to paste the contents of the buffer into this temporary file and then append the contents to the `/etc/ipsec.conf` file on **vpn-host1**.

```
[vpn-host2]# vi right
```

i. Paste the contents of the buffer at the end of the file.

- Press the lowercase letter `i` key to get into insert mode.

- Select “Edit > Paste” from the terminal window menu to past the contents of the buffer into the file.
- Press “Esc” to exit insert mode.
- Save and close the `right` file.

j. From **vpn-host2**, use the `sftp` command to copy the right file to **vpn-host1**.
- Include the IP address of **vpn-host1**, not the host name, as an argument.
- The systems are not configured to resolve host names.
- Answer `yes` when prompted.
- The `root` user’s password is `oracle`.

```
[vpn-host2]# sftp 192.0.2.111
The authenticity of host ‘192.0.2.111 (192.0.2.111)’ can’t be
established.
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added ‘192.0.2.111’ (ECDSA) to the list ...
Root!192.0.2.111’s password: oracle
Connected to 192.0.2.111.
sftp>
```

k. From the `sftp>` prompt, use the `put` command to copy the `right` file.
- After the copy is complete, use the `quit` command to exit `sftp`.

```
sftp> put right
Uploading right to /root/right ...
sftp> quit
```

l. From **vpn-host1**, use the `cat` command to concatenate the `/etc/ipsec.conf` file and the `/root/right` file into a single file.
- Use the `mv` command to rename `/etc/ipsec.conf` before issuing the `cat` command.
- The example assumes you are still in the `/etc` directory.

```
[vpn-host1]# mv ipsec.conf ipsec.BAK
[vpn-host1]# cat ipsec.BAK /root/right > ipsec.conf
```

m. Use the `cat` command to view the updated `ipsec.conf` file.
- The file now includes the “`leftrsasigkey=`” string and the “`rightrsasigkey=`” string at the end of the file.

- The sample output is shown.

```
[vpn-host1]# cat ipsec.conf
...
leftrsasigkey=0sAQOuaErmqqXZqWP/5tXPI2xXqR/qq8TPyGUnoUQ+rCkHy+WK
q14MrCcmPaHDVZfMIoRAN4Mot2k2535sHnc+SkWxaDyjueGKczTndALmck0eXXWa
WgcfNS94rH9wtleQuZXmTlnSQvW8kiHO1N1o22NrCRYZF8zrpQTNFC1WNAiO2qxW
ZSgdJn2q9iW6MFq0804AsNKI9QrrpC1n7xXyDrWhi+v5B73C0ly4/uYeNIotyK9C
ImM713QK3MUpZOSNnRiACIQYw8aX+YEKSgjPU3+nEHp243QeUVraIf5LE0cKtTQu
S3Ur1cgZfQZCFX1rGyHqD/ZtUyzL9Fvo5j04kjnZgJTywr4f0Tmw7a+2QJPIQQ52
iOv1jnV5WzbKB2zpDICsCzRZ7yVaK7MXrDxvbNss8gjXjK5BXgFLcVlFh/eJgcji
/AUK0S1vqXdYiJjWtZpjznRTDyE7+jqgLsSi0jY5y7i4dYhD+I0RujzTuv6z7ObD
+yLYpa/DoXQFMrFjB3kz9L+uqz7TtmwCthNdCJVJjnKL0jBIZ7IfVqBvIJoS5nra
WYbF/thUq7C6ziHML8AL2tUcx5wIne28ijJOT2LfjeU=

rightrsasigkey=0sAQPXXwWB4r62JUqcItOtIps5GIkOxOe0n51jZ/09Sra5Qth
hlc0WaapVjycZIgDj3tVE4h/UCpBGZbE1MZ7u8DRZjrcv3aXF2CSESJcW8w0hoOD
9SUh3ZvDt1OE5bBWtM7moeJ2iY9rM0OqigRfIMeMKw0ZFdglxGGmuvfWtJrD886c
GYUFTP3K3+1zblg9vlcoOGdfb5jy03jAHgBC2waC1YYAZFQOcHp9XBGVzPq8VkXZ
AnECA8VtPuyExBXt/GBGUgJOdrLjG/HHtweLlqgB3hmy5NZhYiyS8UVpC7RBLpWG
OotjmM2dupw+voGP38bWy8K51T8wfRQbfsbUd84Ga6R7676ZKSZXBSMyDsLrsWl6
e1tf9sShJ9E6YZ3ZqSt1FsR8zMlArQhE2gfp+InlQAp1Q7v8TUODy0z1bih407o0
nsYGFXwB9izXGNGrvxoKgvzgleRj7ROP6DAls/8aXdir0N0que975Rc01YM2o0sj
nWwQq124YvenLn1RCbH5fq5NF6V29U7+B5q/2afL6hCvfmQ==
```

8. Complete the "sitetosite" connection configuration in the /etc/ipsec.conf file on **vpn-host1**.
   a. Use the vi editor to edit /etc/ipsec.conf and add the "conn sitetosite" parameter and the "left" IP address information before the "leftrsasigkey=" line.
      - Indent all lines that start with "left", including the "leftrsasigkey=" line.
      - Do not exit the vi editor until step 9d.

```
[vpn-host1]# vi /etc/ipsec.conf
...
#include /etc/ipsec/d/*.conf
conn sitetosite
    leftid=192.168.1.101
    left=192.168.1.101
    leftsourceip=192.168.2.101
    leftsubnet=192.168.2.0/24
    leftrsasigkey=...
...
```

   b. Add the "right" IP address information after the "leftrsasigkey=" line and before the "rightrsasigkey=" line in the /etc/ipsec.conf file on **vpn-host1**.
      - Indent all lines that start with "right", including the "rightrsasigkey=" line.

```
...
    leftrsasigkey=...
    rightid=192.168.1.102
```

```
    right=192.168.1.102
    rightsourceip=192.168.3.102
    rightsubnet=192.168.3.0/24
    rightrsasigkey=
...
```

c. Add the following two lines at the end of the `/etc/ipsec.conf` file on **vpn-host1**.

```
...
    authby=rsasig
    auto=start
```

d. Save the changes made to the `/etc/ipsec.conf` file and exit the `vi` editor.

9. Check the syntax of the `/etc/ipsec.conf` file on **vpn-host1**.
   - If any errors are returned, the line number is included. Use the `vi` editor and make the necessary corrections to the file.
   - In this example, no errors are returned and the syntax is correct.

```
[vpn-host1]# /usr/libexec/ipsec/addconn --config /etc/ipsec.conf
--checkconfig
```

10. Use the `sftp` command to copy the `/etc/ipsec.conf` file from **vpn-host1** to **vpn-host2**.
    - Include the IP address of **vpn-host2**, not the host name, as an argument.
    - Answer `yes` when prompted.
    - The `root` user's password is `oracle`.

```
[vpn-host1]# sftp 192.0.2.112
The authenticity of host '192.0.2.112 (192.0.2.112)' can't be
established.
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.0.2.112' (ECDSA) to the list ...
root@192.0.2.111's password: oracle
Connected to 192.0.2.112.
sftp>
```

e. From the `sftp>` prompt, use the `put` command to copy the `/etc/ipsec.conf` file to `/etc/ipsec.conf` on **vpn-host2**.
   - After the copy is complete, use the `quit` command to exit `sftp`.

```
sftp> put /etc/ipsec.conf /etc/ipsec.conf
Uploading /etc/ipsec.conf to /etc/ipsec.conf ...
sftp> quit
```

11. Enable IP forwarding on both **vpn-host1** and **vpn-host2**.

a. On **vpn-host1**, use the `sysctl -w` command to enable IP forwarding.

```
[vpn-host1]# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

b. On **vpn-host2**, use the `sysctl -w` command to enable IP forwarding.

```
[vpn-host2]# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

12. Stop the `firewalld` service on both **vpn-host1** and **vpn-host2**.
    - You could add rules to trust the `ipsec` protocols. `libreswan` requires the firewall to allow the following packets:
      - UDP port `500` for the IKE protocol
      - UDP port `4500` for IKE NAT-Traversal
      - Protocol `50` for ESP IPSec packets
      - Protocol `51` for AH IPSec packets
    - For purposes of this exercise you stop the `firewalld` service.

    a. On **vpn-host1**, use the `systemctl` command to stop the `firewalld` service.

```
[vpn-host1]# systemctl stop firewalld
```

    b. On **vpn-host2**, use the `systemctl` command to stop the `firewalld` service.

```
[vpn-host2]# systemctl stop firewalld
```

13. Test connectivity before starting the `ipsec` service.

    a. On **vpn-host1**, use the `netstat -rn` command to view the route table.
      - Note that there is no route to the `192.168.3.0` subnet.

```
[vpn-host1]# netstat -rn
Destination  Gateway    ...  Iface
0.0.0.0      192.0.2.1  ...  eth0
192.0.2.0    0.0.0.0    ...  eth0
192.168.1.0  0.0.0.0    ...  eth1
192.168.2.0  0.0.0.0    ...  eth2
```

    b. From **vpn-host1**, use the `ping` command to test connectivity to `192.168.3.102`.
      - Note that you cannot `ping` this address.

- Press CTRL-C to kill the `ping` command.

```
[vpn-host1]# ping 192.168.3.102
PING 192.168.3.102 (192.168.3.102) 56(84) bytes of data.
CTRL-c
```

c. On **vpn-host2**, use the `netstat -rn` command to view the route table.

- Note that there is no route to the `192.168.2.0` subnet.

```
[vpn-host2]# netstat -rn
Destination  Gateway   ...  Iface
0.0.0.0      192.0.2.1 ...  eth0
192.0.2.0    0.0.0.0   ...  eth0
192.168.1.0  0.0.0.0   ...  eth1
192.168.3.0  0.0.0.0   ...  eth2
```

d. From **vpn-host2**, use the `ping` command to test connectivity to `192.168.2.101`.

- Note that you cannot `ping` this address.
- Press CTRL-C to kill the `ping` command.

```
[vpn-host2]# ping 192.168.2.101
PING 192.168.2.101 (192.168.2.101) 56(84) bytes of data.
CTRL-c
```

14. Start the `ipsec` service on both **vpn-host1** and **vpn-host2**.

a. On **vpn-host1**, use the `systemctl` command to start the `ipsec` service.

```
[vpn-host1]# systemctl start ipsec
```

b. On **vpn-host2**, use the `systemctl` command to start the `ipsec` service.

```
[vpn-host1]# systemctl start ipsec
```

15. Test connectivity after starting the `ipsec` service.

a. On **vpn-host1**, use the `netstat -rn` command to view the route table.

- Note that now there is a route to the `192.168.3.0` subnet.

```
[vpn-host1]# netstat -rn
Destination  Gateway   ...  Iface
0.0.0.0      192.0.2.1 ...  eth0
192.0.2.0    0.0.0.0   ...  eth0
192.168.1.0  0.0.0.0   ...  eth1
192.168.2.0  0.0.0.0   ...  eth2
192.168.3.0  0.0.0.0   ...  eth1
```

b. From **vpn-host1**, use the `ping` command to test connectivity to `192.168.3.102`.

- Note that now you can `ping` this address.

- Press CTRL-C to kill the `ping` command.

```
[vpn-host1]# ping 192.168.3.102
PING 192.168.3.102 (192.168.3.102) 56(84) bytes of data.
64 bytes from 192.168.3.102: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.3.102: icmp_seq=2 ttl=64 time=...
64 bytes from 192.168.3.102: icmp_seq=3 ttl=64 time=...
CTRL-c
```

c. On **vpn-host2**, use the `netstat -rn` command to view the route table.

- Note that now there is a route to the `192.168.2.0` subnet.

```
[vpn-host2]# netstat -rn
Destination  Gateway   ...  Iface
0.0.0.0      192.0.2.1 ...  eth0
192.0.2.0    0.0.0.0   ...  eth0
192.168.1.0  0.0.0.0   ...  eth1
192.168.2.0  0.0.0.0   ...  eth1
192.168.3.0  0.0.0.0   ...  eth2
```

d. From **vpn-host2**, use the `ping` command to test connectivity to `192.168.2.101`.

- Note that now you can `ping` this address.
- Press CTRL-C to kill the `ping` command.

```
[vpn-host2]# ping 192.168.2.101
PING 192.168.2.101 (192.168.2.101) 56(84) bytes of data.
64 bytes from 192.168.2.101: icmp_seq=1 ttl=64 time=...
64 bytes from 192.168.2.101: icmp_seq=2 ttl=64 time=...
64 bytes from 192.168.2.101: icmp_seq=3 ttl=64 time=...
CTRL-c
```

e. From **vpn-host2**, use the `ipsec auto --status` command to view current connection status.

- Note the ESP algorithms supported.
- Note the IKE algorithms supported.
- Note the Connection list, "`sitetosite`".
- Note the Total IPSec connections: 1 loaded, 1 active.

```
[vpn-host2]# ipsec auto --status
000 using kernel interface: netkey
...
000 config setup options:
000
000 configdir=/etc, configfile=/etc/ipsec.conf, secrets=/etc...
...
ESP algorithms supported:
000
000 algorithm ESP encrypt: id=3, name=ESP_3DES, ivlen=8, ...
```

```
000 algorithm ESP encrypt: id=6, name=ESP_CAST, ivlen=8, ...
...
000 algorithm AH/ESP auth: id=1, name=AUTH_ALGORITHM_HMAC_MD5...
000 algorithm AH/ESP auth: id=2, name=AUTH_ALGORITHM_HMAC_SHA...
...
IKE algorithms supported:
000
000 algorithm IKE encrypt: v1id=0, v1name=0??, v2id=16, v2nam...
000 algorithm IKE encrypt: v1id=0, v1name=0??, v2id=15, v2nam...
...
000 algorithm IKE hash: id=1, name=OAKLEY_MD5, hashlen=16
000 algorithm IKE hash: id=2, name=OAKLEY_SHA1, hashlen=20
...
000 Connection list:
000
000 "sitetosite":
192.168.3.0/24===192.168.1.102<192.168.1.102>[92.168.1.102]...19
2.168.1.101<192.168.1.101>===192.168.2.0/24; eroute owner: #4
...
000 Total IPSec connections: loaded 1, active 1
000
000 State list:
000
000 #4: "sitetosite":500 STATE_QUICK_R2 (IPSec SA established...
...
```

16. Shut down the **vpn-host1** and **vpn-host2** virtual machines.
    a. From **vpn-host2**, use the `systemctl poweroff` command to shut down **vpn-host2**.

```
[vpn-host2]# systemctl poweroff
```

    b. From **vpn-host1**, use the `systemctl poweroff` command to shut down **vpn-host1**.

```
[vpn-host1]# systemctl poweroff
```

Do not start the **host01**, **host02**, and **host03** virtual machines at this time.