



Integrated Cloud Applications & Platform Services



Oracle Linux System Administration II

Activity Guide

D103155GC10 | D106227

Learn more from Oracle University at education.oracle.com

ORACLE®

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Table of Contents

Practices for Lesson 1: Course Introduction	7
Course Practice Environment: Security Credentials.....	8
Practice 1-1: Exploring the gateway environment	10
Practice 1-2: Exploring the host01 VM.....	15
Practice 1-3: Exploring the host02 VM.....	19
Practice 1-4: Exploring the host03 VM.....	23
Practice 1-5: Exploring the host04 VM.....	27
Practice 1-6: Exploring the host05 VM.....	31
Practice 1-7: Exploring the host06 VM.....	34
Practice 1-8: Exploring the vpn-host1 VM.....	37
Practice 1-9: Exploring the vpn-host2 VM.....	41
Practice 1-10: Logging Off from Your System.....	45
Practices for Lesson 2: Network Addressing and Name Services.....	47
Practices for Lesson 2: Network Addressing and Name Services	48
Practice 2-1: Configuring a DHCP Server.....	49
Practice 2-2: Configuring a DHCP Client	52
Practice 2-3: Viewing and Testing the DNS Configuration	55
Practice 2-4: Configuring a Caching-Only Nameserver.....	62
Practices for Lesson 3: Authentication and Directory Services	69
Practices for Lesson 3: Authentication and Directory Services.....	70
Practice 3-1: Configuring an OpenLDAP Server	71
Practice 3-2: Implementing OpenLDAP Authentication	86
Practice 3-3: Authenticating from an OpenLDAP Client	92
Practices for Lesson 4: Pluggable Authentication Modules (PAM)	101
Practices for Lesson 4: Overview	102
Practice 4-1: Configuring PAM for a Single Login Session.....	103
Practice 4-2: Configuring PAM to Prevent Non-root Login	108
Practices for Lesson 5: Web and Email Services	111
Practices for Lesson 5: Web and Email Services.....	112
Practice 5-1: Configuring the Apache Web Server.....	113
Practices for Lesson 6: Installing Oracle Linux 7 by Using PXE and Kickstart.....	121
Practices for Lesson 6: Installing Oracle Linux 7 by Using PXE and 6	122
Practice 6-1: Creating a Kickstart File	123
Practice 6-3: Video of Performing a PXE/Kickstart Installation	133
Practices for Lesson 7: Samba Services	135

Practices for Lesson 7: Overview	136
Practice 7-1: Configuring a Samba Server.....	137
Practice 7-2: Accessing Samba Shares from a Client Host.....	143
Practice 7-3: Accessing a Linux Samba Share from a Windows System.....	147
Practices for Lesson 8: Advanced Software Package Management.....	151
Practices for Lesson 8: Overview	152
Practice 8-1: Exploring the host04 VM.....	153
Practice 8-3: Using Yum Utilities	162
Practice 8-4: Creating a binary RPM Package.....	177
Practice 8-5: Managing Software Updates with PackageKit.....	185
Practice 8-6: Working with Yum History and Yum Cache	192
Practices for Lesson 9: Advanced Networking.....	199
Practices for Lesson 9: Overview	200
Practice 9-2: Configuring Network Bonding from the Command Line	219
Practice 9-3: Working with Bonded Interfaces	224
Practice 9-4: Configuring 802.1Q VLAN Tagging by Using the GUI	238
Practice 9-5: Configuring 802.1Q VLAN Tagging from the Command Line.....	247
Practice 9-6: Working with VLAN Interfaces	250
Practice 9-7: Configuring a Site-to-Site VPN	260
Practices for Lesson 10: XFS File System	273
Practices for Lesson 10: Overview	274
Practice 10-1: Creating an XFS File System.....	275
Practice 10-2: Setting Disk Quotas on an XFS File System	280
Practice 10-3: Backing Up and Restoring XFS File Systems.....	286
Practices for Lesson 11: Btrfs File System.....	299
Practices for Lesson 11: Overview	300
Practice 11-1: Creating a Btrfs File System	301
Practice 11-2: Working with Subvolumes and Snapshots	312
Practice 11-3: Recovering from Data Corruption	319
Practices for Lesson 12: Storage Administration.....	323
Practices for Lesson 12: Overview	324
Practice 12-1: Creating Linux LVM Partitions	325
Practice 12-2: Creating a Logical Volume.....	329
Practice 12-3: Creating a File System and Mounting a Logical Volume.....	334
Practice 12-4: Backing Up Volume Group Metadata.....	336
Practice 12-5: Creating a Logical Volume Snapshot	340
Practice 12-6: Increasing the Capacity of a Logical Volume.....	344
Practice 12-7: Restoring Volume Group Metadata.....	349
Practice 12-8: Creating a Thinly Provisioned Logical Volume	351

Practice 12-9: Using Snapper with LVM Thin Provisioned Logical Volumes.....	356
Practice 12-10: Creating a RAID Device.....	361
Practice 12-11: Removing Partitions	366
Practices for Lesson 13: Advanced Storage Administration	371
Practices for Lesson 13: Overview	372
Practice 13-1: Creating and Mounting a File System	373
Practice 13-2: Setting Disk Quotas.....	379
Practice 13-3: Exploring and Configuring Udev Rules.....	385
Practice 13-4: Using kpartx	390
Practices for Lesson 14: File Sharing	395
Practices for Lesson 14: Overview	396
Practice 14-1: Configuring an NFS Server and an NFS Client	397
Practice 14-2: Removing the NFS Configuration	405
Practice 14-3: Configuring an FTP Server	408
Practice 14-4: Downloading a File from an FTP Server	410
Practice 14-5: Restoring VMs to Original Configuration	413
Practices for Lesson 15: Kerberos and IPA Services.....	415
Practices for Lesson 15: Overview	416
Practice 15-1: Kerberos Authentication	417
Practice 15-2: Installing Identity Management	422
Practice 15-3: Configuring SSSD Services	435

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Practices for Lesson 1: Course Introduction

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
host04/OS	root	oracle
host04/OS	oracle	oracle
host05/OS	root	oracle
host05/OS	oracle	oracle
host06/OS	root	oracle
host06/OS	oracle	oracle
vpn-host1/OS	root	oracle
vpn-host1/OS	oracle	oracle
vpn-host2/OS	root	oracle
vpn-host2/OS	oracle	oracle

Practices for Lesson 1: Overview

Practices Overview

In these practices, you will:

- Log in to your classroom **gateway** system and become familiar with the configuration
- Connect to the virtual machines used for the hands-on practices and become familiar with the VM guest configurations

Practice 1-1: Exploring the gateway environment

Overview

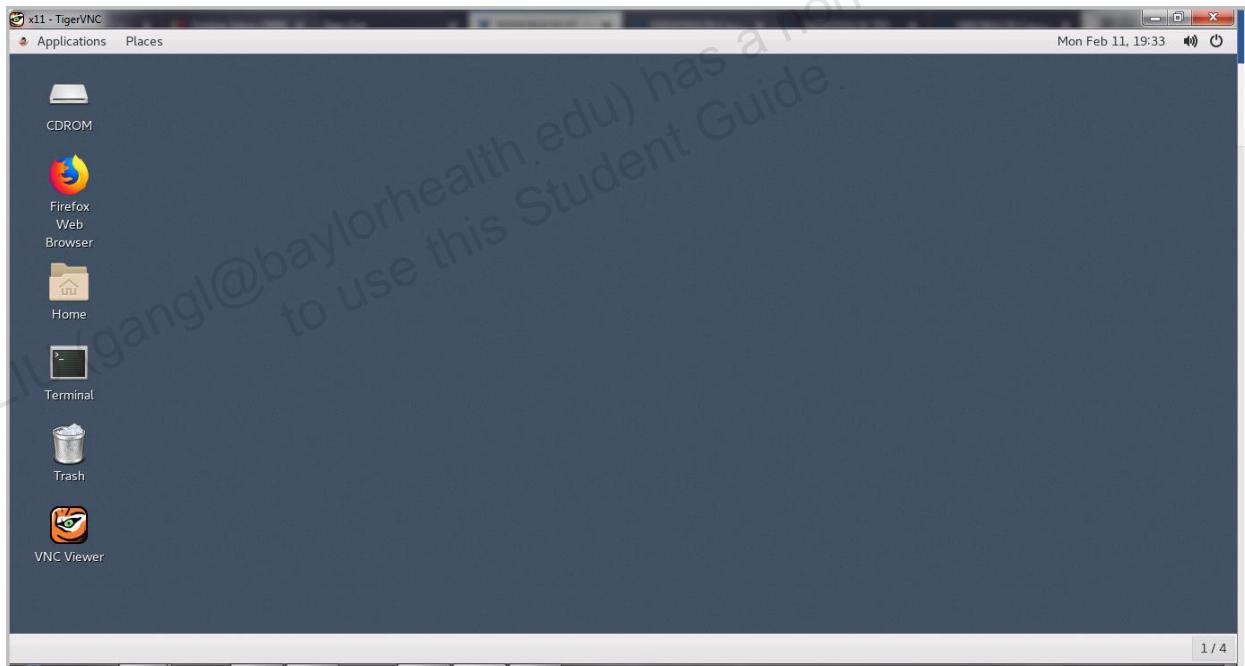
In this practice, you explore the **gateway** configuration and directory structure.

Assumptions

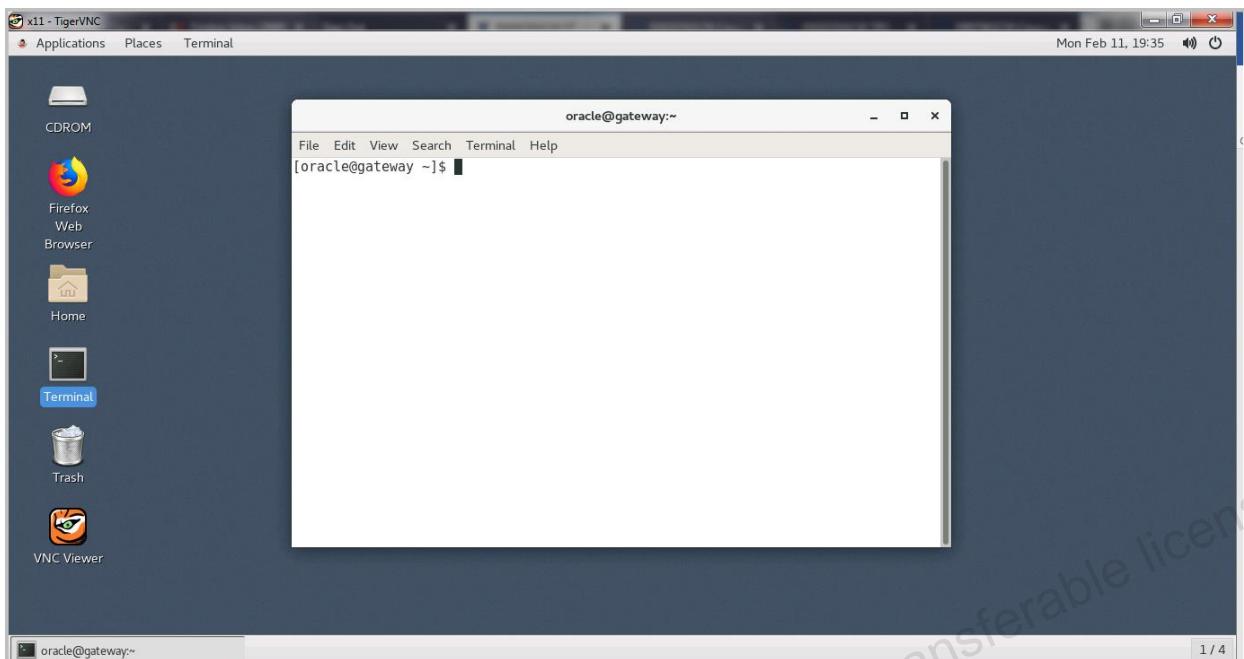
- Your instructor has assigned a **gateway** IP address to you.
- You are logged in to your **gateway** as `oracle` user.
- There are eight guests (virtual machines): **host01**, **host02**, **host03**, **host04**, **host05**, **host06**, **vpn-host1**, and **vpn-host2**.
- All guest VMs are running and have Oracle Linux 7.5 installed.

Tasks

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



2. Become the `root` user.
 - Double-click the **Terminal** icon on the desktop.
 - A terminal window opens as shown.



- Become the `root` user by using the `su -` command in the terminal window.
- Confirm that you are `root` by printing the user identity with the `whoami` command:

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

3. View the operating system that is running on **gateway**.

- a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@gateway ~]# cat /etc/oracle-release  
Oracle Linux Server release 7.5
```

- b. Use the `uname -a` command to print system information.

```
[root@gateway ~]# uname -a  
Linux gateway.example.com 4.14.35-1818.2.1.el7uek.x86_64 #2 SMP  
Mon Aug 27 21:16:31 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.14.35-1818.2.1.el7uek.x86_64.
- The host name is `gateway.example.com`.

4. View the network configuration of **gateway**.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.

```
[root@gateway ~]# ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo
```

```

...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether aa:bb:00:01:80:35 brd ff:ff:ff:ff:ff:ff
    inet 10.237.16.56/23 brd 10.237.17.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:21:f6:dc:00:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.0.2.255 scope global eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:21:f6:dc:00:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth2
...
5: eth3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:21:f6:dc:00:03 brd ff:ff:ff:ff:ff:ff
6: eth4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:21:f6:dc:00:04 brd ff:ff:ff:ff:ff:ff

```

- The address shown for eth0 is different for your system.
- The eth1 interface has an IP address of 192.0.2.1.
- The eth2 interface has an IP address of 192.168.1.1.

b. Verify the /etc/hosts file.

```

[root@gateway ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
192.0.2.1      gateway.example.com      gateway
192.0.2.101    host01.example.com     host01
192.0.2.102    host02.example.com     host02
192.0.2.103    host03.example.com     host03
192.0.2.104    host04.example.com     host04
192.0.2.105    host05.example.com     host05
192.0.2.106    host06.example.local  host06
192.0.2.107    vpn-host1.example.com  vpn-host1
192.0.2.108    vpn-host2.example.com  vpn-host2
10.237.16.56   eg18035

```

- The entry with IP address 10.237.16.56 and hostname eg18035 is different for each system.

c. Verify the /etc/resolv.conf file.

```
[root@gateway ~]# cat /etc/resolv.conf
search edu.oracle.com
nameserver 10.237.18.52
nameserver 152.68.154.3
```

5. Explore the staged files and directories.

a. View the contents of the /stage directory.

- There are 10 entries (2 directories, 8 files) in the /stage directory.
- These files are used in various practices in this course.

```
[root@gateway ~]# ls -l /stage
total 21114884
-rw-r--r-- 1 root root      255 ... base.ldif
-rwxr-x--- 1 root root   1014436 ... btrfs-corrupt-block
-rw-r--r-- 1 root root      459 ... dhcpd.conf
-rw----- 1 root root      522 ... kdc.conf
-rw-r--r-- 1 root root      721 ... krb5.conf
drwxr-xr-x 4 root root    4096 ... local_repo
-rw-r--r-- 1 root root  4440719360 ... OracleLinux-R7-U5-Server-
x86_64-dvd.iso
drwxr-xr-x 2 root root    4096 ... PXE
-rw-r--r-- 1 root root    1350 ... system-auth
-rw-r--r-- 1 root root 17179869184 ... system.img
```

b. View the contents of the /stage/local_repo directory.

- This is the local Yum repository containing packages installed in various practices in this course.

```
[root@gateway ~]# ls -l /stage/local_repo/
total 28
drwxr-xr-x 2 root root  4096 ... repodata
drwxr-xr-x 2 root root 20480 ... rpms
```

c. View the contents of the /stage/PXE directory.

- These files are used in “Practices for Lesson 6 – Installing Oracle Linux with PXE and Kickstart” in this course.

```
[root@gateway ~]# ls -l /stage/PXE/
total 48
-rw-r--r-- 1 root root  113 ... default
-rw-r--r-- 1 root root  912 ... dhcpd.conf
-rw-r--r-- 1 root root 33855 ... httpd.conf
-rw----- 1 root root 1031 ... ks.cfg
```

6. View the services that are running.

a. Show the HTTP service status.

```
[root@gateway ~]# systemctl status httpd
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service;
             enabled; vendor preset: disabled)
     Active: active (running) since ...
             ...

```

b. Show the DNS service status.

```
[root@gateway ~]# systemctl status named
named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service;
             enabled; vendor preset: disabled)
     Active: active (running) since ...
             ...

```

Practice 1-2: Exploring the host01 VM

Overview

In this practice, you perform the following:

- Log in to **host01**.
- View the **host01** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Log on to 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 `hostname` command confirms you have successfully logged in to **host01**.

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

[root@host01 ~]# hostname
host01.example.com
```

2. View the operating system that is running on **host01**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@host01 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@host01 ~]# uname -a
Linux host01.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `host01.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 4 disks (xvda, xvdb, xvdc, xvdd).

```
[root@host01 ~]# lsblk -o name,fstype,size,type,mountpoint,label
  NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
  xvda          20G disk
  |-xvda1      xfs       1G part /boot
  |-xvda2    LVM2_member 19G part
    |-ol-root  xfs       17G lvm   /
    |-ol-swap   swap      2G lvm   [SWAP]
  xvdb        swap       4G disk
  xvdc        swap       20G disk
  xvdd        swap       20G disk
```

4. View the network configuration.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There are 4 NICs (eth0, eth1, eth2, eth3).
- Only `eth0` has an IP address (192.0.2.101).
- 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 9: Advanced Networking.”

```
[root@host01 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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
        noprefixroute eth0
    ...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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 pfifo_fast state UP group default qlen 1000
    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 pfifo_fast state UP group default qlen 1000
```

```
link/ether 00:16:3e:00:04:01 brd ff:ff:ff:ff:ff:ff
...
```

b. View the /etc/hosts file.

```
[root@host01 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain    localhost
192.0.2.1      gateway.example.com     gateway
192.0.2.101    host01.example.com     host01
192.0.2.102    host02.example.com     host02
192.0.2.103    host03.example.com     host03
192.0.2.104    host04.example.com     host04
192.0.2.105    host05.example.com     host05
192.0.2.106    host06.example.local   host06
192.0.2.107    vpn-host1.example.com  vpn-host1
192.0.2.108    vpn-host2.example.com  vpn-host2
```

c. Verify the /etc/resolv.conf file.

```
[root@host01 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

5. View the local Yum repository configuration.

- The local.repo file is configured as shown.
- The local_repo is the only enabled repository.
- The local_repo contains 288 files.

```
[root@host01 ~]# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[root@host01 ~]# grep enabled=1 /etc/yum.repos.d/*.repo
/etc/yum.repos.d/local.repo:enabled=1

[root@host01 ~]# yum repolist
Loaded plugins: langpacks, ulninfo
repo id                      repo name
status
local_repo                     "local_repo packages"
288
repolist: 288
```

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

```
[root@host01 ~]# exit
logout
Connection to host01 closed.
```

Practice 1-3: Exploring the host02 VM

Overview

In this practice, you perform the following:

- Log in to **host02**.
- View the **host02** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Log on to 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 `hostname` command confirms you have successfully logged in to **host02**.

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

[root@host02 ~]# hostname
host02.example.com
```

2. View the operating system that is running on **host02**.
 - a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@host02 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

- b. Use the `uname -a` command to print system information.

```
[root@host02 ~]# uname -a
Linux host02.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `host02.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 4 disks (xvda, xvdb, xvdc, xvdd).

```
[root@host02 ~]# lsblk -o name,fstype,size,type,mountpoint,label
NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
xvda
|-xvda1    xfs        20G disk
|-xvda2    LVM2_member 19G part
  |-ol-root xfs        17G lvm   /
  |-ol-swap  swap       2G lvm   [SWAP]
xvdb      swap        4G disk
xvdc
xvdd      swap        20G disk
```

4. View the network configuration.

- Use the `ip addr` command to display the network configuration.
 - Only partial output is shown.
 - There are 4 NICs (eth0, eth1, eth2, eth3).
 - eth0 has an IP address (192.0.2.102).
 - eth1 has an IP address (192.168.1.102).
 - The `virbr0` (virtual bridge) entries seen on hosts in this course helps communication from Dom0 to the VMs and can be ignored.

```
[root@host02 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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 noprefixroute eth0
    ...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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 noprefixroute eth1
    ...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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 pfifo_fast state UP group default qlen 1000
```

```
link/ether 00:16:3e:00:04:02 brd ff:ff:ff:ff:ff:ff
```

```
...
```

b. View the /etc/hosts file.

```
[root@host02 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain    localhost
192.0.2.1      gateway.example.com     gateway
192.0.2.101    host01.example.com     host01
192.0.2.102    host02.example.com     host02
192.0.2.103    host03.example.com     host03
192.0.2.104    host04.example.com     host04
192.0.2.105    host05.example.com     host05
192.0.2.106    host06.example.local   host06
192.0.2.107    vpn-host1.example.com  vpn-host1
192.0.2.108    vpn-host2.example.com  vpn-host2
```

c. Verify the /etc/resolv.conf file.

```
[root@host02 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

5. View the local Yum repository configuration.

- The local.repo file is configured as shown.
- The local_repo is the only enabled repository.
- The local_repo contains 288 files.

```
[root@host02 ~]# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[root@host02 ~]# grep enabled=1 /etc/yum.repos.d/*.repo
/etc/yum.repos.d/local.repo:enabled=1

[root@host02 ~]# yum repolist
Loaded plugins: langpacks, ulninfo
repo id                      repo name
status
local_repo                     "local_repo packages"
288
repolist: 288
```

6. Use the `exit` command to log off **host02**.

```
[root@host02 ~]# exit
logout
Connection to host02 closed.
```

Practice 1-4: Exploring the host03 VM

Overview

In this practice, you perform the following:

- Log in to **host03**.
- View the **host03** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

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

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

[root@host03 ~]# hostname
host03.example.com
```

2. View the operating system that is running on **host03**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@host03 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@host03 ~]# uname -a
Linux host03.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `host03.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 4 disks (xvda, xvdb, xvdc, xvdd).

```
[root@host03 ~]# lsblk -o name,fstype,size,type,mountpoint,label
  NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
  xvda
  | -xvda1    xfs        20G disk
  | -xvda2    LVM2_member 19G part
  |   |-ol-root xfs        17G lvm   /
  |   |-ol-swap swap       2G lvm   [SWAP]
  xvdb        swap       4G disk
  xvdc        swap       20G disk
  xvdd        swap       20G disk
```

4. View the network configuration.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There are 3 NICs (eth0, eth1, eth2).
- eth0 has IP address 192.0.2.103
- eth1 has IP address 192.0.2.113

```
[root@host03 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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
        noprefixroute eth0
    ...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.113/24 brd 192.0.2.255 scope global
        noprefixroute eth1
    ...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:00:03:03 brd ff:ff:ff:ff:ff:ff
    ...
```

- b. View the `/etc/hosts` file.

```
[root@host03 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain    localhost
192.0.2.1      gateway.example.com     gateway
192.0.2.101    host01.example.com      host01
192.0.2.102    host02.example.com      host02
192.0.2.103    host03.example.com      host03
192.0.2.104    host04.example.com      host04
192.0.2.105    host05.example.com      host05
192.0.2.106    host06.example.local   host06
192.0.2.107    vpn-host1.example.com  vpn-host1
192.0.2.108    vpn-host2.example.com  vpn-host2
```

c. Verify the /etc/resolv.conf file.

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

5. View the local Yum repository configuration.

- The local.repo file is configured as shown.
- The local_repo is the only enabled repository.
- The local_repo contains 288 files.

```
[root@host03 ~]# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[root@host03 ~]# grep enabled=1 /etc/yum.repos.d/*.repo
/etc/yum.repos.d/local.repo:enabled=1

[root@host03 ~]# yum repolist
Loaded plugins: langpacks, ulninfo
repo id                      repo name
status
local_repo                     "local_repo packages"
288
repolist: 288
```

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

```
[root@host03 ~]# exit
logout
Connection to host03 closed.
```

Practice 1-5: Exploring the host04 VM

Overview

In this practice, you perform the following:

- Log in to **host04**.
- View the **host04** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Explore the **host04** VM guest.

a. Use the `ssh` command to log in to **host04**.

- 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 `hostname` command confirms whether you have successfully logged in to **host04**.

```
[root@gateway ~]# ssh host04
The authenticity of host 'host04 (192.0.2.104)' can't be
established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host04,192.0.2.104' (ECDSA) to the
list of known hosts.
root@host04's password:

[root@host04 ~]# hostname
host04.example.com
```

2. View the operating system that is running on **host04**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@host04 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@host04 ~]# uname -a
Linux host04.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `host04.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 2 disks (xvda, xvdb).

```
[root@host04 ~]# lsblk -o name,fstype,size,type,mountpoint,label
NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
xvda            20G disk
|-xvda1        xfs       1G part /boot
|-xvda2    LVM2_member  19G part
  |-ol-root   xfs      17G lvm  /
  |-ol-swap    swap     2G lvm  [SWAP]
xvdb        swap      4G disk
```

4. View the network configuration.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There is 1 NIC (`eth0`).
- `eth0` has IP address 192.0.2.104

```
[root@host04 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:00:01:04 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.104/24 brd 192.0.2.255 scope global noprefixroute eth0
...
...
```

- b. View the `/etc/hosts` file.

```
[root@host04 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain    localhost
192.0.2.1      gateway.example.com      gateway
192.0.2.101    host01.example.com      host01
192.0.2.102    host02.example.com      host02
192.0.2.103    host03.example.com      host03
192.0.2.104    host04.example.com      host04
192.0.2.105    host05.example.com      host05
192.0.2.106    host06.example.local    host06
192.0.2.107    vpn-host1.example.com   vpn-host1
192.0.2.108    vpn-host2.example.com   vpn-host2
```

- c. Verify the `/etc/resolv.conf` file.

- This file includes the DNS information for the Oracle University environment.

```
[root@host04 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search edu.oracle.com example.com
nameserver 192.0.2.1
nameserver 10.237.18.52
nameserver 152.68.154.3
```

5. View the Oracle Linux Yum Server configuration.

- The HTTP proxy server variable is set in the /etc/yum.conf file.

- Entry is highlighted in **bold** font below.

```
[root@host04 ~]# cat /etc/yum.conf
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
...
proxy=http://ges-proxy.us.oracle.com:80
```

- Two Oracle Linux Yum repositories are enabled.

- Enabled repositories are shown in **bold** font.

```
[root@host04 ~]# cat /etc/yum.repos.d/public-yum-ol7.repo
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/latest/$base
arch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
...

[ol7_UEKR4]
name=Latest Unbreakable Enterprise Kernel Release 4 for Oracle
Linux $releasever ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/UEKR4/$basea
rch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
```

```
enabled=1  
...
```

- c. The local_repo is disabled.

```
[root@host04 ~]# cat /etc/yum.repos.d/local.repo  
[local_repo]  
Name="local_repo packages"  
baseurl=http://192.0.2.1/local_repo/  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle  
gpgcheck=1  
enabled=0
```

6. Use the exit command to log off host04.

```
[root@host04 ~]# exit  
logout  
Connection to host04 closed.
```

Practice 1-6: Exploring the host05 VM

Overview

In this practice, you perform the following:

- Log in to **host05**.
- View the **host05** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Explore the **host05** VM guest.

a. Use the `ssh` command to log in to **host05**.

- 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 `hostname` command confirms whether you have successfully logged in to **host05**.

```
[root@gateway ~]# ssh host05
The authenticity of host 'host05 (192.0.2.105)' can't be
established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host05,192.0.2.105' (ECDSA) to the
list of known hosts.
root@host05's password:

[root@host05 ~]# hostname
host05.example.com
```

2. View the operating system that is running on **host05**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@host05 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@host05 ~]# uname -a
Linux host05.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `host05.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 2 disks (xvda, xvdb).

```
[root@host05 ~]# lsblk -o name,fstype,size,type,mountpoint,label
NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
xvda            30G disk
└─xvda1      xfs       1G part /boot
└─xvda2    LVM2_member 29G part
  ├─ol-root  xfs       26G lvm   /
  └─ol-swap   swap      3G lvm   [SWAP]
xvdb            4G disk
```

4. View the network configuration.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There is 1 NIC (`eth0`).
- `eth0` has IP address 192.0.2.105

```
[root@host05 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:00:01:05 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.105/24 brd 192.0.2.255 scope global noprefixroute eth0
...
...
```

- b. View the `/etc/hosts` file.

```
[root@host05 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain    localhost
192.0.2.1      gateway.example.com     gateway
192.0.2.101    host01.example.com      host01
192.0.2.102    host02.example.com      host02
192.0.2.103    host03.example.com      host03
192.0.2.104    host04.example.com      host04
192.0.2.105    host05.example.com      host05
192.0.2.106    host06.example.local    host06
192.0.2.107    vpn-host1.example.com   vpn-host1
192.0.2.108    vpn-host2.example.com   vpn-host2
```

- c. Verify the `/etc/resolv.conf` file.

```
[root@host05 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

5. View the local Yum repository configuration.

- The `local.repo` file is configured as shown.
- The `local_repo` is the only enabled repository.
- The `local_repo` contains 288 files.

```
[root@host05 ~]# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[root@host05 ~]# grep enabled=1 /etc/yum.repos.d/*.repo
/etc/yum.repos.d/local.repo:enabled=1

[root@host05 ~]# yum repolist
Loaded plugins: langpacks, ulninfo
repo id                                repo name
status
local_repo                               "local_repo packages"
288
repolist: 288
```

6. Use the `exit` command to log off `host05`.

```
[root@host05 ~]# exit
logout
Connection to host05 closed.
```

Practice 1-7: Exploring the host06 VM

Overview

In this practice, you perform the following:

- Log in to **host06**.
- View the **host06** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Explore the **host06** VM guest.

a. Use the `ssh` command to log in to **host06**.

- 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 `hostname` command confirms whether you have successfully logged in to **host06**.

```
[root@gateway ~]# ssh host06
The authenticity of host 'host06 (192.0.2.106)' can't be
established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host06,192.0.2.106' (ECDSA) to the
list of known hosts.
root@host06's password:

[root@host06 ~]# hostname
host06.example.local
```

- The **host06** VM has a different domain for eliminating domain name conflict when configuring and installing the Identity Management server in Practice 15-2 “Installing Identity Management”.

2. View the operating system that is running on **host06**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@host06 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@host06 ~]# uname -a
Linux host06.example.local 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.

- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is host06.example.local.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 2 disks (xvda, xvdb).

```
[root@host06 ~]# lsblk -o name,fstype,size,type,mountpoint,label
NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
xvda          swap     30G disk
└─xvda1      xfs      1G part /boot
└─xvda2    LVM2_member 29G part
  ├─ol-root  xfs      26G lvm   /
  └─ol-swap   swap     3G lvm   [SWAP]
xvdb          swap     4G disk
```

4. View the network configuration.

a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There is 1 NIC (`eth0`).
- `eth0` has IP address 192.0.2.106

```
[root@host06 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:16:3e:00:01:06 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.106/24 brd 192.0.2.255 scope global
        noprefixroute eth0
    ...
...
```

b. View the `/etc/hosts` file.

- You are directed to configure this file in the Practices for Lesson 15: Kerberos and IPA Services.

```
[root@host06 ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
```

c. Verify the `/etc/resolv.conf` file.

- The search domain and nameserver for **host06** is different compared to other VM hosts in order to eliminate overlapping domains when installing the Identity Management server in Practice 15-2 “Installing Identity Management.”

```
[root@host06 ~]# cat /etc/resolv.conf
search example.local
nameserver 127.0.0.1
```

5. Show that all Yum repositories are disabled.

```
[root@host06 ~]# cd /etc/yum.repos.d/
[root@host06 yum.repos.d]# ls
local.repo  public-yum-ol7.repo

[root@host06 yum.repos.d]# grep enabled=1 *
[root@host06 yum.repos.d]#
```

- The “grep” command returns no output indicating there are no repositories enabled on **host06**.

6. Use the **exit** command to log off **host06**.

```
[root@host06 yum.repos.d]# exit
logout
Connection to host06 closed.
```

Practice 1-8: Exploring the vpn-host1 VM

Overview

In this practice, you perform the following:

- Log in to **vpn-host1**.
- View the **vpn-host1** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Explore the **vpn-host1** VM guest.

a. Use the `ssh` command to log in to **vpn-host1**.

- 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 `hostname` command confirms whether you have successfully logged in to **vpn-host1**.

```
[root@gateway ~]# ssh vpn-host1
The authenticity of host 'vpn-host1 (192.0.2.107)' can't be
established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vpn-host1,192.0.2.107' (ECDSA) to
the list of known hosts.
root@vpn-host1's password:

[root@vpn-host1 ~]# hostname
vpn-host1.example.com
```

2. View the operating system that is running on **vpn-host1**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@vpn-host1 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@vpn-host1 ~]# uname -a
Linux vpn-host1.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `vpn-host1.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 2 disks (xvda, xvdb).

```
[root@vpn-host1 ~]# lsblk -o
name,fstype,size,type,mountpoint,label
NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
xvda          swap      30G disk
└─xvda1      xfs       1G part /boot
└─xvda2    LVM2_member 29G part
  ├─ol-root   xfs      26G lvm  /
  └─ol-swap    swap      3G lvm  [SWAP]
xvdb          swap      4G disk
```

4. View the network configuration.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There are 3 NICs (eth0, eth1, eth2).
- eth0 has an IP address (192.0.2.107) and MAC (00:16:3e:01:01:01)
- eth1 has an IP address (192.168.1.101) and MAC (00:16:3e:01:02:01)
- eth2 has an IP address (192.168.2.101) and MAC (00:16:3e:01:03:01)

```
[root@vpn-host1 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
  link/ether 00:16:3e:01:01:01 brd ff:ff:ff:ff:ff:ff
  inet 192.0.2.107/24 brd 192.0.2.255 scope global
    noprefixroute eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
  link/ether 00:16:3e:01:02:01 brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.101/24 brd 192.168.1.255 scope global
    noprefixroute eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
  link/ether 00:16:3e:01:03:01 brd ff:ff:ff:ff:ff:ff
  inet 192.168.2.101/24 brd 192.168.2.255 scope global
    noprefixroute eth2
```

b. View the /etc/hosts file.

```
[root@vpn-host1 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain localhost
192.0.2.1      gateway.example.com      gateway
192.0.2.101    host01.example.com     host01
192.0.2.102    host02.example.com     host02
192.0.2.103    host03.example.com     host03
192.0.2.104    host04.example.com     host04
192.0.2.105    host05.example.com     host05
192.0.2.106    host06.example.local   host06
192.0.2.107    vpn-host1.example.com  vpn-host1
192.0.2.108    vpn-host2.example.com  vpn-host2
```

c. Verify the /etc/resolv.conf file.

```
[root@vpn-host1 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

5. View the local Yum repository configuration.

- The local.repo file is configured as shown.
- The local_repo is the only enabled repository.
- The local_repo contains 288 files.

```
[root@vpn-host1 ~]# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[root@vpn-host1 ~]# grep enabled=1 /etc/yum.repos.d/*.repo
/etc/yum.repos.d/local.repo:enabled=1

[root@vpn-host1 ~]# yum repolist
Loaded plugins: langpacks, ulninfo
repo id                      repo name
status
local_repo                     "local_repo packages"
288
repolist: 288
```

6. Use the `exit` command to log off **vpn-host1**.

```
[root@vpn-host1 ~]# exit
logout
Connection to vpn-host1 closed.
```

Practice 1-9: Exploring the vpn-host2 VM

Overview

In this practice, you perform the following:

- Log in to **vpn-host2**.
- View the **vpn-host2** configuration.

Assumptions

- You are logged on to **gateway** as the **root** user.

Tasks

1. Explore the **vpn-host2** VM guest.

a. Use the `ssh` command to log in to **vpn-host2**.

- 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 `hostname` command confirms whether you have successfully logged in to **vpn-host2**.

```
[root@gateway ~]# ssh vpn-host2
The authenticity of host 'vpn-host2 (192.0.2.108)' can't be
established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vpn-host2,192.0.2.108' (ECDSA) to
the list of known hosts.
root@vpn-host2's password:

[root@vpn-host2 ~]# hostname
vpn-host2.example.com
```

2. View the operating system that is running on **vpn-host2**.

a. Use the `cat` command to view the contents of the `/etc/oracle-release` file.

```
[root@vpn-host2 ~]# cat /etc/oracle-release
Oracle Linux Server release 7.5
```

b. Use the `uname -a` command to print system information.

```
[root@vpn-host2 ~]# uname -a
Linux vpn-host2.example.com 4.1.12-112.16.4.el7uek.x86_64 #2 SMP
Mon Mar 12 23:57:12 PDT 2018 x86_64 x86_64 x86_64 GNU/Linux
```

- The operating system Linux, specifically Oracle Linux 7 Update 5.
- The Linux kernel version is 4.1.12-112.16.4.el7uek.x86_64.
- The host name is `vpn-host2.example.com`.

3. View the disk configuration.

- Run the `lsblk` command with options shown.
- There are 2 disks (xvda, xvdb).

```
[root@vpn-host2 ~]# lsblk -o
name,fstype,size,type,mountpoint,label
NAME      FSTYPE      SIZE TYPE MOUNTPOINT LABEL
xvda          swap      30G disk
└─xvda1      xfs       1G part /boot
└─xvda2    LVM2_member 29G part
  ├─ol-root   xfs      26G lvm  /
  ├─ol-swap    swap      3G lvm  [SWAP]
xvdb          swap      4G disk
```

4. View the network configuration.

- a. Use the `ip addr` command to display the network configuration.

- Only partial output is shown.
- There must be 3 NICs (eth0, eth1, eth2).
- `eth0` has an IP address (192.0.2.108) and MAC (00:16:3e:01:01:02)
- `eth1` has an IP address (192.168.1.102) and MAC (00:16:3e:01:02:02)
- `eth2` has an IP address (192.168.3.102) and MAC (00:16:3e:01:03:02)

```
[root@vpn-host2 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:01:01:02 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.108/24 brd 192.0.2.255 scope global
        noprefixroute eth0
        ...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:01:02:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.102/24 brd 192.168.1.255 scope global
        noprefixroute eth1
        ...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:16:3e:01:03:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.3.102/24 brd 192.168.3.255 scope global
        noprefixroute eth2
```

...

b. View the /etc/hosts file.

```
[root@vpn-host2 ~]# cat /etc/hosts
127.0.0.1      localhost.localdomain localhost
192.0.2.1      gateway.example.com      gateway
192.0.2.101    host01.example.com     host01
192.0.2.102    host02.example.com     host02
192.0.2.103    host03.example.com     host03
192.0.2.104    host04.example.com     host04
192.0.2.105    host05.example.com     host05
192.0.2.106    host06.example.local   host06
192.0.2.107    vpn-host1.example.com  vpn-host1
192.0.2.108    vpn-host2.example.com  vpn-host2
```

c. Verify the /etc/resolv.conf file.

```
[root@vpn-host2 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

5. View the local Yum repository configuration.

- The local.repo file is configured as shown.
- The local_repo is the only enabled repository.
- The local_repo contains 288 files.

```
[root@vpn-host2 ~]# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[root@vpn-host2 ~]# grep enabled=1 /etc/yum.repos.d/*.repo
/etc/yum.repos.d/local.repo:enabled=1

[root@vpn-host2 ~]# yum repolist
Loaded plugins: langpacks, ulninfo
repo id                      repo name
status
local_repo                     "local_repo packages"
288
repolist: 288
```

6. Use the **exit** command to log off **vpn-host2**.

```
[root@vpn-host2 ~]# exit
logout
Connection to vpn-host2 closed.
```

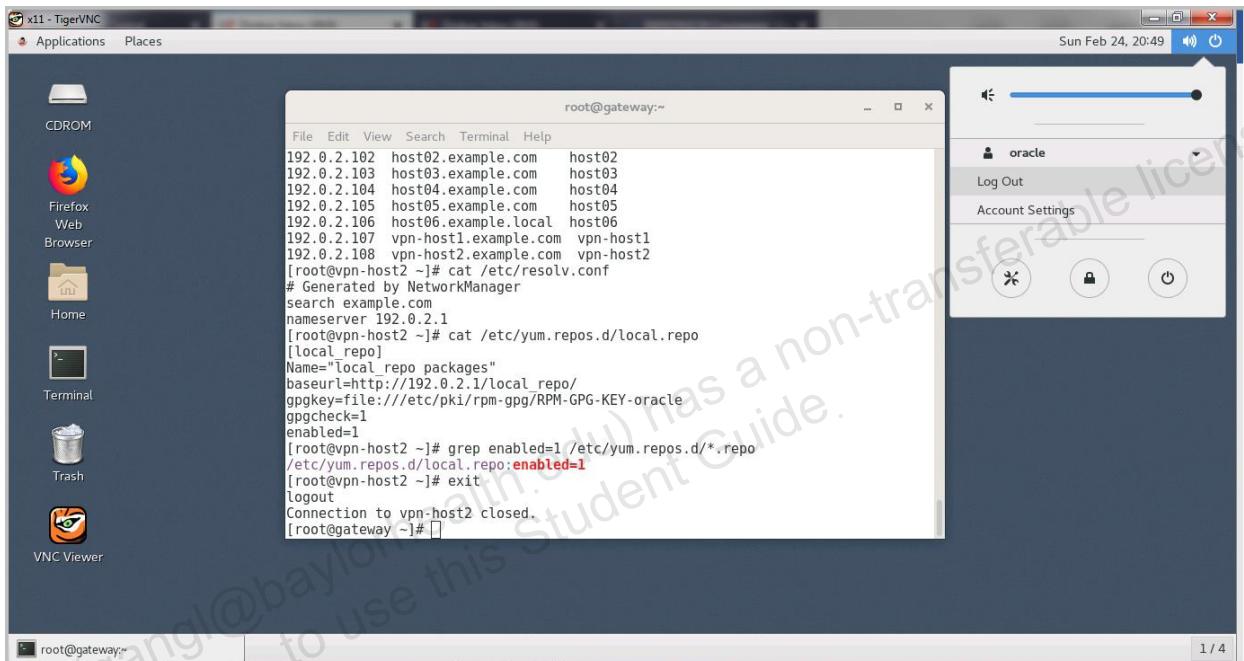
Practice 1-10: Logging Off from Your System

Overview

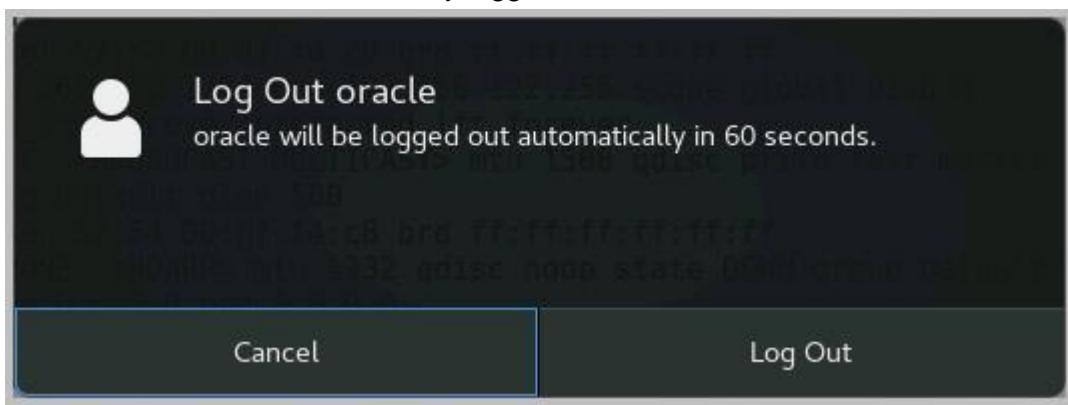
In this practice, you learn how to log off from your system.

Tasks

1. Learn how to log off your student system.
 - a. Open the menu in the upper right corner by clicking the power symbol.



- b. Select "Log Out."
 - 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.



GANG LIU (gangli@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Practices for Lesson 2: Network Addressing and Name Services

Practices for Lesson 2: Network Addressing and Name Services

Practices Overview

In these practices, you:

- Configure **host02** VM as a DHCP server and **host01** VM as a DHCP client
- Dynamically obtain an IP address for `eth1` on **host02**
- View and test the DNS server configuration on **gateway**
- Configure **host01** as a caching-only nameserver
- Test the DNS configuration

Practice 2-1: Configuring a DHCP Server

Overview

In this practice, you configure **host02** VM as a DHCP server.

Assumptions

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

Tasks

1. Log in to the **host02** VM guest.
 - a. Use the `ssh` command to log in to **host02**.


```
[gateway]# ssh host02
root@host02's password:
Last login: ...
[host02]#
```
2. Install the `dhcp` package on **host01** 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-common-4.2.5-68.0.1.el7.x86_64
dhcp-libs-4.2.5-68.0.1.el7.x86_64
```
 - 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           12:4.2.5-68.0.1.el7_5.1          local_repo
dhcp-common.x86_64     12:4.2.5-68.0.1.el7_5.1          local_repo
dhcp-libs.x86_64       12:4.2.5-68.0.1.el7_5.1         local_repo
```
 - c. Use the `yum` command to install the `dhcp` package.
 - Answer `y` when prompted “Is this ok.”

```
# yum install dhcp
...
Is this ok [y/d/N]: y
...
Complete!
```
3. Use the `vi` editor to edit `/etc/dhcp/dhcpd.conf` as follows:
 - This `dhcpd.conf` file also exists on **gateway** in the `/stage` directory. You can use the `scp` command to copy the file from `/stage` on **gateway** to the `/etc/dhcp/` directory on **host02**.

```
# 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 `eth1` 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 `eth1` 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 eth1
...
```

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 dhcpcd
Created symlink from /etc/systemd/system/multi-
user.target.wants/dhcpcd.service to
/etc/systemd/system/dhcpcd.service.
```

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

```
# systemctl start dhcpcd
```

- c. Use the `systemctl` command to view the status of the `dhcpd` service.
- Note that the server is only listening on `eth1`.

```
# systemctl status dhcpcd
dhcpcd.service - DHCPv4 Server Daemon
   Loaded: loaded (/etc/systemd/system/dhcpcd.service; enabled;
             vendor preset: disabled)
     Active: active (running) since ...
             ...
<date_time> host02....: Listening on LPF/eth1/00:16...
             ...
```

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 (**host02**) for the `eth1` network interface

You begin this practice by opening a second terminal window on **gateway** and logging in to **host01** as the `root` user. You are already logged in as the `root` user to **host02** from Practice 2-1.

Assumptions

- This practice is performed on **host01** and **host02** VMs.
- You are currently logged in to **host02** (from Practice 2-1).
- The prompts include either **host01** or **host02** to indicate which system to enter the command from.

Tasks

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

```
[oracle@gateway ~]$ su -  
Password:  
Last login: ...  
[root@gateway ~]#
```

- c. As the `root` user on **gateway**, use the `ssh` command to log in to **host01**.

```
[gateway]# ssh host01  
root@host01's password:  
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
...
DEVICE=eth1
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 state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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 noprefixroute eth0
    ...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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.

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

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

- 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 state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    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 noprefixroute eth0
    ...
3: eth1: <inet&ampgt mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    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
        valid_lft forever preferred_lft forever
    ...
...
```

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    state UP group default qlen 1000
        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 dynamic
            eth1
            ...

```

6. View information about the lease.

- 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;
    ...
}
```

- View information about the lease on the server (**host02**).

```
[host02]# 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.
[gateway]#
```

- In this window, you are logged in as the `root` user on **gateway**.
- 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 **gateway**
- Test the lookup functionality of DNS from **host02**

Assumptions

- **Gateway** is already configured as an authoritative nameserver for the example.com domain.
- This practice is performed on **gateway** and on **host02** VM.
- You are logged in as the `root` user on **gateway** from one terminal window.
- You are logged in as the `root` user on **host02** from a second terminal window.
- The prompts in the solution section include either **gateway** or **host02** to indicate which system to enter the command from.

Tasks

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

- In this example, the package is installed.

```
[gateway]# rpm -qa | grep bind
bind-libs-9.9.4-61.el7_5.1.x86_64
bind-libs-lite-9.9.4-61.el7_5.1.x86_64
keybinder3-0.3.0-1.el7.x86_64
bind-9.9.4-61.el7_5.1.x86_64
bind-license-9.9.4-61.el7_5.1.noarch
bind-utils-9.9.4-61.el7_5.1.x86_64
rpcbind-0.2.0-44.el7.x86_64
```

2. Ensure that the `named` service is enabled and running on **gateway**.

- Use the `systemctl` command to verify that the `named` service is started on **gateway**.
 - In this example, the service is enabled and active.

```
[gateway]# systemctl status named
   named.service - Berkeley Internet Name Domain (DNS)
       Loaded: loaded (/usr/lib/systemd/system/named.service;
                 enabled; vendor preset: disabled)
         Active: active (running) since ...
             ...
             ...
```

3. View the DNS configuration on **gateway**.

- 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.

```
[gateway]# cat /etc/named.conf
...
options {
...
    directory "/var/named";
...
zone "example.com" IN {
    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 VM guest systems.
 - The SMB-HOST03 (192.0.2.113) entry is for the “Practices for Lesson 7: Samba Services”.

```
[gateway]# 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
host05       A      192.0.2.105
host06       A      192.0.2.106
vpn-host01   A      192.0.2.107
vpn-host02   A      192.0.2.108
...
```

- c. View the /var/named/data/reverse-192.0.2 file.
- This file defines “PTR” records for reverse name resolution.

```
[gateway]# 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.
105 PTR host05.example.com.
106 PTR host06.example.local.
107 PTR vpn-host1.example.com.
108 PTR vpn-host2.example.com.
...
```

Perform the next task from **host02**.

4. Test host name to IP resolution on **host02**.

a. Use the `ping` command to contact **host01** and **host03**.

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

```
[host02]# 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
[host02]# ping host03
PING host03.example.com (192.0.2.103) 56(84) bytes of data.
64 bytes from host03.example.com (192.0.2.103): 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.

```
[host02]# vi /etc/hosts
127.0.0.1 localhost.localdomain localhost
192.0.2.1 gateway.example.com gateway
#192.0.2.101 host01.example.com host01
#192.0.2.102 host02.example.com host02
#192.0.2.103 host03.example.com host03
192.0.2.104 host04.example.com host04
192.0.2.105 host05.example.com host05
192.0.2.106 host06.example.local host06
192.0.2.107 vpn-host1.example.com vpn-host1
192.0.2.108 vpn-host2.example.com vpn-host2
```

- c. Use the **ping** command to contact **host01** and **host03**.

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

```
[host02]# 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
[host02]# ping host03
PING host03.example.com (192.0.2.103) 56(84) bytes of data.
64 bytes from host03.example.com (192.0.2.103): 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.

```
[host02]# 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.

```
[host02]# vi /etc/nsswitch.conf
hosts: files dns # old entry
hosts: files # new entry
```

- f. Use the **ping** command to contact **host01** and **host03**.

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

```
[host02]# ping host01
ping: host01: Name or service not known
[host02]# ping host03
ping: host03: Name or service not known
```

- g. Use the **vi** editor to edit the **/etc/nsswitch.conf** file and restore the “dns” argument to the “hosts” entry as follows.

```
[host02]# vi /etc/nsswitch.conf
hosts: files # old entry
hosts: files dns # new entry
```

h. Use the ping command to contact **host01** and **host03**.

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

```
[host02]# 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
[host02]# ping host03
PING host03.example.com (192.0.2.103) 56(84) bytes of data.
64 bytes from host03.example.com (192.0.2.103): 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.

```
[host02]# cat /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

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

```
[host02]# vi /etc/resolv.conf
# Generated by NetworkManager
#search example.com
#nameserver 192.0.2.1
```

k. Use the ping command to contact **host01** and **host03**.

- You cannot contact these systems by name now.

```
[host02]# ping host01
ping: host01: Name or service not known
[host02]# ping host03
ping: host03: Name or service not known
```

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.

```
[host02]# vi /etc/resolv.conf
# Generated by NetworkManager
search example.com
nameserver 192.0.2.1
```

- m. Use the **ping** command to contact **host01** and **host03**.

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

```
[host02]# 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
[host02]# ping host03
PING host03.example.com (192.0.2.103) 56(84) bytes of data.
64 bytes from host03.example.com (192.0.2.103): icmp_seq=1...
...
CTRL-C
```

5. Use the **host** command to perform DNS lookups on **host02**.

- a. Query DNS for the nameserver for the `example.com` domain.

```
[host02]# host -t NS example.com
example.com name server dns.example.com.
```

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

```
[host02]# 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.

```
[host02]# 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.

```
[host02]# 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. ...
```

6. Use the `dig` command to perform DNS lookups on **host02**.

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

```
[host02]# 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 **host02** as a caching-only nameserver.

Assumptions

- You are the `root` user on **host02**.
- All commands in this practice with one exception are executed on **host02**.
- The one command that needs to be run on **gateway** includes **gateway** in the prompt.

Tasks

1. Install the bind software package on **host02**.

- 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
bind-libs-lite-9.9.4-61.el7.x86_64
rpcbind-0.2.0-44.el7.x86_64
bind-utils-9.9.4-61.el7.x86_64
keybinder3-0.3.0-1.el7.x86_64
bind-libs-9.9.4-61.el7.x86_64
bind-license-9.9.4-61.el7.noarch
```

- b. Use the `yum` command to install the `bind` package.

- Answer `y` when prompted “Is this ok”.

```
# yum install bind
...
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....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 **host02**.

- 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
```

```
Created symlink from /etc/systemd/system/multi-
user.target.wants/named.service to
/usr/lib/systemd/system/named.service.
```

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

```
# systemctl start named
```

- d. From the second terminal window on **gateway**, ssh to **host02** 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.
- You might want to enlarge this window to see more of the journal entries.

```
[gateway]# ssh root@host02
root@host02's password:
[host02]# journalctl -f
-- Logs begin at ...
...
```

- e. In the original window, use the `ping` command to contact **host01** and **host03**.

- 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 host03
PING host02.example.com (192.0.2.103) 56(84) bytes of data.
64 bytes from host03.example.com (192.0.2.103): icmp_seq=1...
...
CTRL-C
```

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

```
[root@host02 ~]# journalctl -f
-- Logs begin at ...
<date_time> host02... 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 **host02** from this second window.

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

4. In the first terminal window on **host02**, 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 **host02** 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
Removed symlink /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 **host02**.

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

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

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Practices for Lesson 3: Authentication and Directory Services

Practices for Lesson 3: Authentication and Directory Services

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 **gateway**.

Tasks

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

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

```
[gateway]# ssh host03
root@host03's password:
Last login: ...
[root@host03 ~]#
```

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
...
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
total 4
-rw-r--r--. 1 root root 845 ... 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
total 4
-rw-r--r--. 1 ldap ldap 845 ... DB_CONFIG
```

4. Start the slapd service.

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

```
# systemctl enable slapd
Created symlink from /etc/systemd/system/multi-
user.target.wants/slapd.service to
/usr/lib/systemd/system/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
total 19124
-rw-r--r--. 1 ldap ldap 2048 ... alock
-rw-----. 1 ldap ldap 2326528 ... __db.001
-rw-----. 1 ldap ldap 17448960 ... __db.002
-rw-----. 1 ldap ldap 1884160 ... __db.003
-rw-r--r--. 1 ldap ldap 845 ... DB_CONFIG
-rw-----. 1 ldap ldap 8192 ... dn2id.bdb
```

```
-rw-----. 1 ldap ldap    32768 ... id2entry.bdb
-rw-----. 1 ldap ldap 10485760 ... log.0000000001
```

5. View the /etc/openldap directory.

- 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
total 12
drwxr-xr-x. 2 root root   90 ... certs
-rw-r--r--. 1 root root  121 ... check_password.conf
-rw-r--r--. 1 root root  363 ... ldap.conf
drwxr-xr-x. 2 root root 4096 ... schema
drwxr-x---. 3 ldap ldap   45 ... 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
total 8
drwxr-x---. 3 ldap ldap 4096 ... cn=config
-rw-----. 1 ldap ldap  589 ... cn=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
total 20
drwxr-x---. 2 ldap ldap  29 ... cn=schema
-rw-----. 1 ldap ldap 378 ... cn=schema.ldif
-rw-----. 1 ldap ldap 513 ... olcDatabase={0}config.ldif
-rw-----. 1 ldap ldap 443 ... olcDatabase={-1}frontend.ldif
-rw-----. 1 ldap ldap 562 ... olcDatabase={1}monitor.ldif
-rw-----. 1 ldap ldap 609 ... 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 ldap://` 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. Press Enter after each entry.
- Insert a blank line as shown, by pressing the Enter key a 2nd time, before entering “EOF”.

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

- The “EOF” entry 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.
- Once again, insert a blank line as shown before entering “EOF”.

```
# 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 7c and 7d 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.
- Insert a blank line as shown before entering “EOF”.

```
# 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.
 - Repeat step 8c to correct.
- 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. Insert a blank line as shown before entering “EOF”.

- 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    duaconfd.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 `gateway` in the `/stage` directory.

- You can create the `base.ldif` file as follows by using the `vi` command, or you can use the `scp` command and copy `/stage/base.ldif` from `gateway` to `/etc/openldap/base.ldif` on `host03`.

```
# 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.

- 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

```

- At around line number 71, change the value of `$DEFAULT_MAIL_DOMAIN` from `padl.com` to `example.com`.

<code>\$DEFAULT_MAIL_DOMAIN = "padl.com";</code>	(old value)
<code>\$DEFAULT_MAIL_DOMAIN = "example.com";</code>	(new value)

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

<code>\$DEFAULT_BASE = "dc=padl,dc=com";</code>	(old value)
<code>\$DEFAULT_BASE = "dc=example,dc=com";</code>	(new value)

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

15. Migrate the users.

- 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

```

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

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

- 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:
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`”.

```
# ldapsearch -x "cn=oracle" -b "dc=example,dc=com"
...
# oracle, People, example.com
dn: uid=oracle,ou=People,dc=example,dc=com
uid: oracle
cn: oracle
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.

- 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
```

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

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

- 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
```

- 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:
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: ssh dhcpcv6-client ldap
...
```

Practice 3-2: Implementing OpenLDAP Authentication

Overview

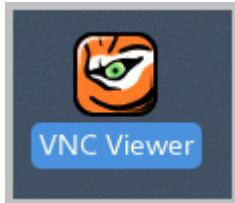
In this practice, you use the Authentication Configuration Tool to implement OpenLDAP authentication.

Assumptions

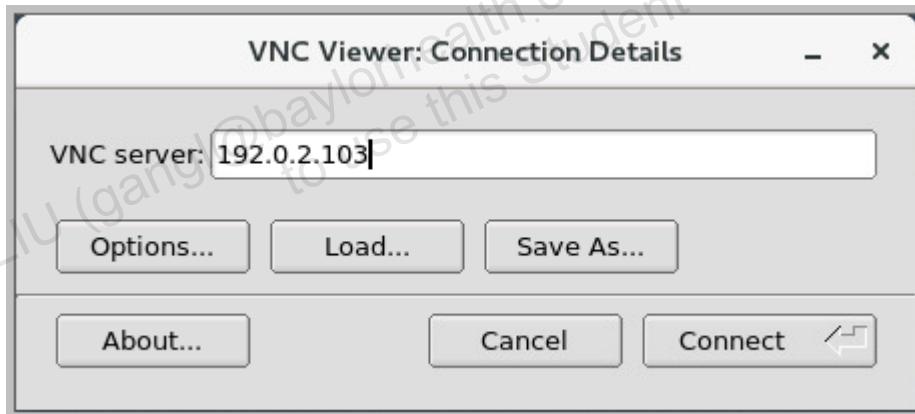
- This practice uses a GUI so you need to connect to **host03** by using VNC Viewer.

Tasks

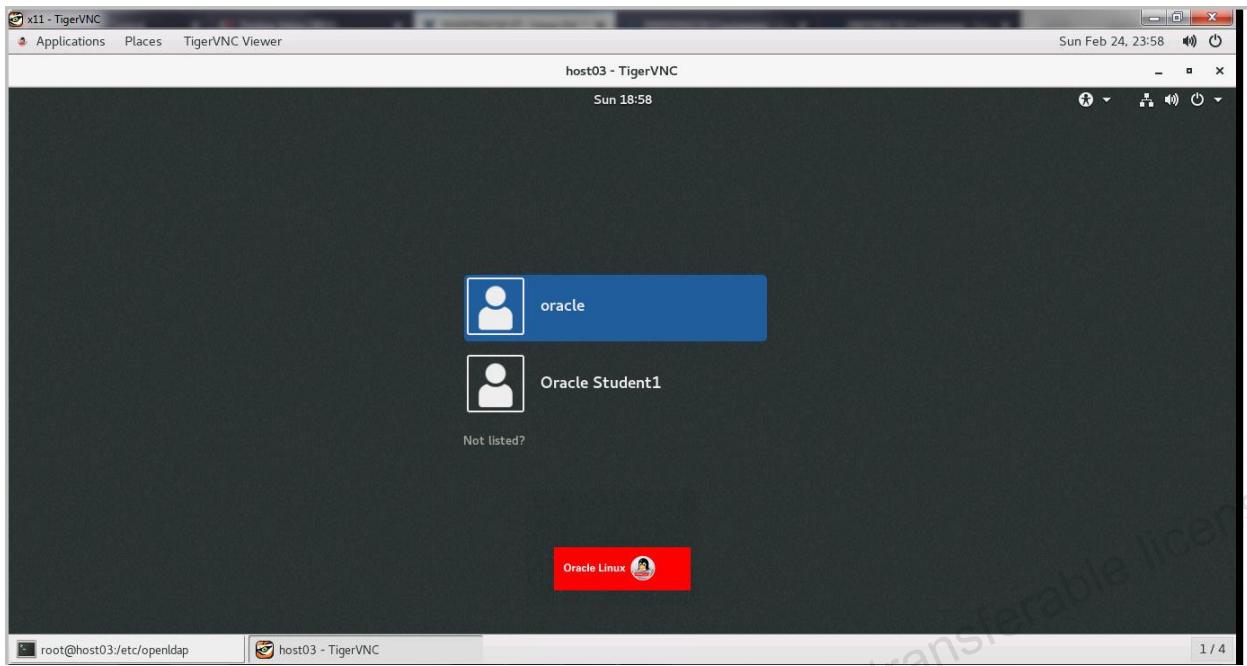
1. From the **gateway** desktop, connect to **host03** by using VNC Viewer.
 - a. From **gateway**, double click the VNC Viewer icon on the desktop.



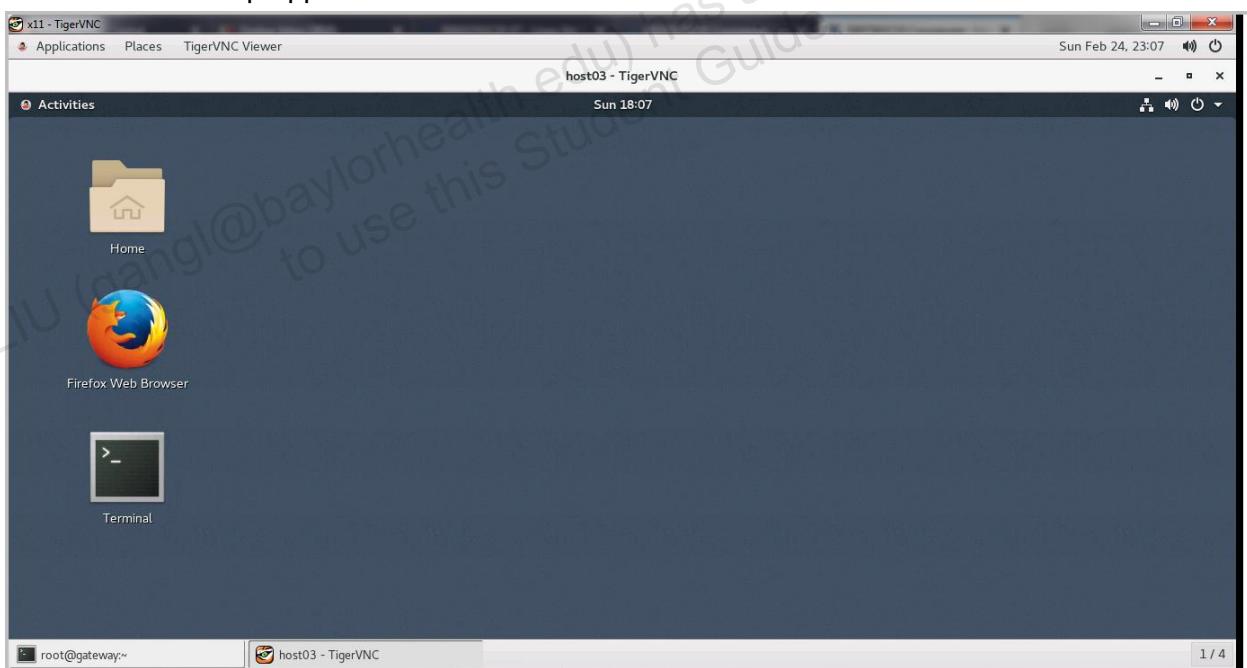
- b. Enter 192.0.2.103 as shown in the following window.
 - Click Connect.



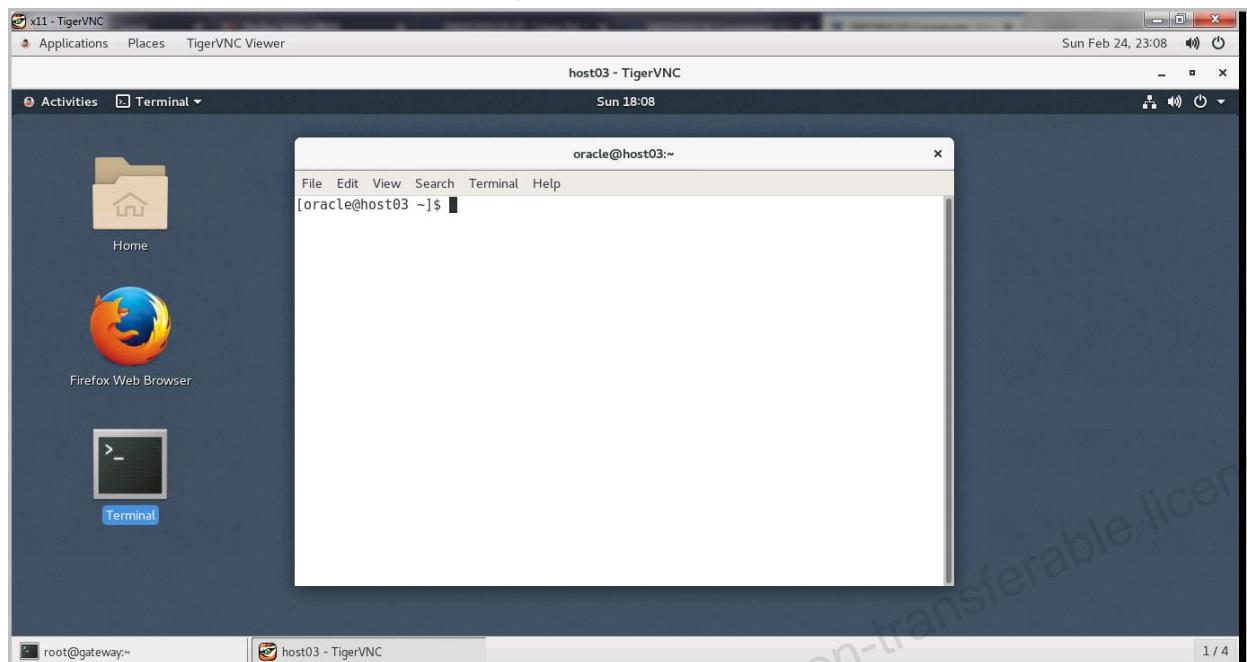
- The login window appears.
- Click in the window to see the `oracle` user.



- c. Click the `oracle` user and login.
• The desktop appears as shown.



- d. Double-click the Terminal icon to open a terminal window as shown.



- e. In the terminal window, use the `su -` command to become the `root` user.

```
[oracle@host03 ~]$ su -
Password:
Last login: ...
[root@host03 ~]#
```

2. 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
...
Is this ok [y/d/N]: y
...
Complete!
```

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

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

- The GUI appears as follows.



- Make the following changes.
 - Select LDAP from the User Account Database drop-down list.
 - The following dialog box is displayed.

The /lib64/libnss_ldap.so.2 file was not found, but it is required for LDAP support to work properly. Install the nss-pam-ldapd package, which provides this file.

 - Do not click "Install."
 - Install this package from the command line in the next step.
 - From a terminal window on **host03**, use the `yum` command to install the `nss-pam-ldapd` software package.

- Answer **y** when prompted “Is this ok”.

```
# yum install nss-pam-ldapd  
...  
Is this ok [y/d/N]: y  
...  
Complete!
```

- c. Switch back to the Authentication Configuration Tool and 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.



- d. Click “Apply” to save your changes.
- After a few seconds, the Authentication Configuration Tool closes.

5. 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 **gateway** 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 **gateway**.
 - a. If necessary, open a second terminal window on **gateway**.
 - b. From the second terminal window on **gateway**, use the **su -** command to become the **root** user.

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

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

```
[gateway]# ssh host01
root@host01's password:
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”.

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

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

- Answer `y` when prompted “Is this ok.”

```
# yum install nss-pam-ldapd
...
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
total 4
drwxr-xr-x. 2 root root 70 ... certs
-rw-r--r--. 1 root root 356 ... 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**.

- a. 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 gateway in the /stage directory.

- You can edit the system-auth file as follows by using the vi command, or you can use the scp command and copy /stage/system-auth from gateway to /etc/pam.d/system-auth on host01.
- 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 required pam_failsafe.so delay=2000000
auth sufficient pam_fprintd.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_authok
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  
oracle
```

- 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 nsldcd service.

```
# systemctl stop nsldcd
```

- 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 `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
Removed symlink /etc/systemd/system/multi-
user.target.wants/slapd.service.
```

Do not close the VNC connection or log off **host03**.

The next practice (Practice 4-1) assumes that you are still logged on to **host03**.

Practices for Lesson 4: Pluggable Authentication Modules (PAM)

Practices for Lesson 4: Overview

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           liveinst           smartcard-auth
authconfig    login              smartcard-auth-ac
...
...
```
 - 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
# Used with polkit to reauthorize users in remote sessions
-auth     optional      pam_reauthorize.so prepare
account   required      pam_nologin.so
account   include       password-auth
password  include       password-auth
# pam_sselinux.so close should be the first session rule
session   required      pam_sselinux.so close
session   required      pam_loginuid.so
# pam_sselinux.so open should only be followed by sessions to be
executed in the user context
```

```

session      required      pam_selinux.so open env_params
session      required      pam_namespace.so
session      optional     pam_keyinit.so force revoke
session      include      password-auth
session      include      postlogin
# Used with polkit to reauthorize users in remote sessions
-session    optional     pam_reauthorize.so prepare

```

- 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_lastlog.so          pam_sepermit.so
pam_cap.so             pam_ldap.so            pam_shells.so
...
...
```

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

```
[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 another course 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 **gateway**, use the `ssh` command to log in to **host01** as the `oracle` user.

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

- b. From **host01**, use the `ssh` command to connect to **host03**.
 - Answer yes to “Are you sure”.

```
[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:
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**.

- Note that the connection is denied.

```
[oracle@host01 ~]$ ssh host03
oracle@host03's password:
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**.

- Note that the connection is allowed, and no longer denied by the PAM authentication module.

```
[oracle@host01 ~]$ ssh host03
oracle@host03's password:
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 **gateway**, use the `ssh` command to log in to **host01** as `root`.

```
[gateway]# ssh host01
root@host01's password:
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”.
- 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:
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:
No logins allowed at this time.

Authentication failed.

[host03]#
```

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 other files including the `sshd` file.
- 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/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 close the VNC connection or log off **host03**.

The next practice (Practice 5-1) assumes that you are still logged on to **host03**.

Practices for Lesson 5: Web and Email Services

Practices for Lesson 5: Web and Email Services

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 VNC Viewer.
- 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
...
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
Created symlink from /etc/systemd/system/multi-
user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
```

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

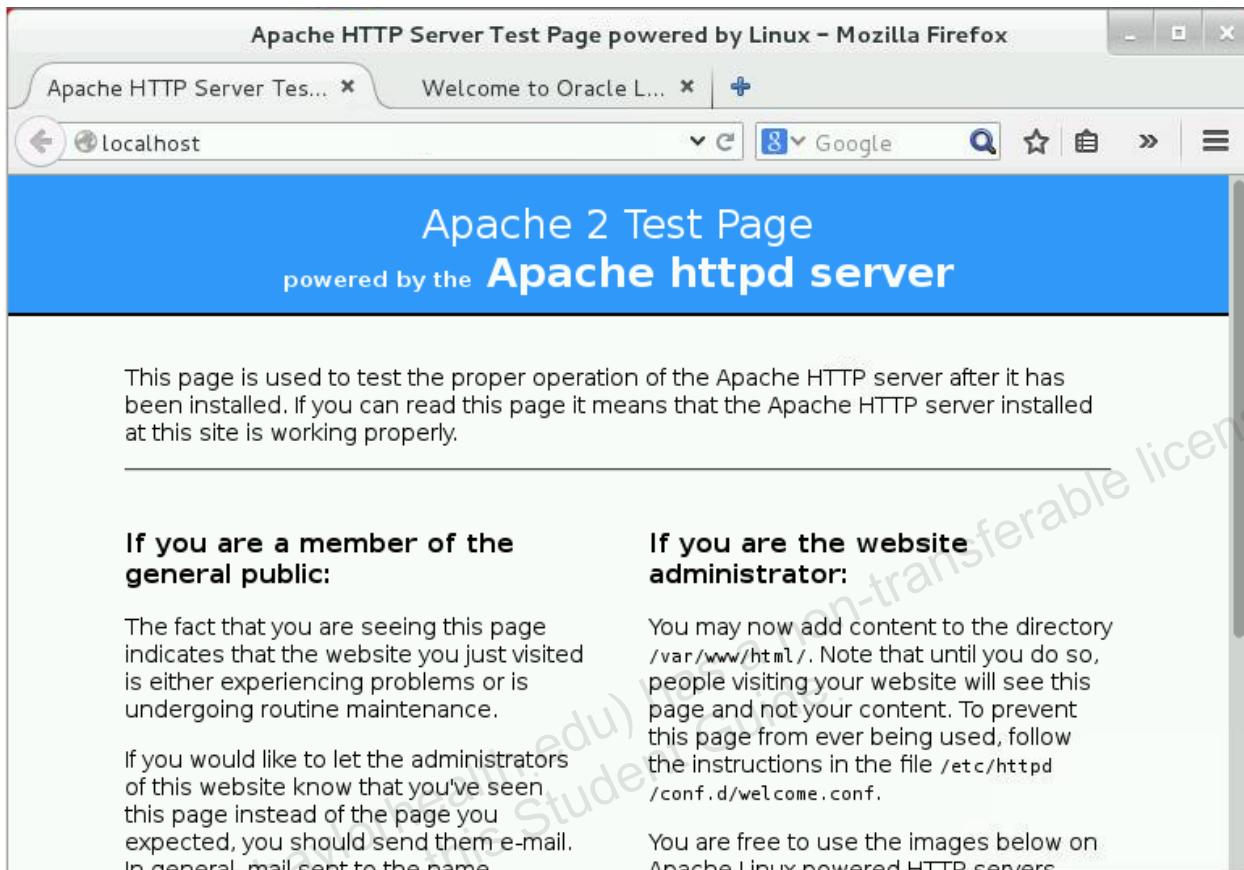
```
# systemctl start httpd
```

2. Confirm that Apache is working, by pointing a browser on **host03** to <http://localhost>.
 - a. On the **host03** desktop, double-click the “Firefox Web Browser” icon.

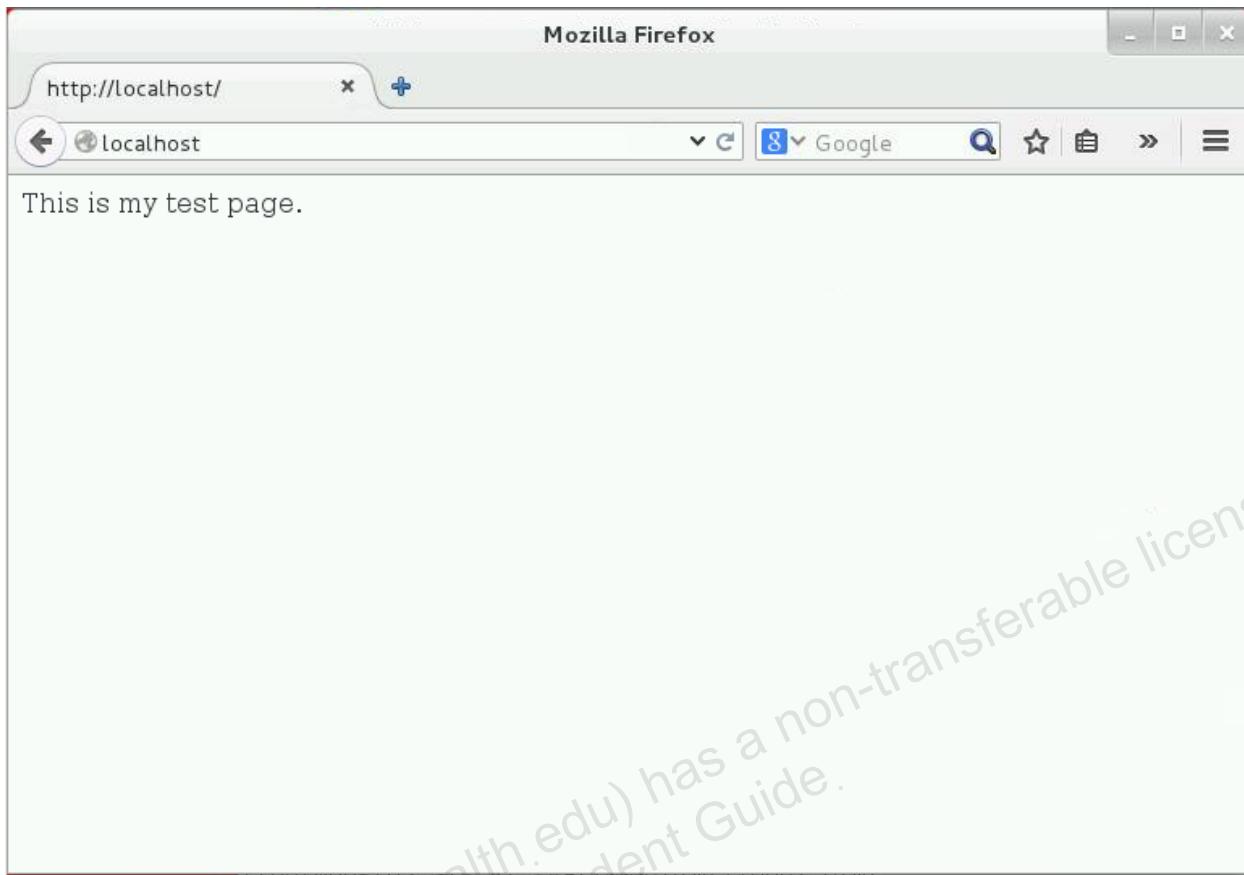


- The browser window opens.

- 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.
- 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>
```
- 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](http://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](http://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](http://www.example1.com).

- a. Restart the Firefox browser and point to <http://www.example1.com>.
- The test webpage for [www.example1.com](http://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](http://www.example2.com).
- 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>
```

- Use the vi editor to edit the /etc/hosts file to append [www.example2.com](http://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
```

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

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

- 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
```

- 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>
```

- 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.

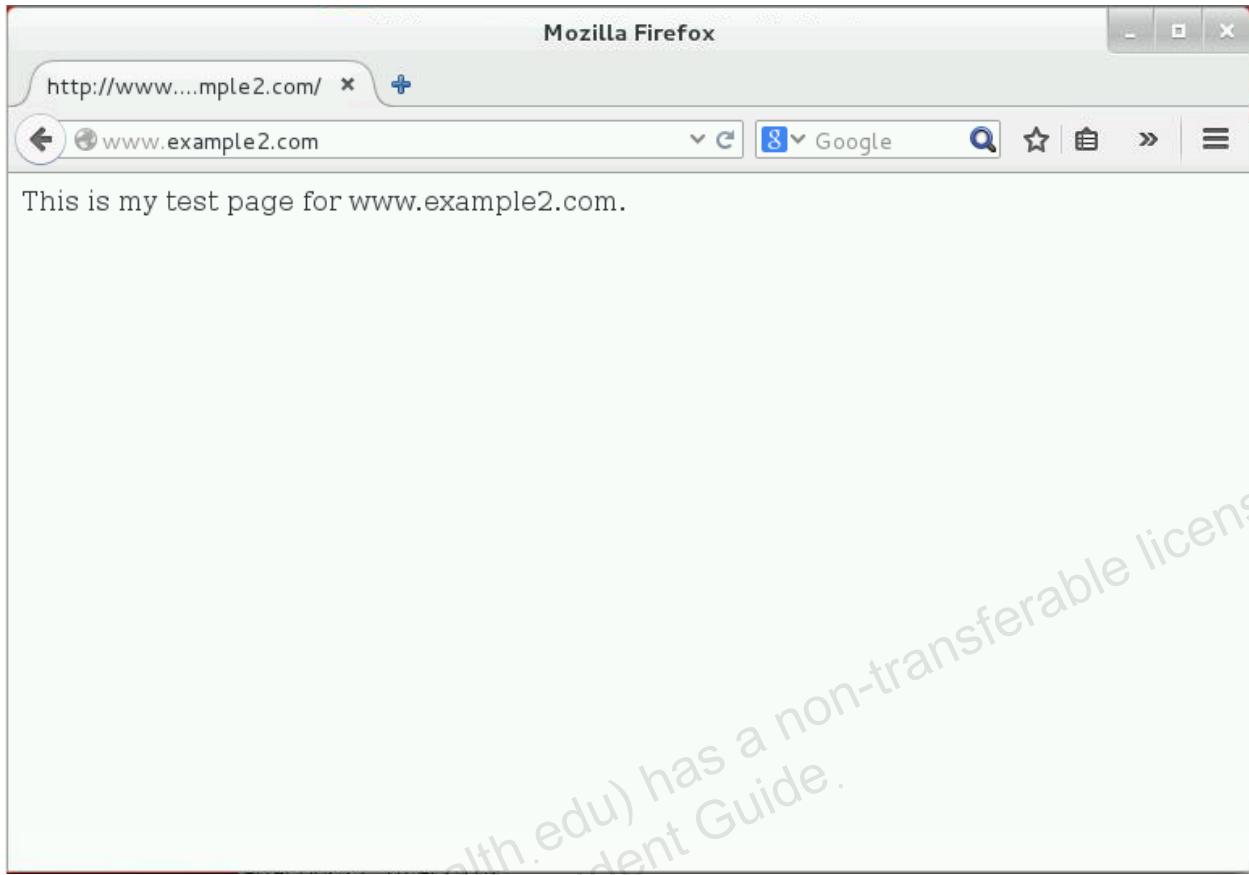
```
apachectl configtest
Syntax OK
```

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

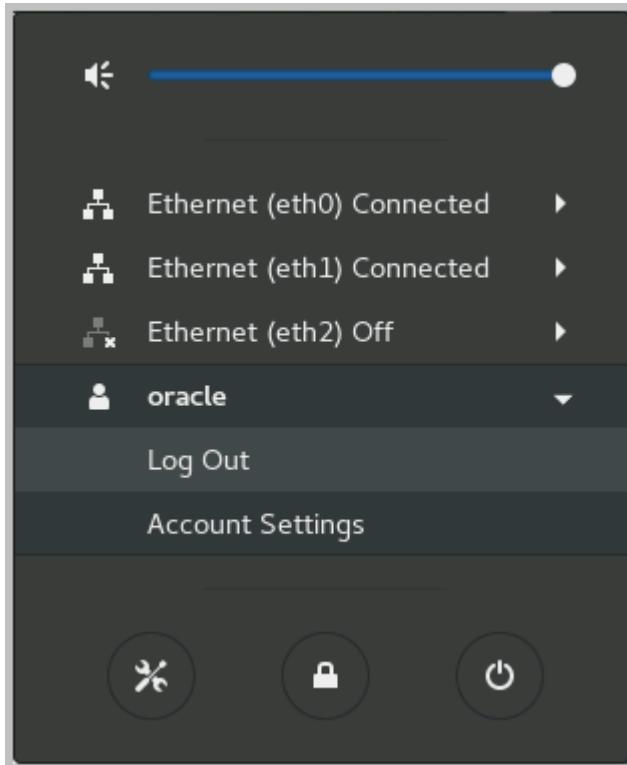
```
apachectl graceful
```

7. View the test webpage for [www.example2.com](http://www.example2.com).

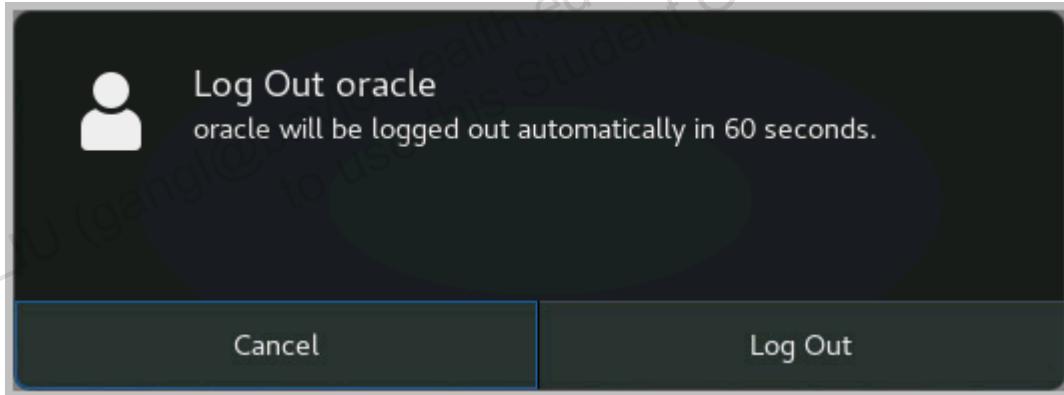
- Restart the Firefox browser and point to <http://www.example2.com>.
  - The test webpage for [www.example2.com](http://www.example2.com) appears.



- b. Close the Firefox web browser by clicking the "X" in the top-right corner of the window.
8. Log off from **host03** and close the VNC connection.
  - a. Click the down arrow in the top-right corner of the **host03** desktop window to display the drop-down menu.
  - b. Click the `oracle` user, then click Log Out as shown.



- c. Click Log Out as shown.



- The **host03** desktop closes.
- The following window might appear. Close the window by clicking the “X” in the upper-right corner.



GANG LIU (gangli@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.

## **Practices for Lesson 6: Installing Oracle Linux 7 by Using PXE and Kickstart**

## Practices for Lesson 6: Installing Oracle Linux 7 by Using PXE and 6

---

### Practices Overview

In these practices, you:

- Create a Kickstart file
- Configure the HTTP server on the **gateway** and stage the Oracle Linux ISO installation tree on the server
- Install the required packages to support PXE services on **host05**
- Configure services to support PXE clients on **host05**
- Watch a video on the PXE and Kickstart installation process

**Note:** To protect the integrity of the environment for this course and other active courses, booting a host to practice a network installation is not allowed.

## Practice 6-1: Creating a Kickstart File

---

### Overview

In this practice, you perform the following steps to create a kickstart file on the **gateway**:

### Assumptions

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

### Tasks

1. 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.
- As `root` from the **gateway**, use the `scp` command to copy `/root/anaconda-ks.cfg` from **host01** to `/var/www/html/ks.cfg` on the **gateway**.

```
[root@gateway ~]# cd /var/www/html
scp host01:~/anaconda-ks.cfg ks.cfg
root@host01's password:
anaconda-ks.cfg 100% ...
```

- The Kickstart file is now available from the HTTP server running on the **gateway**.
- You use the `vi` editor to change this Kickstart file as instructed in step 3b.

**Note:** A preconfigured `ks.cfg` file exists on the **gateway** in the `/stage/PXE` directory.

- If you do not want to edit the `ks.cfg` file as instructed in step 1c, you can use the `cp` command to copy `/stage/PXE/ks.cfg` to `/var/www/html/ks.cfg`. If you use this Kickstart file, you need not edit the file in step 1c.
- 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
```

- 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 do not appear in the following screen output.
- 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://192.0.2.1:0L75/

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
```

## Practice 6-1: Configure gateway as an HTTP Server

---

### Overview

In this practice, you:

- Configure the **gateway** as an HTTP server
- Make the installation tree available from the HTTP server

### Assumptions

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

### Tasks

- Make a backup copy HTTP configuration file on **gateway** before proceeding to step 2.
  - gateway** is already configured as an HTTP server to provide the local Yum repository to the **host0n** VMs.
  - Therefore, it is critical to make a backup copy of the `/etc/httpd/conf/httpd.conf` file before continuing with this practice.

```
cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.BAK
```

- Configure the **gateway** as an HTTP server.
  - As the `root` user on the **gateway**, use the `rpmquery` command to verify the installation of the `httpd` package.
    - This confirms that the `httpd` package is already installed on **gateway**.

```
rpmquery httpd
httpd-2.4.6-80.0.1.el7_5.1.x86_64
```

- Use the `systemctl` command to query the status of the `httpd` service.
  - The `httpd` service is enabled and active.

```
systemctl status httpd
httpd.service - The Apache HTTP Server
 Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
 Active: active (running) since ...
 ...
 ...
```

- Make the installation tree available.
  - In this task, you make the installation tree available from the HTTP server running on the **gateway**.
    - From the **gateway**, use the `cd` command to change to the `/stage` directory. Use the `ls` command to list the contents of the directory.
      - The Oracle Linux 7.5 DVD image is the `OracleLinux-R7-U5-Server-x86_64-dvd.iso` file in the `/stage` directory.

```
cd /stage
ls
...
OracleLinux-R7-U5-Server-x86_64-dvd.iso
```

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

```
mkdir /mnt/iso
```

- c. Use the `mount` command to mount the OL7.5 DVD image on `/mnt/iso`.

```
mount -t iso9660 -o loop OracleLinux-R7-U5-Server-x86_64-dvd.iso /mnt/iso
```

- d. Use the `mkdir` command to create the `/var/www/html/OL75` directory.

```
mkdir /var/www/html/OL75
```

- e. Use the `cp` command to copy all files and directories from `/mnt/iso` to `/var/www/html/OL75`.

- This command takes a minute to complete.

- This command extracts the `.treeinfo` file at the root of the ISO image.

```
cp -a -T /mnt/iso/ /var/www/html/OL75/
```

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

- 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
```

**Note:** You can just read through the following step (4) because you are not actually going to perform an installation in this practice.

If you perform step 4, ensure you have made a backup copy of the `/etc/httpd/conf/httpd.conf` file on **gateway**.

At the end of this Practice 6-3, you are directed to restore the HTTP server configuration using the backup copy of the `/etc/httpd/conf/httpd.conf` file.

Restoring the initial HTTP server configuration is necessary to ensure that **gateway** can provide local Yum repository for subsequent practices.

4. Edit the `/etc/httpd/conf/httpd.conf` file on the **gateway** and enclose the following directives within a new `<Directory>` section to the bottom of the file.

- `<Directory "/var/www/html/OL75">` - This is the full path to the directory where you copied ISO installation files.
- `Options Indexes FollowSymLinks` - The `Indexes` option returns a formatted listing of the directory when there is no `index.html` file in that directory. The `FollowSymLinks` option allows the server to follow symbolic links in the directory.
- `AllowOverride None` directive disables the Apache server from running `.htaccess` files, which slows down the server.
- `Order allow,deny` directive turns off authentication, and the `Allow from all` directive allows access from all hosts in the classroom domain directive

**Note:** A preconfigured `httpd.conf` file exists on the **gateway** in the `/stage/PXE` directory.

- If you do not want to edit the `httpd.conf` file as instructed, you can use the `scp` command to copy `/stage/PXE/httpd.conf` to `/etc/httpd/conf/httpd.conf`. If you use this HTTP file, you need not edit the file in this step.

```
vi /etc/httpd/conf/httpd.conf

ErrorLog logs/dummy-host.example.com-error_log
CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
...

<Directory "/var/www/html/OL75">
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

## Practice 6-2: Configuring PXE Services

---

### Overview

In this practice, you:

- Install required packages for the pre-boot environment on **host05**
- Configure these packages to support PXE clients
  - Configure DHCP
  - Configure TFTP
- Configure the Boot Loader for BIOS-based PXE clients

### Assumptions

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

### Tasks

1. Install the required packages for the pre-boot execution environment:
  - a. From a terminal window on the **gateway**, log in to **host05** as the `root` user.

```
[root@gateway ~]# ssh host05
root@host05's password:
Last login: ...
[root@host05 ~]#
```

- b. Use the `yum install` command to install the required packages.
    - The `dhcp` service provides DHCP functionality.
    - The `tftp-server` service provides TFTP functionality.
    - The `xinetd` service provides the same functionality as `inetd` by starting services listed in its configuration file.
    - The `syslinux` service provides the PXE boot loader files.
  - c. Install the `dhcp`, `tftp-server`, `xinetd`, and `syslinux` packages.

```
[root@host05 ~]# yum install dhcp tftp-server xinetd syslinux
...
Is this ok [y/d/N]: y
...
Complete!
```

2. Configure each of these packages to support PXE services.

- a. Enable the `tftp` service by modifying the `/etc/xinetd.d/tftp` configuration file. Change the “`disable`” parameter from “`yes`” to “`no`”; where “`no`” means to “`enable`.”

```
vi /etc/xinetd.d/tftp
default: off
description: The tftp server serves files using the trivial
file transfer protocol.
The tftp protocol is often used to boot diskless workstations,
download configuration files to network-aware printers, and to
start the installation process for some operating systems.
```

```

service tftp
{
 socket_type = dgram
 protocol = udp
 wait = yes
 user = root
 server = /usr/sbin/in.tftpd
 server_args = -s /var/lib/tftpboot
 disable = no
 per_source = 11
 cps = 100 2
 flags = IPv4
}

```

- b. Configure the DHCP services by modifying the `/etc/dhcp/dhcpd.conf` file.
- Edit the `/etc/dhcp/dhcpd.conf` file and configure entries that differentiate between BIOS-based PXE and UEFI-based PXE clients.

**Note:** The student lab environment uses a BIOS-based client.

**Note:** A preconfigured `dhcpd.conf` file exists on the **gateway** in the `/stage/PXE` directory. If you do not want to edit the `dhcpd.conf` file as instructed in this step, you can use the `scp` command to copy `/stage/PXE/dhcpd.conf` to `/etc/dhcp/dhcpd.conf`. If you use this DHCP file, you need not edit the file.

```
scp gateway:/stage/PXE/dhcpd.conf /etc/dhcp/dhcpd.conf
```

The key entries to add are:

- Interface** <interface port>: The interface to listen on for the incoming client request.
- dhcp-range**=<interface, starting IP, ending IP, netmask, duration>: Reserve a range of available IP addresses.
- domain**=<domain name>: Specify the DNS domain for the DHCP server.
- dhcp-boot**=<name of bootloader file, PXE servername, server IP>: Specify the name of the boot loader file.
- tftp-root**=<root directory>: Specify the location of the boot loader file, kernel, initial RAM disk, and pxe configuration file.
- subnet** and **netmask**
- range**: Provide a range of IP addresses.
- enable-tftp**: Enable the TFTP service provided by dnsmasq.

```

vi /etc/dhcp/dhcpd.conf

#
DHCP Server Configuration file.
see /usr/share/doc/dhcp*/dhcpd.conf.example
see dhcpd.conf(5) man page

```

```

#
set vendorclass = option vendor-class-identifier;
option pxe-system-type code 93 = unsigned integer 16;
set pxetype = option pxe-system-type;

subnet 192.0.2.0 netmask 255.255.255.0 {
 interface eth0;
 option routers 192.0.2.1;
 option subnet-mask 255.255.255.0;
 option domain-name "example.com";

 option broadcast-address 192.0.2.255;

 option time-offset -25200; # Mountain
Standard Time
 default-lease-time 21600;
 max-lease-time 43200;

 if substring(vendorclass, 0, 9)="PXEClient" {
 if pxetype=00:06 or pxetype=00:07 {
 filename "efi/grubx64.efi";
 } else {
 filename "pxelinux/pxelinux.0";
 }
 }
pool {

 range 192.0.2.200 192.0.2.254;

}
next-server 192.0.2.103;
}

```

- Start the dhcpcd service and configure it to start after a reboot.

```

systemctl start dhcpcd
systemctl enable dhcpcd
Created symlink from /etc/systemd/system/multi-
user.target.wants/dhcpcd.service to
/usr/lib/systemd/system/dhcpcd.service.

```

### 3. Configure the boot loader files for BIOS-based PXE clients.

- The syslinux package contains the boot loader files.
- You installed this package earlier in this practice.

- Create the pxelinux and pxelinux.cfg directories.
  - PXE uses the pxelinux directory to store the boot loader files.
  - PXE uses the pxelinux.cfg directory for the boot loader configuration files.

```
mkdir -p /var/lib/tftpboot/pxelinux/pxelinux.cfg
```

- Copy the BIOS boot loader files, the installation kernel (vmlinuz), and the ram-disk image file (initrd.img) to the pxelinux directory.
  - Copy the boot loader files to the pxelinux directory.

```
cp /usr/share/syslinux/* /var/lib/tftpboot/pxelinux/
```

- Create default configuration file in the pxelinux.cfg directory and add the following directives:

- prompt 0
- default o17
- timeout 0
- label o17
- kernel vmlinuz
- append initrd=initrd.img inst.ks=http://192.0.2.1/ks.cfg

Note that you can copy the preconfigured default from the /stage/PXE directory on the **gateway** to the /var/lib/tftpboot/pxelinux/pxelinux.cfg directory on **host05**.

```
scp gateway:/stage/PXE/default
/var/lib/tftpboot/pxelinux/pxelinux.cfg
cd /var/lib/tftpboot/pxelinux/pxelinux.cfg
vi default
prompt 0
default o17
timeout 0
label o17
kernel vmlinuz
append initrd=initrd.img inst.ks=http://192.0.2.1/ks.cfg
```

- Log off **host05**.

```
[root@host05 ~]# exit
logout
Connection to host05 closed.
[root@gateway ~]#
```

- Restore HTTP Server configuration on **gateway**.

- In “Practice 6-2: Configure **gateway** as an HTTP Server”, you were instructed to make a backup copy of the /etc/httpd/conf/httpd.conf file before continuing with this practice.
- In this task, you are instructed to restore from this backup copy and verify the local Yum repository functionality on **gateway**.
  - Use the following cp command to restore the httpd.conf file from the backup copy.

```
cp /etc/httpd/conf/httpd.conf.BAK /etc/httpd/conf/httpd.conf
cp: overwrite '/etc/httpd/conf/httpd.conf'? y
```

- b. Use the following commands to verify the local Yum repository functionality on **gateway**.
- You are instructed to log on to **host05**, test the yum repository, then log off **host05**.

```
[root@gateway ~]# ssh host05
root@host05's password:
Last login: ...

[root@host05 ~]# yum clean all
Loaded plugins: ulninfo
Cleaning repos: local_repo
Cleaning up everything
...

[root@host05 ~]# yum repolist
Loaded plugins: ulninfo
local_repo
local_repo/primary_db | 2.9 kB 00:00
repo id repo name status
local_repo "local_repo packages" 288
repolist: 288

[root@host05 ~]# exit
logout
Connection to host05 closed.
[root@gateway ~]#
```

## Practice 6-3: Video of Performing a PXE/Kickstart Installation

### Overview

This video demonstrates the following:

- A review of the files associated with configuring a network installation server.
- Boot the installation host and note items associated with the PXE installation process.
- Log in to the host and verify the installation.

### Assumptions

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

### Tasks

1. You have two methods for watching the video of the PXE network installation.
  - a. To play the video with audio, point a browser on your PC to the IP address of your student machine ([http://<IP\\_address\\_of\\_your\\_student\\_system/demos](http://<IP_address_of_your_student_system/demos)).
    - The following shows IP address of 10.237.16.160. Your IP address is different.
    - From the index of the `/demos` directory, click the link to `Practice6-4.mp4`



- b. To play the video from the **gateway**, click the Firefox icon from the **gateway** desktop, and then enter the following URL: <http://localhost/demos>, as shown.
  - Audio is unavailable when watching the video from the **gateway**.



- Click the `Practice6-4.mp4` link to launch the video.

A screenshot of a Firefox browser window. The title bar says "Index of /demos". The address bar contains "localhost/demos/" with a red box around it. Below the address bar are navigation icons (back, forward, search, home) and links for "Most Visited", "Oracle Linux Home", "Oracle Linux Support", and "Oracle Linux Blog". The main content area displays an "Index of /demos" page with a table:

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>			
<a href="#">Practice6-4.mp4</a>	2019-01-23 12:18	8.0M	

- Close the browser window by clicking the X in the upper-right corner.

## **Practices for Lesson 7: Samba Services**

## Practices for Lesson 7: Overview

---

### 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:

- Review 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 `smb.conf` file.
- Create a password for user `user01`.

### Assumptions

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

### Tasks

1. Install the `samba` packages on **host03**.

- a. As the `root` user on **gateway**, use the `ssh` command to log in to **host03**.

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

- b. From **host03**, use the `rpm -qa` command to list the installed `samba` packages.

- In this example, six `samba` packages are installed.

```
rpm -qa | grep samba
samba-libs-4.8.3-4.el7.x86_64
samba-4.8.3-4.el7.x86_64
samba-common-libs-4.8.3-4.el7.x86_64
samba-common-tools-4.8.3-4.el7.x86_64
samba-client-libs-4.8.3-4.el7.x86_64
samba-common-4.8.3-4.el7.noarch
```

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; vendor preset: disabled)
 Active: inactive (dead)
...
...
```

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

```
systemctl enable smb
Created symlink from /etc/systemd/system/multi-
user.target.wants/smb.service to
/usr/lib/systemd/system/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;
 vendor preset: disabled)
 Active: active (running) since ...
...
...
```

3. Modify `firewalld` to allow access to the samba service.

- a. Use the `firewall-cmd` command to list the services that are trusted for the active zone.

- In this example, the `ssh`, `dhcpv6-client`, and `ldap` services are trusted.

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

- b. 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. Use 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.

- Note that there are two `smb.conf` files in this directory.

```
cd /etc/samba
ls
lmhosts smb.conf smb.conf.example
```

- b. Use the `head` command to view the first few lines of the `smb.conf` file.

- Note that the `smb.conf.example` file provides a more detailed config file.

```
head smb.conf
See smb.conf.example for a more detailed config file or
read the smb.conf manpage.
Run 'testparm' to verify the config is correct after
you modified it.

[global]
 workgroup = SAMBA
 security = user

 passdb backend = tdbsam
```

- Use the `smb.conf.example` file in this practice.

- c. Use the `mv` command to rename `smb.conf.example` file to `smb.conf`.

- Make a backup copy of the `smb.conf` file before overwriting it as shown.

```
cp smb.conf smb.conf.BAK

mv smb.conf.example smb.conf
mv: overwrite 'smb.conf'? y
```

- d. 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
```

- e. At around line number 84, 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
```

- f. At around line number 87, 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
```

- g. At around line number 116, 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
```

- h. At around line number 275, 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
```

- i. At or around line number 281, 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
```

- j. Save the changes to the `smb.conf` file and exit `vi`.

6. Use the `testparm` command to check the syntax of the `smb.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]
log file = /var/log/samba/log.%m
```

```

max log size = 50
netbios name = SMB-HOST03
security = USER
server string = Samba Server Version %v
workgroup = GROUPA
idmap config * : backend = tdb
cups options = raw

[homes]
browseable = No
comment = Home Directories
read only = No

[tmp]
guest ok = Yes
path = /tmp
read only = No

[printers]
browseable = No
comment = All Printers
path = /var/spool/samba
printable = Yes

```

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;
 vendor preset: disabled)
 Active: active (running) since ...
 ...

```

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**.

- If necessary, open a new terminal window on **gateway**.
- Use the `su -` command to become the `root` user on **gateway**.

```
[gateway]$ su -
Password:
```

- As the `root` user on **gateway**, use the `ssh` command to log in to **host01**.

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

- From **host01**, use the `rpm -qa` command to list the installed `samba-client` packages.
  - In this example, four `samba-client` packages are installed.
  - The `samba-client` packages are installed.

```
[host01]# rpm -qa | grep samba
samba-common-4.8.3-4.el7.noarch
samba-common-libs-4.8.3-4.el7.x86_64
samba-client-libs-4.8.3-4.el7.x86_64
samba-client-4.8.3-4.el7.x86_64
```

2. From **host01**, access the Samba shares on **host03** as user `user01`.

- 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
Try "help" to get a list of possible commands.
smb: \>
```

- 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
```

- 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: \>
```

- Use the `exit` command to exit the `smb` session on **host01**.

```
smb: \> exit
```

- Use the `smbclient` command to access the home directory for user `user01` on **host03**.

- The Samba password for `user01` is the same password created earlier.

```
[host01]# smbclient //host03/user01 -U user01
```

```
Enter user01's password: MyOracle1
```

```
Try "help" to get a list of possible commands.
```

```
smb: \>
```

- 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 another course.

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

- Use the `exit` command to exit the `smb` session.

```
smb: \> exit
```

- 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
```

- On **host01**, re-issue the `smbclient` command to access the home directory for user `user01` on **host03**.

- The Samba password for user01 is the same password created earlier.

```
[host01]# smbclient //host03/user01 -U user01
Enter user01's password:
Try "help" to get a list of possible commands.
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 on **host03**, 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 /homedit
```

- 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
/homedit
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 17G 4.0G 14G 23% /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,cache=strict,username=user01,domain=SMB-
HOST03,uid=0,noforceuid,gid=0,noforcegid,addr=192.0.2.103,unix,p
osixpaths,serverino,mapposix,acl,rsize=1048576,wsize=65536,actim
eo=1)
```

- 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 `exit` command to log off **host03**.

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

## 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.

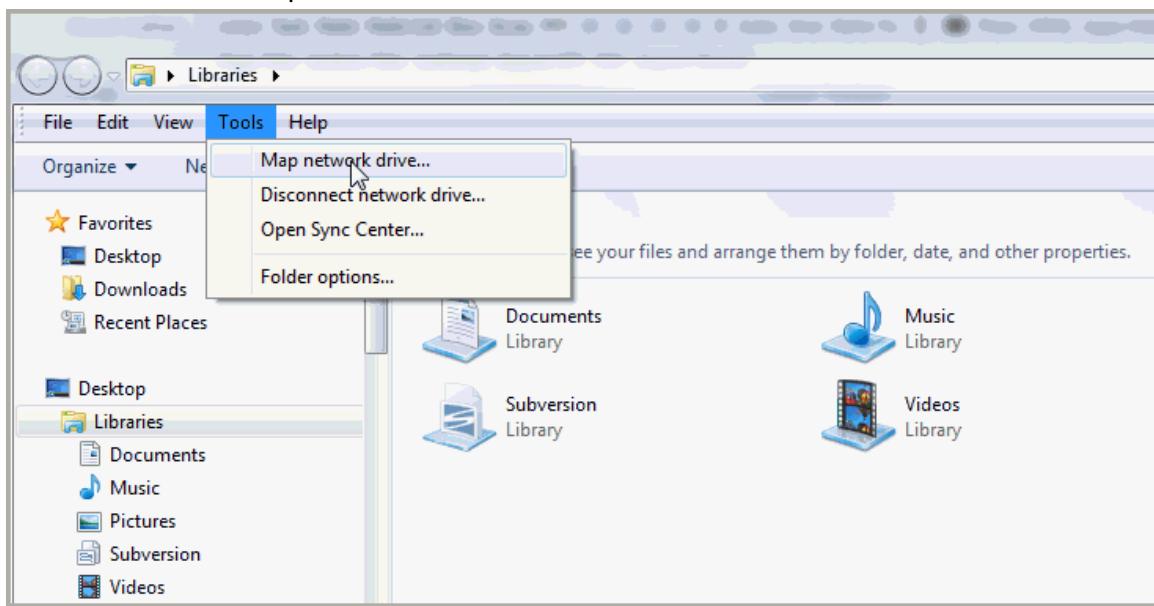
### Assumptions

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

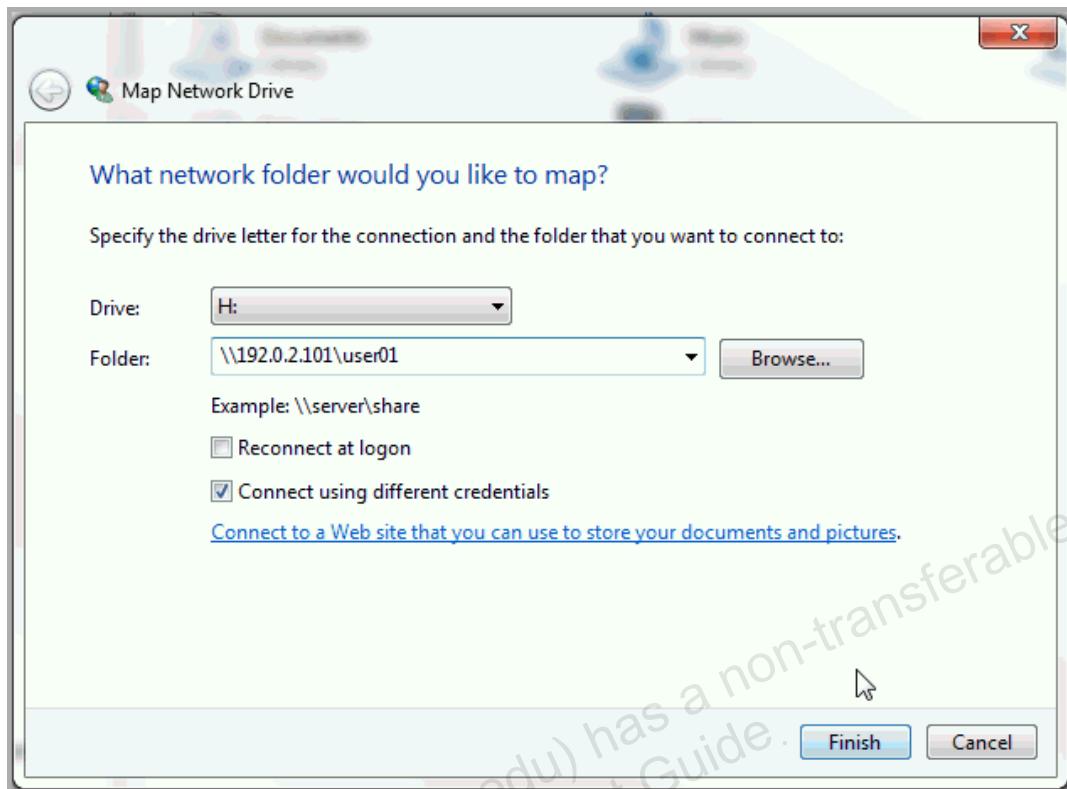
### Tasks

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.

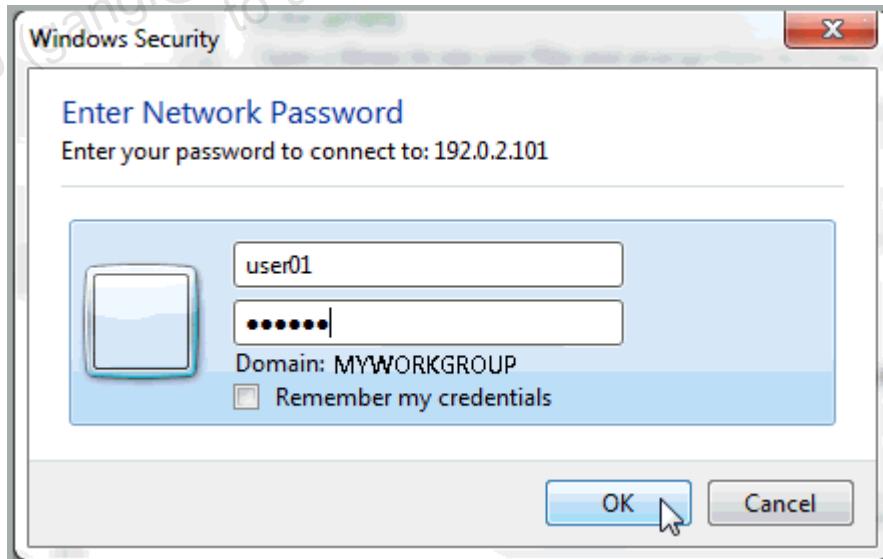
- a. Launch the tool to map a network drive.



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

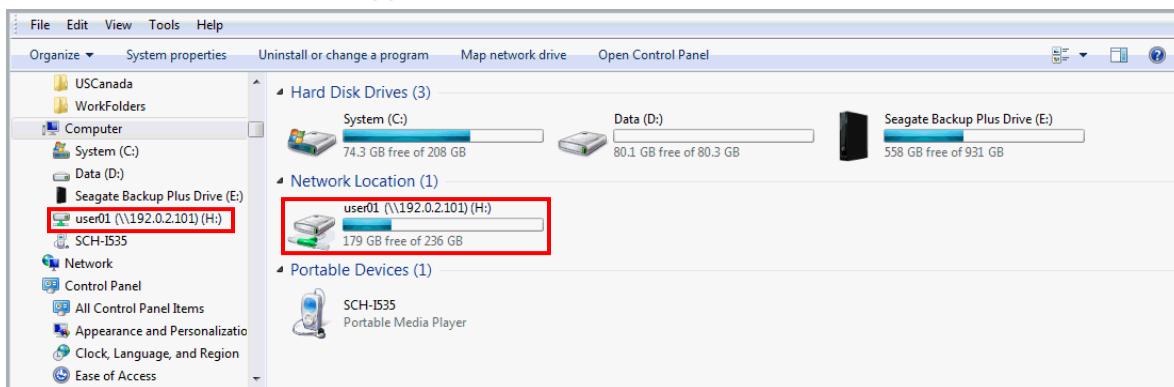


- Select "Connect using different credentials" to provide your Linux username and its associated Samba password.
- c. In the Windows Security window, enter the credentials for the share as user01 and the Samba password as MyOracle1.

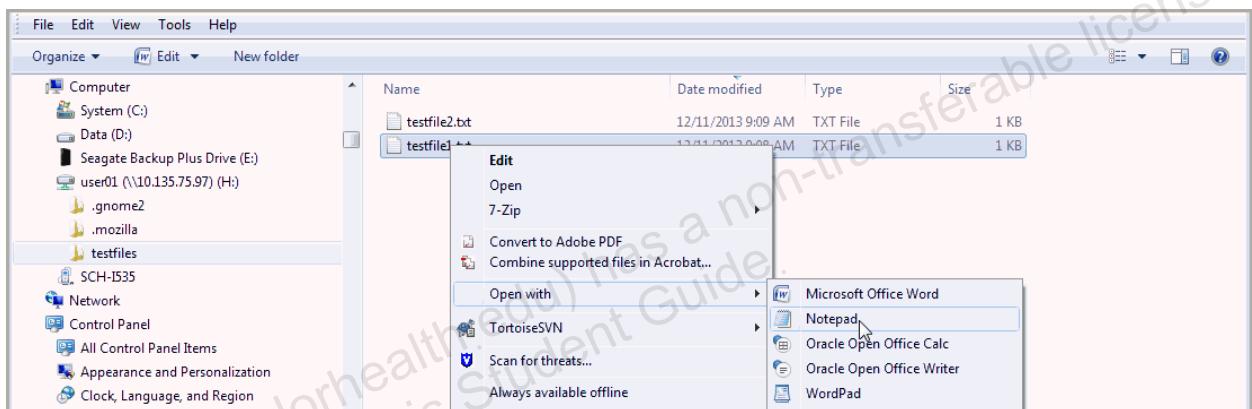


- Click OK to access and map the drive.

- After successful completion of the mapping operation, the home directory for user01 on host01 is mapped to drive H:..



- You can view and manipulate the files in the H: drive:



- Use Disconnect to release the network share.

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.

## **Practices for Lesson 8: Advanced Software Package Management**

## Practices for Lesson 8: Overview

---

### 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:

- Log in to **host04**.
- View Oracle Linux Yum Server configuration on **host04**.

### Assumptions

- You are the `root` user on **gateway**.
- The **host04** VM is preconfigured to access Oracle Linux Yum Server.

### Tasks

1. Log in to **host04**.

- As the `root` user on **gateway**, use the `ssh` command to log in to **host04**.

```
[gateway]# ssh host04
root@host04's password:
Last login: ...
[root@host04 ~]#
```

2. View the Oracle Linux Yum Server configuration on **host04**.

- The **host04** VM is preconfigured to access Oracle Linux Yum Server.
  - There are four files that provide access to Oracle Linux Yum Server:
    - `/etc/sysconfig/network-scripts/ifcfg-eth0`
    - `/etc/resolv.conf`
    - `/etc/yum.conf`
    - `/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=10.237.18.52
DNS3=152.68.154.3
DOMAIN="edu.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 edu.oracle.com example.com example.com
```

```
nameserver 192.0.2.1
nameserver 10.237.18.52
nameserver 152.68.154.3
```

- c. Use the `cat` command to view the `/etc/yum.conf` file.  
• The HTTP proxy server variable is set in the last line of this file.

```
cat /etc/yum.conf
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
...
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 Oracle Linux Yum repositories are enabled:

- `ol7_latest` (`enabled=1`)
- `ol7_UEKR4` (`enabled=1`)

```
cat /etc/yum.repos.d/public-yum-ol7.repo
[ol7_latest]
...
enabled=1
...
[ol7_UEKR4]
...
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
Loaded plugins: langpacks, ulninfo
Cleaning repos: ol7_UEKR4 ol7_latest
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space
taken by orphaned data from disabled or removed repos
```

- b. Run the `yum repolist` command.
  - Many `yum` commands display the plug-ins; this is just one example.
  - The first `yum` command might take a few minutes to complete because the `yum` repository(s) 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` and the `ulninfo` Yum plug-in are loaded.

```
yum repolist
Loaded plugins: langpacks, ulninfo
...
repo id repo name
status
ol7_UEKR4/x86_64 Latest Unbreakable Enterprise Kernel Release 4
for Oracle 96
ol7_latest/x86_64 Oracle Linux 7Server Latest (x86_64)
12,218
repolist: 12,314
```

- 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 three configuration files but only two plug-ins listed in the output of step 1a and step 1b. 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
-rw-r--r--. ... ulninfo.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
evolution-data-server-langpacks-3.28.5-1.el7.noarch
yum-langpacks-0.4.2-7.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/yum-plugins/langpacks.py
/usr/lib/yum-plugins/langpacks.pyc
/usr/lib/yum-plugins/langpacks.pyo
/usr/share/doc/yum-langpacks-0.4.2
/usr/share/doc/yum-langpacks-0.4.2/COPYING
```

```
/usr/share/doc/yum-langpacks-0.4.2/README
/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, ulninfo
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, ulninfo
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, ulninfo
...
Is this ok [y/d/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, ulninfo
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, ulninfo
...
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.

```
yum list available | grep yum-plugin
kabi-yum-plugins.noarch 1.0-3.0.1.el7
ol7_latest
yum-plugin-aliases.noarch 1.1.31-50.0.1.el7
ol7_latest
yum-plugin-changelog.noarch 1.1.31-50.0.1.el7
ol7_latest
yum-plugin-copr.noarch 1.1.31-46.0.1.el7_5
ol7_latest
yum-plugin-ovl.noarch 1.1.31-50.0.1.el7
ol7_latest
yum-plugin-pre-transaction-actions.noarch
yum-plugin-tmprepo.noarch 1.1.31-50.0.1.el7
ol7_latest
yum-plugin-verify.noarch 1.1.31-50.0.1.el7
ol7_latest
yum-plugin-versionlock.noarch 1.1.31-50.0.1.el7
ol7_latest
```

- 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/lib/yum-plugins/aliases.py
/usr/lib/yum-plugins/aliases.pyc
/usr/lib/yum-plugins/aliases.pyo
/usr/share/doc/yum-plugin-aliases-1.1.31
/usr/share/doc/yum-plugin-aliases-1.1.31/COPYING
/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, ulninfo
Alias ALL = --enablerepo=development --enablerepo=updates...
Alias ALLDBG = --enablerepo=fedora-debuginfo --enablerepo=...
Alias ALLSRC = --enablerepo=fedora-source --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 5.0.2-31.el7_1.1 ol7_latest
zziplib.i686 0.13.62-9.el7 ol7_latest
zziplib.x86_64 0.13.62-9.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, ulninfo
ELBA-2018-3230 bugfix GeoIP-1.5.0-13.el7.x86_64
ELBA-2018-1406 bugfix NetworkManager-1:1.10.2-
 14.el7_5.x86_64
ELBA-2018-2000 bugfix NetworkManager-1:1.10.2-
 16.el7_5.x86_64
ELBA-2018-3207 bugfix NetworkManager-1:1.12.0-
 6.el7.x86_64
ELBA-2018-3349 bugfix NetworkManager-1:1.12.0-
 7.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-1:1.12.0-
 8.el7_6.x86_64
...
ELSA-2018-3665 Important/Sec. NetworkManager-wifi-1:1.12.0-
 8.el7_6.x86_64
```

```

ELSA-2018-1224 Moderate/Sec. PackageKit-1.1.5-
2.0.1.el7_5.x86_64
ELSA-2018-1224 Moderate/Sec. PackageKit-command-not-found-
1.1.5-2.0.1.el7_5.x86_64
ELSA-2018-1224 Moderate/Sec. PackageKit-glib-1.1.5-
2.0.1.el7_5.x86_64
ELSA-2018-1224 Moderate/Sec. PackageKit-gstreamer-plugin-1.1.5-
2.0.1.el7_5.x86_64
ELSA-2018-1224 Moderate/Sec. PackageKit-gtk3-module-1.1.5-
2.0.1.el7_5.x86_64
ELSA-2018-1224 Moderate/Sec. PackageKit-yum-1.1.5-
2.0.1.el7_5.x86_64
...
ELEA-2018-3114 enhancement alsalib-1.1.6-2.el7.x86_64
ELEA-2018-3138 enhancement alsaplugins-pulseaudio-1.1.6-
1.el7.x86_64
ELEA-2018-3139 enhancement alsauutils-1.1.6-1.el7.x86_64
ELBA-2018-3035 bugfix anaconda-widgets-21.48.22.147-
1.0.1.el7.x86_64
...
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, ulninfo
CVE-2018-15688 Important/Sec. NetworkManager-1:1.12.0-
8.el7_6.x86_64
CVE-2018-15688 Important/Sec. NetworkManager-adsl-1:1.12.0-
8.el7_6.x86_64
CVE-2018-15688 Important/Sec. NetworkManager-config-server-
1:1.12.0-8.el7_6.noarch
...
CVE-2018-15688 Important/Sec. NetworkManager-wifi-1:1.12.0-
8.el7_6.x86_64
CVE-2018-1106 Moderate/Sec. PackageKit-1.1.5-
2.0.1.el7_5.x86_64
CVE-2018-1106 Moderate/Sec. PackageKit-command-not-found-
1.1.5-2.0.1.el7_5.x86_64
CVE-2018-1106 Moderate/Sec. PackageKit-glib-1.1.5-
2.0.1.el7_5.x86_64
CVE-2018-1106 Moderate/Sec. PackageKit-gstreamer-plugin-
1.1.5-2.0.1.el7_5.x86_64
CVE-2018-1106 Moderate/Sec. PackageKit-gtk3-module-1.1.5-
2.0.1.el7_5.x86_64

```

```

CVE-2018-1106 Moderate/Sec. PackageKit-yum-1.1.5-
2.0.1.el7_5.x86_64

...
CVE-2018-7208 Low/Sec. binutils-2.27-
34.base.0.1.el7.x86_64
CVE-2018-7569 Low/Sec. binutils-2.27-
34.base.0.1.el7.x86_64
CVE-2018-7642 Low/Sec. binutils-2.27-
34.base.0.1.el7.x86_64
CVE-2018-8945 Low/Sec. binutils-2.27-
34.base.0.1.el7.x86_64
CVE-2018-10372 Low/Sec. binutils-2.27-
34.base.0.1.el7.x86_64

...
CVE-2018-1111 Critical/Sec. dhclient-12:4.2.5-
68.0.1.el7_5.1.x86_64
CVE-2018-1111 Critical/Sec. dhcp-common-12:4.2.5-
68.0.1.el7_5.1.x86_64
CVE-2018-1111 Critical/Sec. dhcp-libs-12:4.2.5-
68.0.1.el7_5.1.x86_64
CVE-2018-5148 Important/Sec. firefox-52.7.3-
1.0.1.el7_5.x86_64
...
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-2018-15688
Loaded plugins: aliases, langpacks, ulninfo
ELSA-2018-3665 Important/Sec. NetworkManager-1:1.12.0-
8.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-adsl-1:1.12.0-
8.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-config-server-
1:1.12.0-8.el7_6.noarch
ELSA-2018-3665 Important/Sec. NetworkManager-glib-1:1.12.0-
8.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-libnm-1:1.12.0-
8.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-ppp-1:1.12.0-
8.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-team-1:1.12.0-
8.el7_6.x86_64

```

```

ELSA-2018-3665 Important/Sec. NetworkManager-tui-1:1.12.0-
8.el7_6.x86_64
ELSA-2018-3665 Important/Sec. NetworkManager-wifi-1:1.12.0-
8.el7_6.x86_64
ELSA-2019-0049 Important/Sec. libgudev1-219-
62.0.4.el7_6.2.x86_64
ELSA-2019-0049 Important/Sec. systemd-219-62.0.4.el7_6.2.x86_64
ELSA-2019-0049 Important/Sec. systemd-libs-219-
62.0.4.el7_6.2.x86_64
ELSA-2019-0049 Important/Sec. systemd-python-219-
62.0.4.el7_6.2.x86_64
ELSA-2019-0049 Important/Sec. systemd-sysv-219-
62.0.4.el7_6.2.x86_64
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-2018-15688
Loaded plugins: aliases, langpacks, ulninfo
=====
NetworkManager security update
=====
Update ID : ELSA-2018-3665
Release : Oracle Linux 7
Type : security
Status : final
Issued : 2018-11-27
CVEs : CVE-2018-15688
Description : [1:1.12.0-8]
 : - dhcp: fix out-of-bounds heap write for DHCPv6
 : with internal plugin (CVE-2018-15688)
Severity : Important
=====
systemd security update
=====
Update ID : ELSA-2019-0049
Release : Oracle Linux 7
Type : security
Status : final
Issued : 2019-01-14
CVEs : CVE-2018-15688

```

```

: CVE-2018-16864
: CVE-2018-16865
Description : [219-62.0.4]
: - do not create utmp update symlinks for reboot
: and poweroff [Orabug: 27854896]
: - OL7 udev rule for virtio net standby interface
: [Orabug: 28826743]
: - fix _netdev is missing for iscsi entry in
: /etc/fstab [Orabug: 25897792]
: (tony.l.lam@oracle.com)
: - set 'RemoveIPC=no' in logind.conf as default for
: OL7.2 [22224874]
: - allow dm remove ioctl to co-operate with UEK3
: (Vaughan Cao) [Orabug: 18467469]
: - add hv dynamic memory support (Jerry Snitselaar)
: [Orabug: 18621475]
:
: [219-62.2]
: - journald: do not store the iovec entry for
: process commandline on stack (#1657788)
: - journald: set a limit on the number of fields
: (1k) (#1657792)
: - journal-remote: set a limit on the number of
: fields in a message (#1657792)
:
: [219-62.1]
: - dhcp6: make sure we have enough space for the
: DHCP6 option header (CVE-2018-15688)

Severity : Important
updateinfo info done

```

e. Update the packages affected by the specific CVE.

- Answer **y** when asked “Is this ok.”

```

yum update --cve CVE-2018-15688
Loaded plugins: aliases, langpacks, ulninfo
...
Is this ok [y/d/N]: y
...
Complete!

```

2. View the Oracle Database preinstallation packages (`oracle-database`).
- Use the `yum` command to list the Oracle Database preinstallation package (`oracle-database`) that are available for installation.
    - You can use the `lsa` alias instead of `list available`.

```
yum list available | grep oracle-database
oracle-database-preinstall-18c.x86_64 1.0-1.el7 ol7_latest
oracle-database-server-12cR2-preinstall.x86_64
```

- View more information for the Oracle Database preinstallation package.
  - In this example, the package name is `oracle-database-preinstall-18c`.

```
yum info oracle-database-preinstall-18c
Loaded plugins: aliases, langpacks, ulninfo
Available Packages
Name : oracle-database-preinstall-18c
Arch : x86_64
Version : 1.0
Release : 1.el7
Size : 18 k
Repo : ol7_latest/x86_64
Summary : Sets the system for Oracle Database single
instance and Real
 : Application Cluster install for Oracle Linux 7
License : GPLv2
Description : The Oracle Preinstallation RPM package installs
software packages
 : and sets system parameters required for Oracle
Database single
 : instance and Oracle Real Application Clusters
installations for
 : Oracle Linux Release 7 Files affected:
/etc/sysctl.conf,
 : /boot/grub/menu.lst OR /boot/grub2/grub.cfg Files
added:
 : /etc/security/limits.d/oracle-database-preinstall-
18c.conf
```

- 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-database-preinstall-18c` package in the next step.

```
repoquery --requires oracle-database-preinstall-18c
/bin/bash
```

```
/bin/sh
/etc/redhat-release
bc
bind-utils
binutils
compat-libcap1
compat-libstdc++-33
ethtool
glibc
...
util-linux-ng
xorg-x11-utils
xorg-x11-xauth
```

- d. Use the --downloadonly option to download the oracle-database-preinstall-18c package and any missing dependent packages.
- In this example, six packages are downloaded in addition to the oracle-database-preinstall-18c-1.0-1.el7.x86\_64.rpm package.

```
yum install oracle-database-preinstall-18c --downloadonly
Loaded plugins: aliases, langpacks, ulninfo.
Resolving Dependencies
--> Running transaction check
---> Package oracle-database-preinstall-18c.x86_64 0:1.0-1.el7
will be installed
---> Processing Dependency: compat-libcap1 for package: oracle-database-preinstall-18c-1.0-1.el7.x86_64
---> Processing Dependency: ksh for package: oracle-database-preinstall-18c-1.0-1.el7.x86_64
---> Processing Dependency: libaio-devel for package: oracle-database-preinstall-18c-1.0-1.el7.x86_64
---> Processing Dependency: compat-libstdc++-33 for package: oracle-database-preinstall-18c-1.0-1.el7.x86_64
---> Processing Dependency: libstdc++-devel for package: oracle-database-preinstall-18c-1.0-1.el7.x86_64
---> Running transaction check
---> Package compat-libcap1.x86_64 0:1.10-7.el7 will be
installed
---> Package compat-libstdc++-33.x86_64 0:3.2.3-72.el7 will be
installed
---> Package ksh.x86_64 0:20120801-139.0.1.el7 will be installed
---> Package libaio-devel.x86_64 0:0.3.109-13.el7 will be
installed
---> Package libstdc++-devel.x86_64 0:4.8.5-36.0.1.el7 will be
installed
```

```
--> Processing Dependency: libstdc++(x86-64) = 4.8.5-36.0.1.el7
for package: libstdc++-devel-4.8.5-36.0.1.el7.x86_64
--> Running transaction check
--> Package libstdc++.x86_64 0:4.8.5-28.0.1.el7 will be updated
--> Package libstdc++.x86_64 0:4.8.5-36.0.1.el7 will be an
update
--> Finished Dependency Resolution

Dependencies Resolved

=====
 Package Arch Version
Repository Size
=====
Installing:
 oracle-database-preinstall-18c x86_64 1.0-1.el7
ol7_latest 18 k
Installing for dependencies:
 compat-libcap1 x86_64 1.10-7.el7
ol7_latest 17 k
 compat-libstdc++-33 x86_64 3.2.3-72.el7
ol7_latest 190 k
 ksh x86_64 20120801-139.0.1.el7
ol7_latest 883 k
 libaio-devel x86_64 0.3.109-13.el7
ol7_latest 12 k
 libstdc++-devel x86_64 4.8.5-36.0.1.el7
ol7_latest 1.5 M
Updating for dependencies:
 libstdc++ x86_64 4.8.5-36.0.1.el7
ol7_latest 304 k

Transaction Summary
=====
Install 1 Package (+5 Dependent packages)
Upgrade (1 Dependent package)

Total download size: 2.9 M
Background downloading packages, then exiting:
No Presto metadata available for ol7_latest
(1/7): compat-libcap1-1.10-7.el7.x86_64.rpm | 17 kB 00:00
(2/7): ksh-20120801-139.0.1.el7.x86_64.rpm | 883 kB 00:00
```

```
(3/7) : libaio-devel-0.3.109-13.el7.x86_64.rpm | 12 kB 00:00
(4/7) : libstdc++-4.8.5-36.0.1.el7.x86_64.rpm | 304 kB 00:00
(5/7) : compat-libstdc++-33-3.2.3-72.el7.x86_64.rpm | 190 kB 00:01
(6/7) : oracle-database-preinstall-18c-1.0-1.el7.x86_64.rpm | 18 kB 00:00
(7/7) : libstdc++-devel-4.8.5-36.0.1.el7.x86_64.rpm | 1.5 MB 00:00

Total 1.6 MB/s |
2.9 MB 00:01
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/o17_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/o17_latest/packages
ls
compat-libcap1-1.10-7.el7.x86_64.rpm
compat-libstdc++-33-3.2.3-72.el7.x86_64.rpm
ksh-20120801-139.0.1.el7.x86_64.rpm
libaio-devel-0.3.109-13.el7.x86_64.rpm
libstdc++-4.8.5-36.0.1.el7.x86_64.rpm
libstdc++-devel-4.8.5-36.0.1.el7.x86_64.rpm
oracle-database-preinstall-18c-1.0-1.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
/etc/bash_completion.d
/etc/bash_completion.d/yum-utils.bash
/usr/bin/debuginfo-install
/usr/bin/find-repos-of-install
/usr/bin/needs-restarting
```

```
/usr/bin/package-cleanup
/usr/bin/repo-graph
/usr/bin/repo-rss
/usr/bin/repoclosure
/usr/bin/repodiff
/usr/bin/repomanage
/usr/bin/repoquery
/usr/bin/reposync
/usr/bin/repotrack
/usr/bin/show-changed-rco
/usr/bin/show-installed
/usr/bin/verifytree
/usr/bin/yum-builddep
/usr/bin/yum-config-manager
/usr/bin/yum-debug-dump
/usr/bin/yum-debug-restore
/usr/bin/yum-groups-manager
/usr/bin/yumdownloader
/usr/lib/python2.7/site-packages/yumutils
/usr/lib/python2.7/site-packages/yumutils/__init__.py
...
/usr/sbin/yum-complete-transaction
/usr/sbin/yumdb
/usr/share/doc/yum-utils-1.1.31
/usr/share/doc/yum-utils-1.1.31/COPYING
/usr/share/doc/yum-utils-1.1.31/README
/usr/share/doc/yum-utils-1.1.31/yum-util-cli-template
/usr/share/locale/da/LC_MESSAGES/yum-utils.mo
/usr/share/man/man1/debuginfo-install.1.gz
/usr/share/man/man1/find-repos-of-install.1.gz
...
/usr/share/man/man8/yumdb.8.gz
```

- b. Use the --downloadonly option of the downloadonly plug-in to download the xorg-x11-server-Xorg program.

```
yum install xorg-x11-server-Xorg --downloadonly
Loaded plugins: aliases, langpacks, ulninfo
Resolving Dependencies
--> Running transaction check
--> Package xorg-x11-server-Xorg.x86_64 0:1.19.5-5.el7 will be updated
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for package: xorg-x11-drv-fbdev-0.4.3-25.el7.x86_64
```

```
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-dummy-0.3.7-1.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-vesa-2.3.2-25.1.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-v4l-0.2.0-47.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-vmware-13.2.1-1.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: 1:xorg-x11-drv-nouveau-1.0.13-3.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-intel-2.99.917-27.20160929.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-ati-7.10.0-1.el7.x86_64
--> Processing Dependency: xserver-abi(videodrv-23) >= 0 for
package: xorg-x11-drv-qxl-0.1.5-3.el7.x86_64
---> Package xorg-x11-server-Xorg.x86_64 0:1.20.1-5.2.el7_6 will
be an update
--> Processing Dependency: xorg-x11-server-common >= 1.20.1-
5.2.el7_6 for package: xorg-x11-server-Xorg-1.20.1-
5.2.el7_6.x86_64
--> Running transaction check
---> Package xorg-x11-drv-ati.x86_64 0:7.10.0-1.el7 will be
updated
---> Package xorg-x11-drv-ati.x86_64 0:18.1.0-1.el7_6 will be an
update
---> Package xorg-x11-drv-dummy.x86_64 0:0.3.7-1.el7 will be
updated
---> Package xorg-x11-drv-dummy.x86_64 0:0.3.7-1.el7.1 will be
an update
---> Package xorg-x11-drv-fbdev.x86_64 0:0.4.3-25.el7 will be
updated
---> Package xorg-x11-drv-fbdev.x86_64 0:0.5.0-1.el7 will be an
update
---> Package xorg-x11-drv-intel.x86_64 0:2.99.917-
27.20160929.el7 will be updated
---> Package xorg-x11-drv-intel.x86_64 0:2.99.917-
28.20180530.el7 will be an update
---> Package xorg-x11-drv-nouveau.x86_64 1:1.0.13-3.el7 will be
updated
---> Package xorg-x11-drv-nouveau.x86_64 1:1.0.15-1.el7 will be
an update
---> Package xorg-x11-drv-qxl.x86_64 0:0.1.5-3.el7 will be
updated
---> Package xorg-x11-drv-qxl.x86_64 0:0.1.5-4.el7.1 will be an
update
```

```
---> Package xorg-x11-drv-v41.x86_64 0:0.2.0-47.el7 will be updated
---> Package xorg-x11-drv-v41.x86_64 0:0.2.0-49.el7 will be an update
---> Package xorg-x11-drv-vesa.x86_64 0:2.3.2-25.1.el7 will be updated
---> Package xorg-x11-drv-vesa.x86_64 0:2.4.0-1.el7 will be an update
---> Package xorg-x11-drv-vmware.x86_64 0:13.2.1-1.el7 will be updated
---> Package xorg-x11-drv-vmware.x86_64 0:13.2.1-1.el7.1 will be an update
---> Package xorg-x11-server-common.x86_64 0:1.19.5-5.el7 will be updated
---> Package xorg-x11-server-common.x86_64 0:1.20.1-5.2.el7_6 will be an update
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
 Package Arch Version
Repository Size
=====
Updating:
 xorg-x11-server-Xorg x86_64 1.20.1-5.2.el7_6
ol7_latest 1.4 M
Updating for dependencies:
 xorg-x11-drv-ati x86_64 18.1.0-1.el7_6
ol7_latest 164 k
 xorg-x11-drv-dummy x86_64 0.3.7-1.el7.1
ol7_latest 17 k
 xorg-x11-drv-fbdev x86_64 0.5.0-1.el7
ol7_latest 20 k
 xorg-x11-drv-intel x86_64 2.99.917-28.20180530.el7
ol7_latest 675 k
 xorg-x11-drv-nouveau x86_64 1:1.0.15-1.el7
ol7_latest 95 k
 xorg-x11-drv-qxl x86_64 0.1.5-4.el7.1
ol7_latest 92 k
 xorg-x11-drv-v41 x86_64 0.2.0-49.el7
ol7_latest 21 k
 xorg-x11-drv-vesa x86_64 2.4.0-1.el7
ol7_latest 25 k
 xorg-x11-drv-vmware x86_64 13.2.1-1.el7.1
ol7_latest 82 k
```

```

xorg-x11-server-common x86_64 1.20.1-5.2.el7_6
ol7_latest 53 k

Transaction Summary
=====
Upgrade 1 Package (+10 Dependent packages)

Total download size: 2.7 M
Background downloading packages, then exiting:
No Presto metadata available for ol7_latest
(1/11): xorg-x11-drv-ati-18.1.0-1.el7_6.x86_64.rpm | 164
kB 00:00
(2/11): xorg-x11-drv-fbdev-0.5.0-1.el7.x86_64.rpm | 20
kB 00:00
(3/11): xorg-x11-drv-dummy-0.3.7-1.el7.1.x86_64.rpm | 17
kB 00:00
(4/11): xorg-x11-drv-nouveau-1.0.15-1.el7.x86_64.rpm | 95
kB 00:00
(5/11): xorg-x11-drv-intel-2.99.917-28.20180530.el7.x86_64 | 675
kB 00:00
(6/11): xorg-x11-drv-qxl-0.1.5-4.el7.1.x86_64.rpm | 92
kB 00:00
(7/11): xorg-x11-drv-v4l-0.2.0-49.el7.x86_64.rpm | 21
kB 00:00
(8/11): xorg-x11-drv-vesa-2.4.0-1.el7.x86_64.rpm | 25
kB 00:00
(9/11): xorg-x11-drv-vmware-13.2.1-1.el7.1.x86_64.rpm | 82
kB 00:00
(10/11): xorg-x11-server-common-1.20.1-5.2.el7_6.x86_64.rp | 53
kB 00:00
(11/11): xorg-x11-server-Xorg-1.20.1-5.2.el7_6.x86_64.rpm | 1.4
MB 00:00

Total 1.5 MB/s |
2.7 MB 00:01
exiting because "Download Only" specified

```

- 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: langpacks
```

- 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, ulninfo
package: xorg-x11-server-Xorg.x86_64 1.20.1-5.2.el7_6
 dependency: config(xorg-x11-server-Xorg) = 1.20.1-5.2.el7_6
 provider: xorg-x11-server-Xorg.x86_64 1.20.1-5.2.el7_6
 dependency: libGL.so.1()(64bit)
 provider: mesa-libGL.x86_64 17.2.3-8.20171019.el7
 provider: libglvnd-glx.x86_64 1:1.0.1-0.8.git5baale5.el7
 dependency: libXau.so.6()(64bit)
 provider: libXau.x86_64 1.0.8-2.1.el7
...
 dependency: libcrypto.so.10()(64bit)
 provider: openssl-libs.x86_64 1:1.0.2k-16.0.1.el7
 dependency: libcrypto.so.10(libcrypto.so.10)(64bit)
 provider: openssl-libs.x86_64 1:1.0.2k-16.0.1.el7
 dependency: libdl.so.2()(64bit)
 provider: glibc.x86_64 2.17-260.0.15.el7_6.3
 dependency: libdl.so.2(GLIBC_2.2.5)(64bit)
 provider: glibc.x86_64 2.17-260.0.15.el7_6.3
 dependency: libdrm.so.2()(64bit)
 provider: libdrm.x86_64 2.4.91-3.el7
 dependency: libepoxy.so.0()(64bit)
 provider: libepoxy.x86_64 1.5.2-1.el7
 dependency: libgbm.so.1()(64bit)
 provider: mesa-libgbm.x86_64 18.0.5-4.el7_6
 dependency: libm.so.6()(64bit)
...
 dependency: libudev.so.1(LIBUDEV_183)(64bit)
 provider: systemd-libs.x86_64 219-62.0.4.el7_6.3
```

```

dependency: libxshmfence.so.1()(64bit)
 provider: libxshmfence.x86_64 1.2-1.el7
dependency: rtld(GNU_HASH)
 provider: glibc.x86_64 2.17-260.0.15.el7_6.3
 provider: glibc.i686 2.17-260.0.15.el7_6.3
dependency: system-setup-keyboard
 provider: systemd.x86_64 219-62.0.4.el7_6.3
dependency: xorg-x11-server-common >= 1.20.1-5.2.el7_6
 provider: xorg-x11-server-common.x86_64 1.20.1-5.2.el7_6

```

- 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-drivers-0:7.7-6.el7.x86_64
xorg-x11-drv-ati-0:7.10.0-1.el7.x86_64
xorg-x11-drv-ati-0:18.0.1-1.el7.x86_64
xorg-x11-drv-ati-0:18.1.0-1.el7_6.x86_64
xorg-x11-drv-dummy-0:0.3.7-1.el7.x86_64
xorg-x11-drv-dummy-0:0.3.7-1.el7.1.x86_64
xorg-x11-drv-evdev-0:2.10.5-2.1.el7.x86_64
...
xorg-x11-drv-v4l-0:0.2.0-47.el7.x86_64
xorg-x11-drv-v4l-0:0.2.0-49.el7.x86_64
xorg-x11-drv-vesa-0:2.3.2-25.1.el7.x86_64
xorg-x11-drv-vesa-0:2.4.0-1.el7.x86_64
xorg-x11-drv-vmmouse-0:13.1.0-1.el7.x86_64
xorg-x11-drv-vmmouse-0:13.1.0-1.el7.1.x86_64
xorg-x11-drv-vmware-0:13.2.1-1.el7.x86_64
xorg-x11-drv-vmware-0:13.2.1-1.el7.1.x86_64
xorg-x11-drv-void-0:1.4.1-2.el7.x86_64
xorg-x11-drv-void-0:1.4.1-2.el7.1.x86_64
xorg-x11-drv-wacom-0:0.34.2-4.el7.x86_64
xorg-x11-drv-wacom-0:0.34.2-5.el7.x86_64
xorg-x11-drv-wacom-0:0.36.1-1.el7.x86_64
xorg-x11-server-Xorg-0:1.20.1-5.2.el7_6.x86_64

```

## Practice 8-4: Creating a binary 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 `rpmdevtools` command.
    - In this example, the `rpmdevtools` package is not installed.
  - 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
...
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 `rpmbuilder` 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.
- 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/
```

- 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
```

- Use the `ls` command to verify that the new `.tar.gz` file is in the `SOURCES` directory.

```
$ ls SOURCES
hello-1.0.tar.gz
```

- Create and populate the `spec` file.

- 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.
```

- 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
```

- Use the `cd` command to change to the `SPECS` directory.

```
$ cd SPECS
```

- Use the `vi` editor to edit the `hello.spec` file and populate the header section by making the following changes:

- 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:

```
$ vi 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
```

- 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	
<del>%configure</del>	delete this line
<del>make %{?_smp_mflags}</del>	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

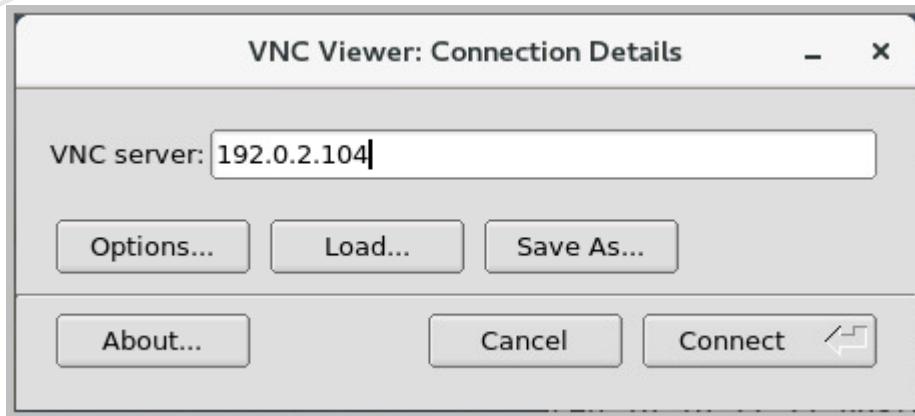
- You are the `root` user on **host04**.
- You need to log on to **host04** by using VNC Viewer.

### Tasks

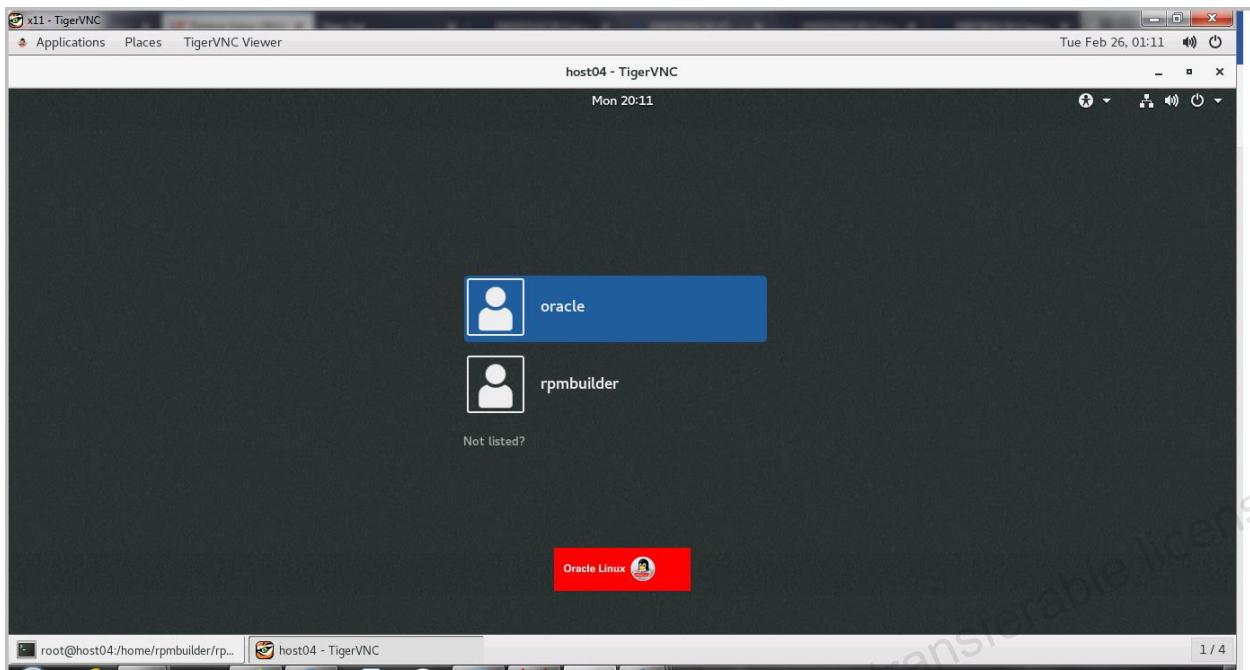
1. Log on to **host04** by using VNC Viewer.
  - a. Double-click the “VNC Viewer” icon on the **gateway** desktop.



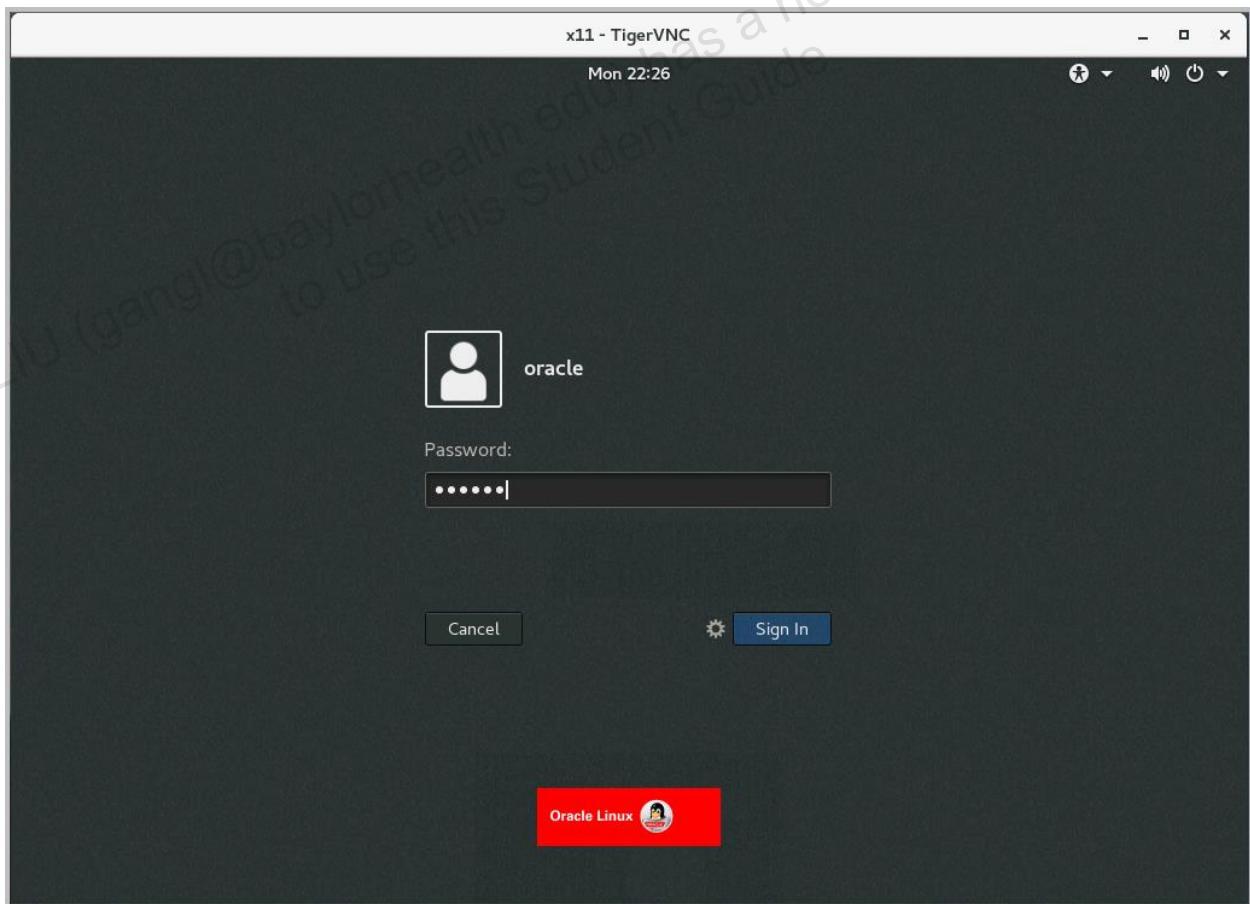
- b. Enter `192.0.2.104` as shown in the following window.
  - Click Connect.



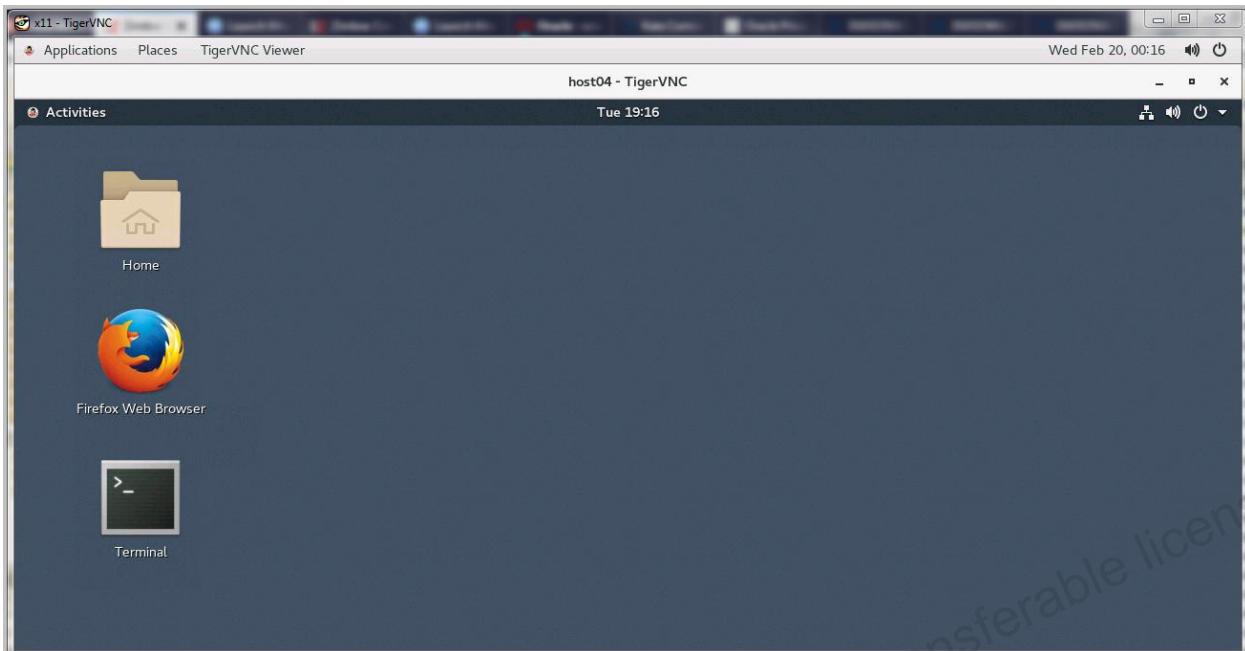
- The login window appears.
- Click in the window to see the `oracle` user.



- c. Click the `oracle` user and login.

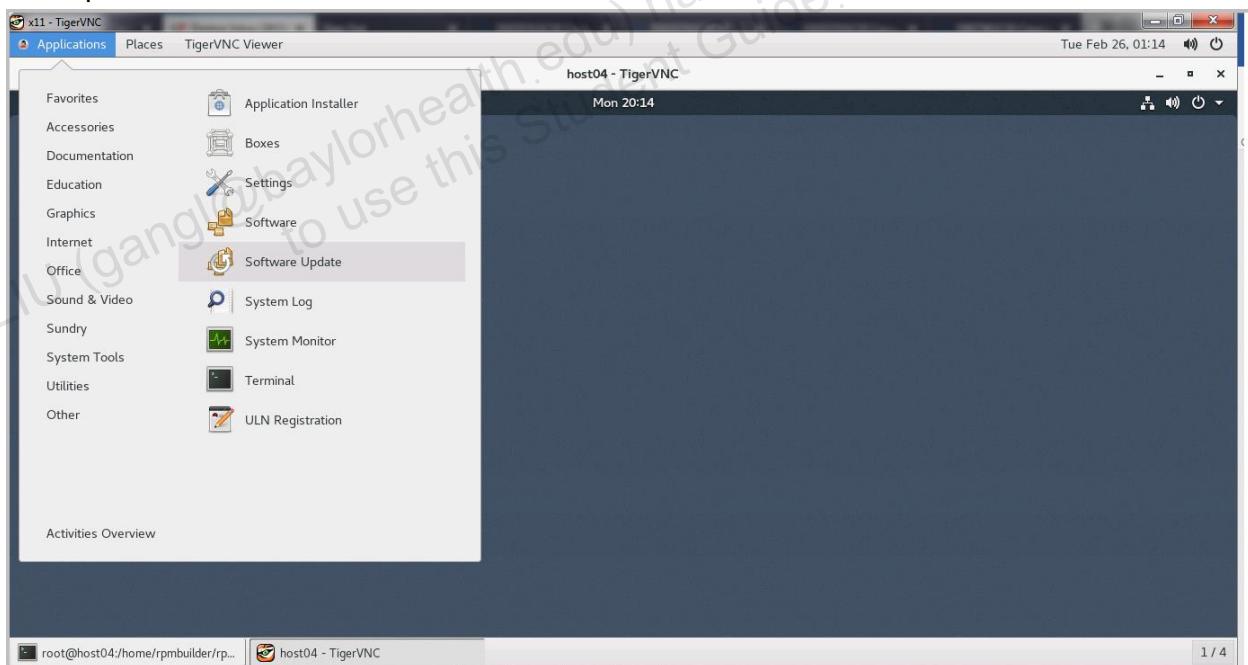


- The desktop appears as shown.

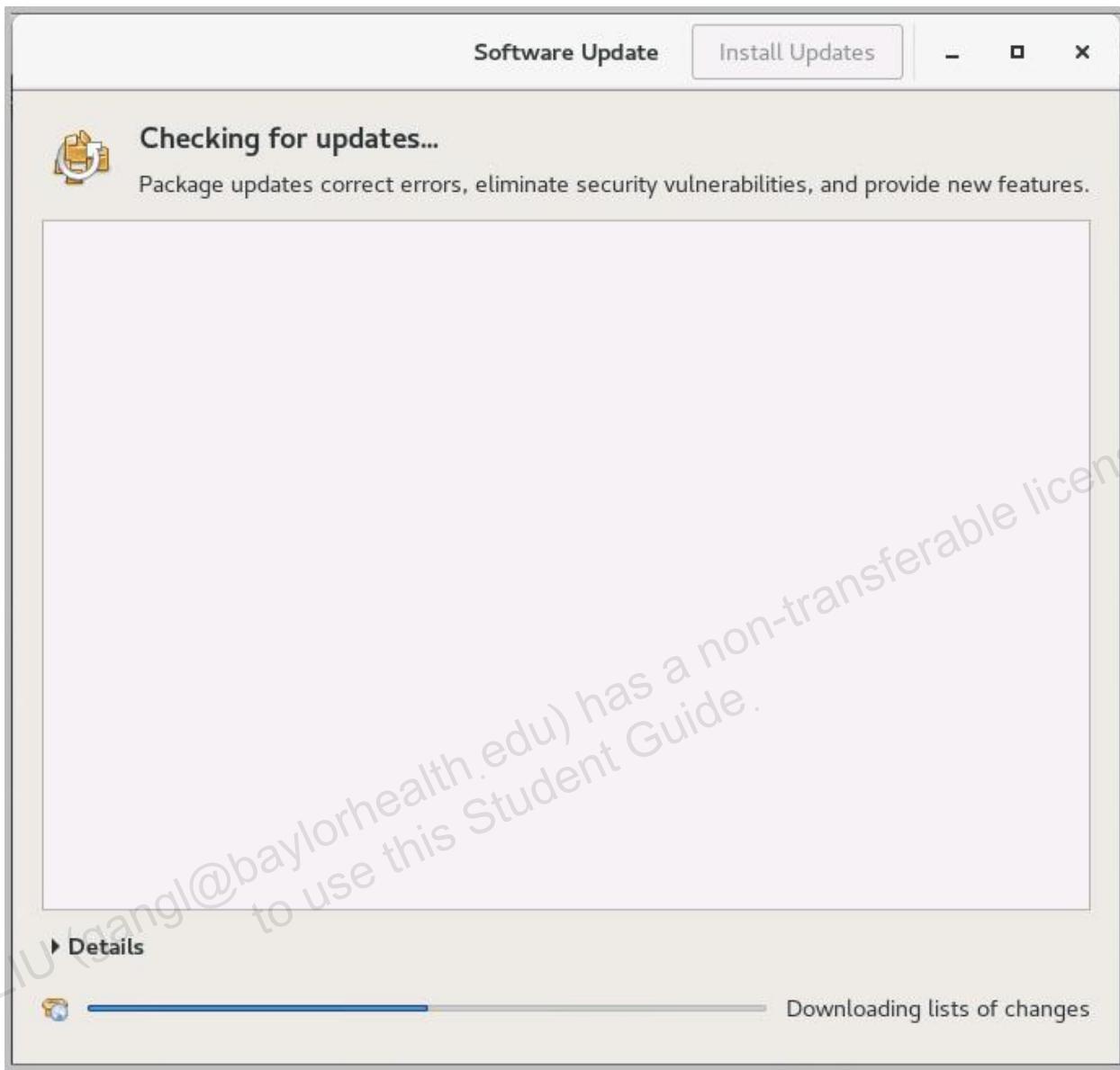


2. Launch Software Update.

- In the **host04** desktop task bar, select “Application > System Tools > Software Update.”



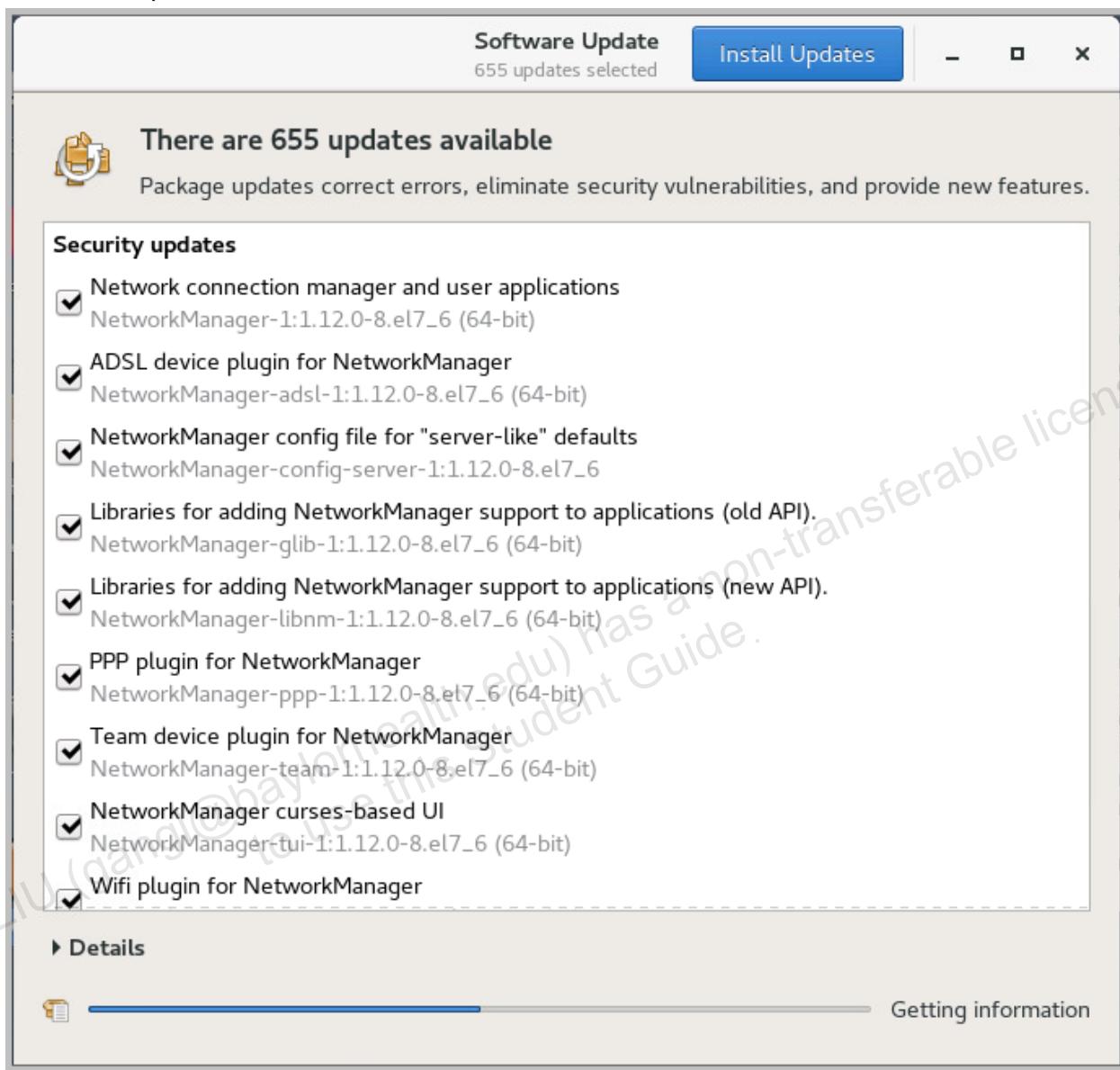
- The Software Update window appears.
- “Checking for updates” might take several minutes to complete.



3. View the update(s) flagged by the Software Update program.
  - If the Software Update program fails with an error message, run the following commands from a terminal window on **host04**.
  - 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 655 updates.
  - This is sample output. Your environment might be different because updates have been added since this example was captured.

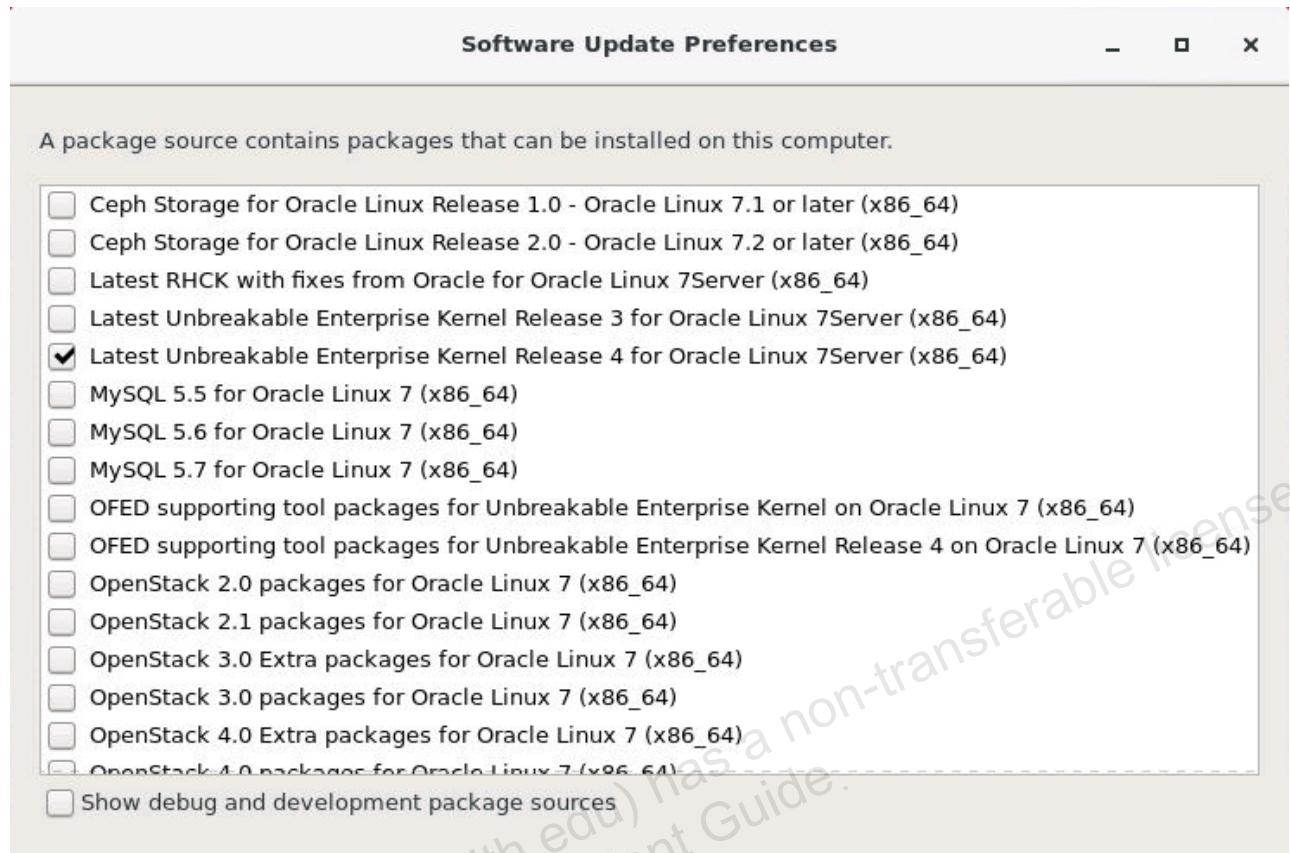
- **Do not click the “Install Updates” button** because it takes too long to install all of the updates.



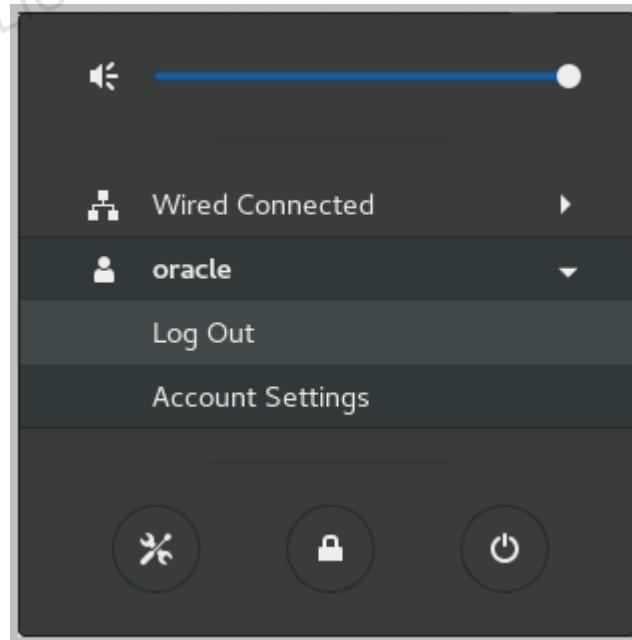
- b. Scroll down through the list of updates.
  - Note that there are “Security updates,” “Bug fix updates,” and “Other updates.”
  - c. Click “x” to exit the Software Update program.
4. Run the Software Update Preferences program.
  - a. From a terminal window as the `root` user, run the `gpk-prefs` command to view the Software Update Preferences GUI.

```
gpk-prefs
```

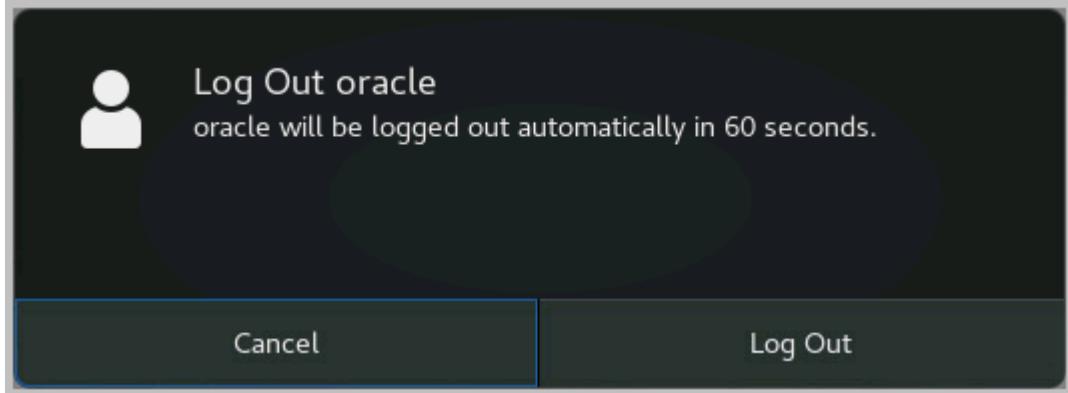
- The “Software Update Preferences” window appears.
- In this window, you can see and select Yum repositories.



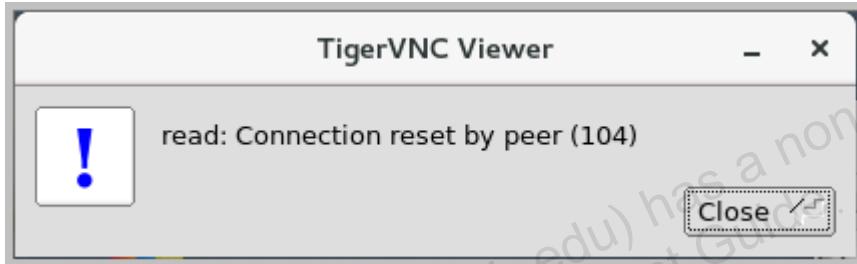
- b. Click “x” to close the “Software Update Preferences” window.
5. Log off **host04** and close the VNC connection.
  - a. Click the down arrow in the top-right corner of the window to display the drop-down menu.
  - b. Click the `oracle` user, then click Log Out as shown.



- c. Click Log Out as shown.



- The **host04** desktop closes.
- The following window might appear. Close the window by clicking the X in the upper-right corner.



## 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

- If necessary, log in to **host04**.

- As the `root` user on **gateway**, use the `ssh` command to log in to **host04**.

```
[gateway]# ssh host04
root@host04's password:
Last login: ...
[root@host04 ~]#
```

- Display Yum history information.

- 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, ulninfo
ID | Login user | Date and time | Action(s)
| Altered

| 12 | root <root> | 2019-02-25 19:46 | I, U
| 11
| 11 | root <root> | 2019-02-25 19:23 | Update
| 17 EE
| 10 | root <root> | 2019-02-25 19:10 | Install
| 1
```

```

 9 | root <root> | 2019-02-25 19:09 | Erase
 |
 1
 8 | root <root> | 2019-02-25 19:07 | Install
 |
 1
 7 | root <root> | 2019-02-18 01:49 | I, U
 |
 11
 6 | root <root> | 2019-02-13 09:55 | I, O, U
 |
 28 EE
 5 | root <root> | 2019-02-13 09:49 | Update
 |
 1
 4 | root <root> | 2019-02-13 09:49 | Update
 |
 1
 3 | root <root> | 2019-02-13 09:47 | Install
 |
 1
 2 | root <root> | 2019-02-13 09:42 | Install
 |
 1
 1 | System <unset> | 2019-02-13 06:19 | Install
 |
 1328
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 12.

```

yum history info 12
Loaded plugins: aliases, langpacks, ulninfo
Transaction ID : 12
Begin time : Mon Feb 25 19:46:20 2019
Begin rpmbdb : 1340:b2c0a0560519f0234de09f2abcd660c275fb62b6
End time : 19:46:21 2019 (1 seconds)
End rpmbdb : 1347:ba3fe5bbcbe3d737d3c7e075aae02da9ff385d4f
**
User : root <root>
Return-Code : Success
Command Line : install rpmdevtools
Transaction performed with:
 Updated rpm-4.11.3-32.el7.x86_64
@anaconda/7.5
 Installed yum-3.4.3-158.0.1.el7.noarch
@anaconda/7.5
 Installed yum-metadata-parser-1.1.4-10.el7.x86_64
@anaconda/7.5
 Installed yum-plugin-aliases-1.1.31-50.0.1.el7.noarch
@ol7_latest
Packages Altered:
 Dep-Install dwz-0.11-3.el7.x86_64
@ol7_latest

```

```

Dep-Install patch-2.7.1-10.el7_5.x86_64
@el7_latest

Dep-Install perl-Thread-Queue-3.02-2.el7.noarch
@el7_latest

Dep-Install perl-srpm-macros-1-8.el7.noarch
@el7_latest

Dep-Install redhat-rpm-config-9.1.0-87.0.1.el7.noarch
@el7_latest

Updated rpm-4.11.3-32.el7.x86_64
@anaconda/7.5

Update 4.11.3-35.el7.x86_64
@el7_latest

Dep-Install rpm-build-4.11.3-35.el7.x86_64
@el7_latest

Updated rpm-build-libs-4.11.3-32.el7.x86_64
@anaconda/7.5

Update 4.11.3-35.el7.x86_64
@el7_latest

Updated rpm-libs-4.11.3-32.el7.x86_64
@anaconda/7.5

Update 4.11.3-35.el7.x86_64
@el7_latest

Updated rpm-python-4.11.3-32.el7.x86_64
@anaconda/7.5

Update 4.11.3-35.el7.x86_64
@el7_latest

Install rpmdevtools-8.3-5.0.1.el7.noarch
@el7_latest

history info

```

3. 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
...
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, ulninfo

```

ID	Login user	Date and time	Action(s)
Altered			
<hr/>			
13   root <root>		2019-02-25 20:28	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 13
...
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-plugin-changelog` package are uninstalled when you use the `yum history undo` command.

```
yum history list
Loaded plugins: aliases, langpacks, ulninfo
ID | Login user | Date and time | Action(s)
| Altered

14 | root <root> | 2019-02-25 20:30 | Erase
| 2
13 | root <root> | 2019-02-25 20:28 | Install
| 2 <
...
history list
```

#### 4. 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. 4 root root 4096 ... ol7_latest
drwxr-xr-x. 4 root root 4096 ... ol7_u4_base
drwxr-xr-x. 4 root root 4096 ... ol7_UEKR4
-rw-r--r--. 1 root root 36 ... timedhosts
```

- 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 Linux 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
total 23868
-rw-r--r--. 1 root root 0 ... cachecookie
-rw-r--r--. 1 root root 675267 ... comps.xml
-rw-r--r--. 1 root root 10853376 ... filelists.xml.gz
drwxr-xr-x. 2 root root 4096 ... gen
drwxr-xr-x. 2 root root 4096 ... packages
-rw-r--r--. 1 root root 12061485 ... primary.xml.gz
-rw-r--r--. 1 root root 1429 ... repomd.xml
-rw-r--r--. 1 root root 814969 ... updateinfo.xml.gz
```

## 5. Clean the Yum cache.

- Use the `yum clean packages` command to clean the packages in the Yum cache.
  - The number of package files removed might differ in your environment.

```
yum clean packages
Loaded plugins: aliases, langpacks, ulninfo
Cleaning repos: ol7_UEKR4 ol7_latest
18 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
total 358232
-rw-r--r--. 1 root root 132486516 ... filelists.xml
-rw-r--r--. 1 root root 71178240 ... filelists.xml.sqlite
-rw-r--r--. 1 root root 70523686 ... primary.xml
-rw-r--r--. 1 root root 87933952 ... primary.xml.sqlite
-rw-r--r--. 1 root root 4696080 ... 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, langpacks, ulninfo
Cleaning repos: ol7_UEKR4 ol7_latest
17 metadata files removed
4 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 Linux 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 Linux 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
```

- f. Use the `exit` command to log off **host04**.

```
exit
logout
Connection to host04 closed.
```

## **Practices for Lesson 9: Advanced Networking**

## Practices for Lesson 9: Overview

---

### Overview

In these practices, you do the following:

- Configure network bonding by using the TUI and the command line.
- Explore network bonding interface configuration.
- Configure 802.1q VLAN tagging interfaces by using the GUI and the command line.
- Explore 802.1q VLAN tagging interface configuration.
- Configure a site-to-site VPN.

## Practice 9-1: Configuring Network Bonding by Using the TUI

---

### Overview

In this practice, you:

- View the network configuration on **host02**
- Configure network bonding on **host02** by using the `nmtui` TUI

### Assumptions

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

### Tasks

#### 1. Log in to **host02**.

- As the `root` user on **gateway**, use the `ssh` command to log in to **host02**.

```
[root@gateway ~]# ssh host02
root@host02's password:
Last login: ...
[root@host02 ~]#
```

#### 2. 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 state UNKNOWN group default
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 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
 noprefixroute eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 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
 noprefixroute eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 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 pfifo_fast state UP group default qlen 1000
```

```
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.

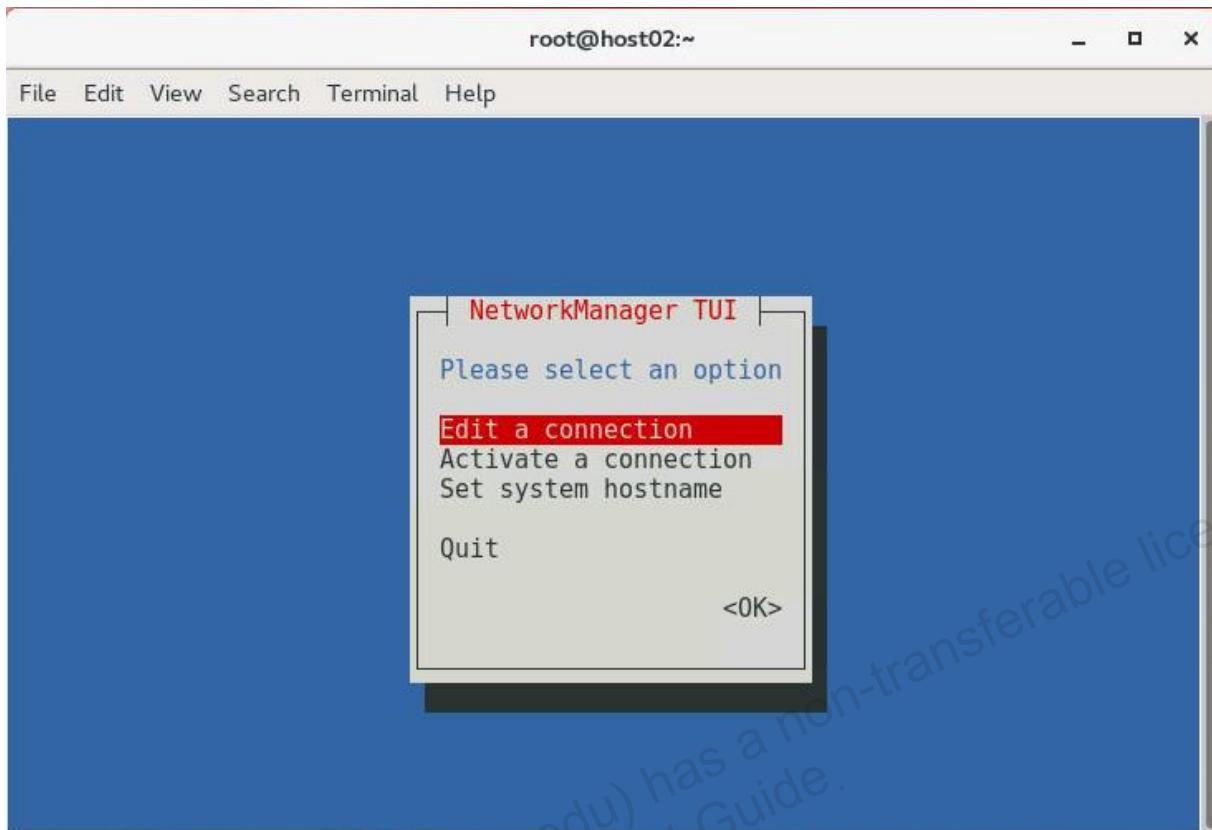
```
nmcli con
NAME UUID TYPE DEVICE
eth0 ... ethernet eth0
eth1 ... ethernet eth1
...
eth2 ... ethernet --
eth3 ... ethernet --
```

3. Use the text user interface tool `nmtui` to configure network bonding.

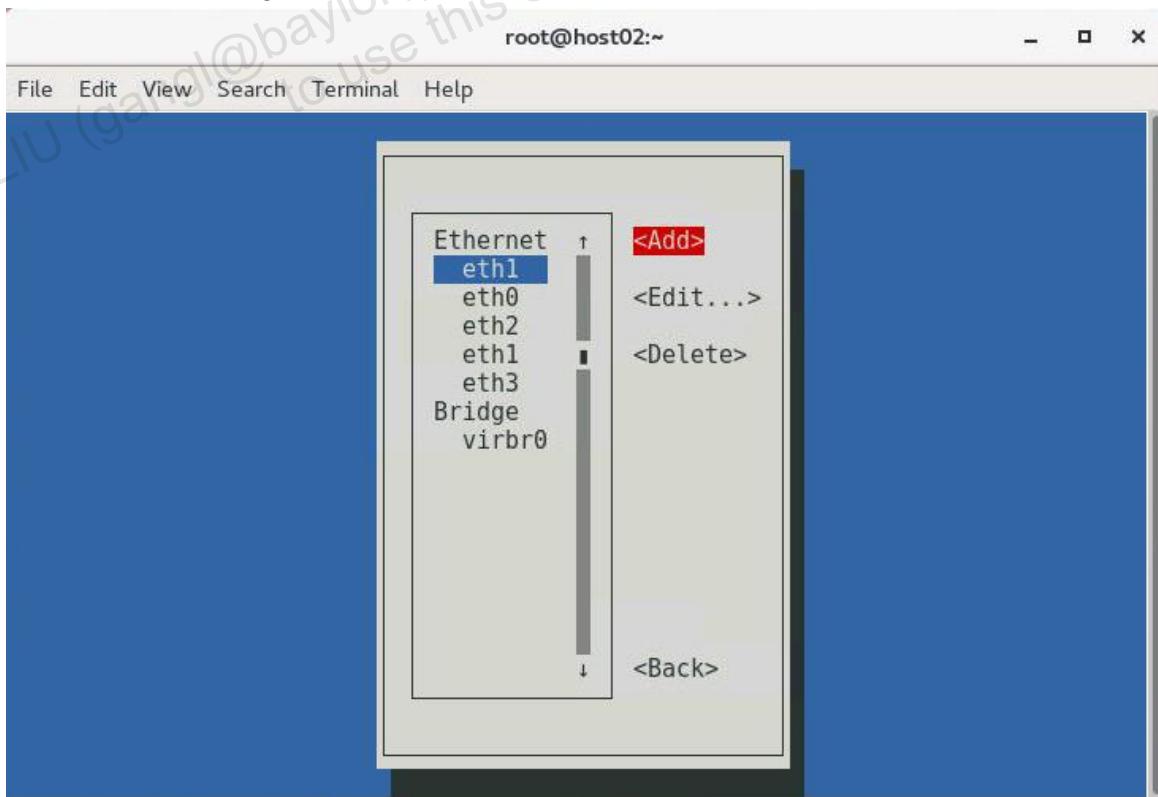
- a. From the terminal, run the `nmtui` command.

```
nmtui
```

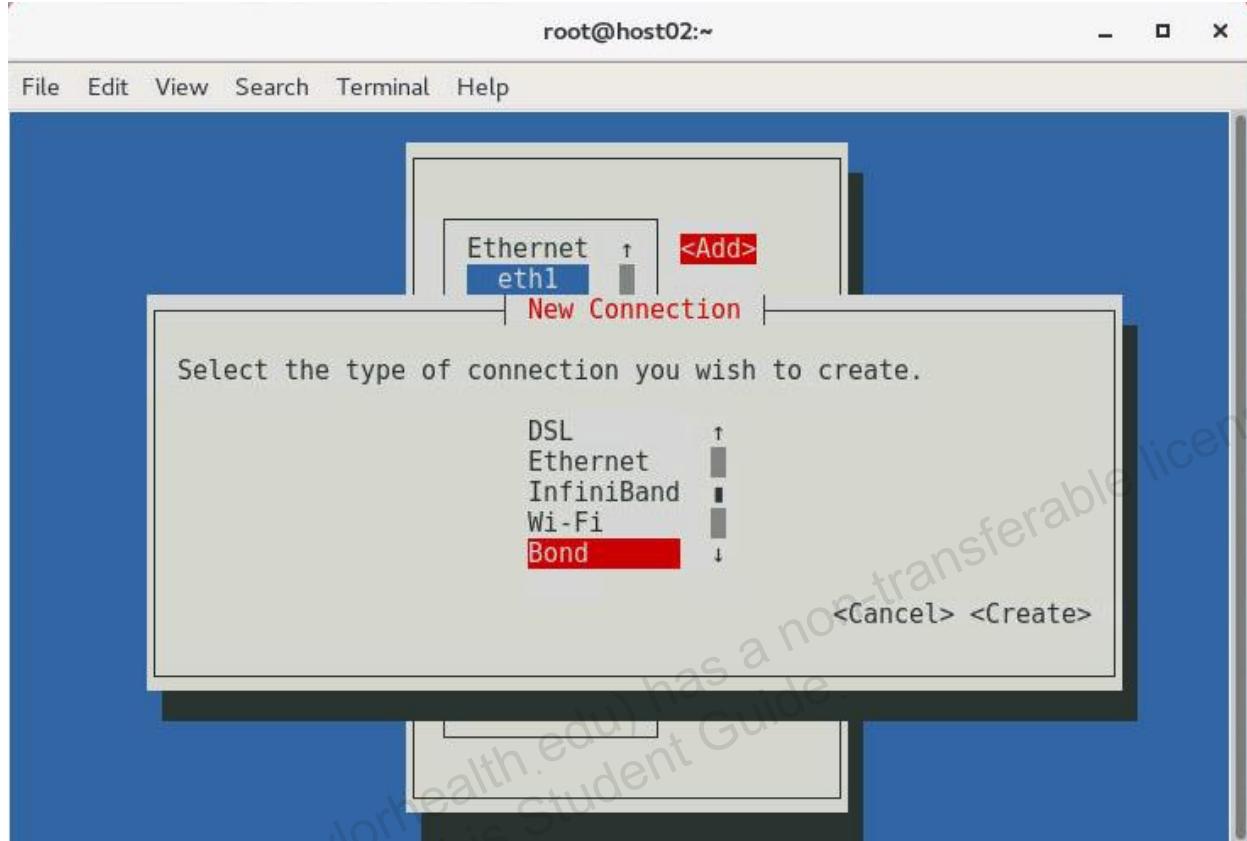
- The “NetworkManager TUI” window appears.



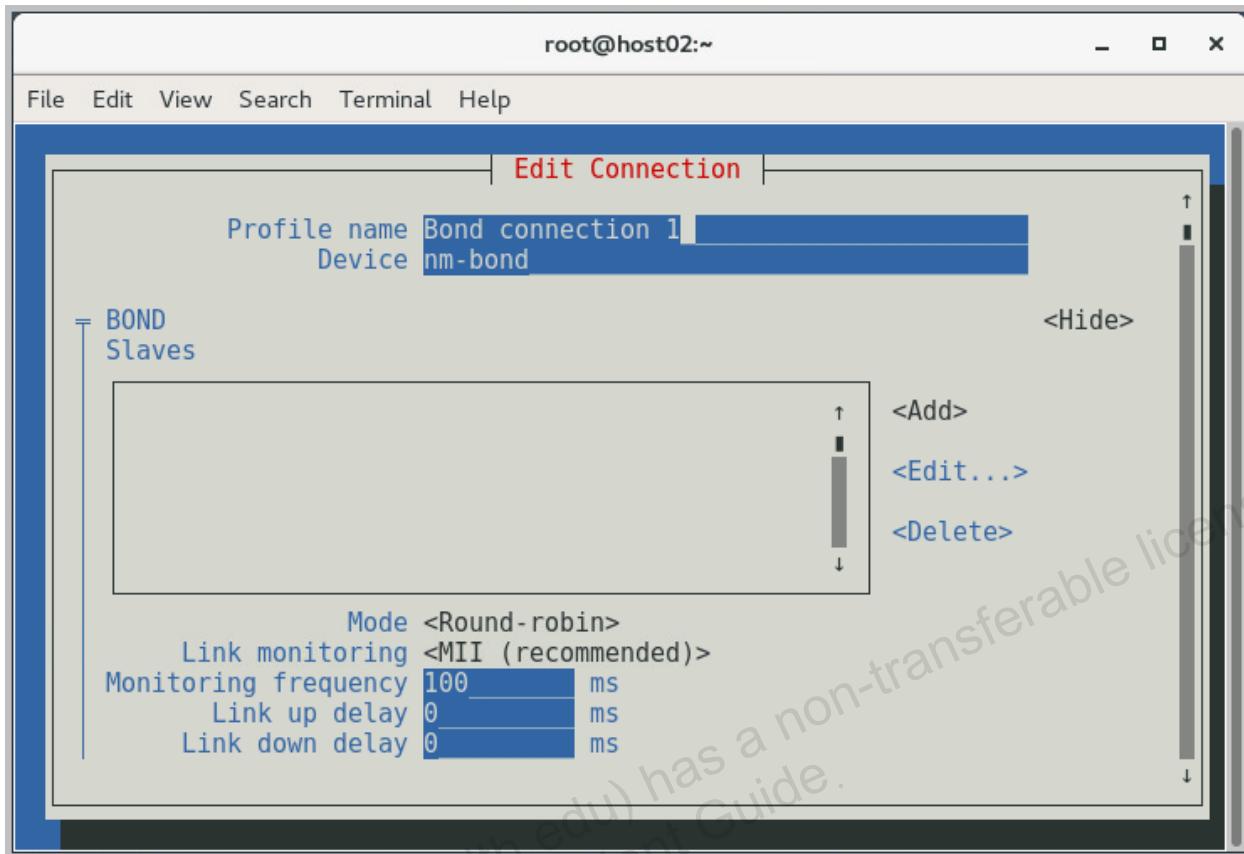
- b. Select “Edit a connection” and press Enter.  
• The following window appears.



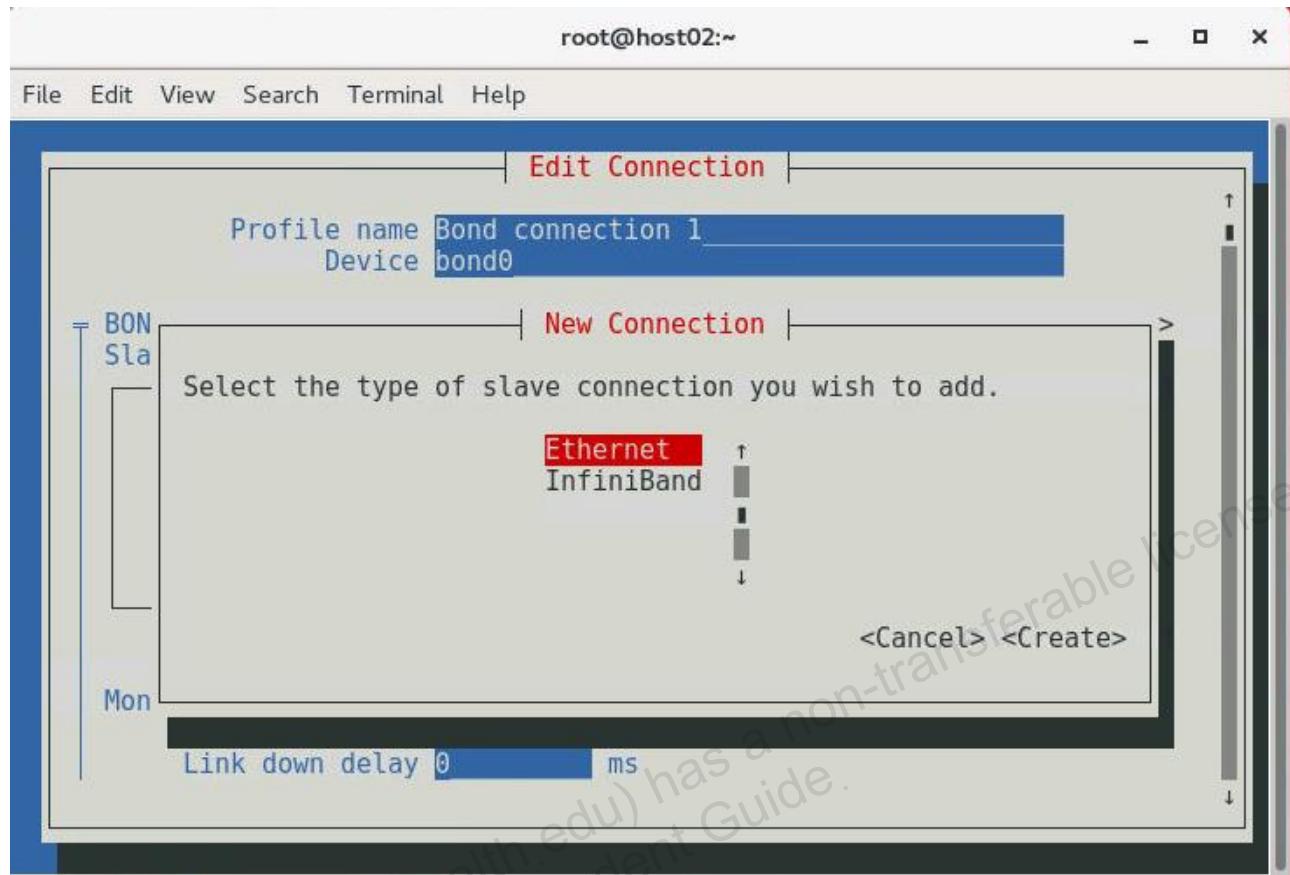
- c. Use the right arrow to select “<Add>” and press Enter.  
• The “New Connection” window appears.



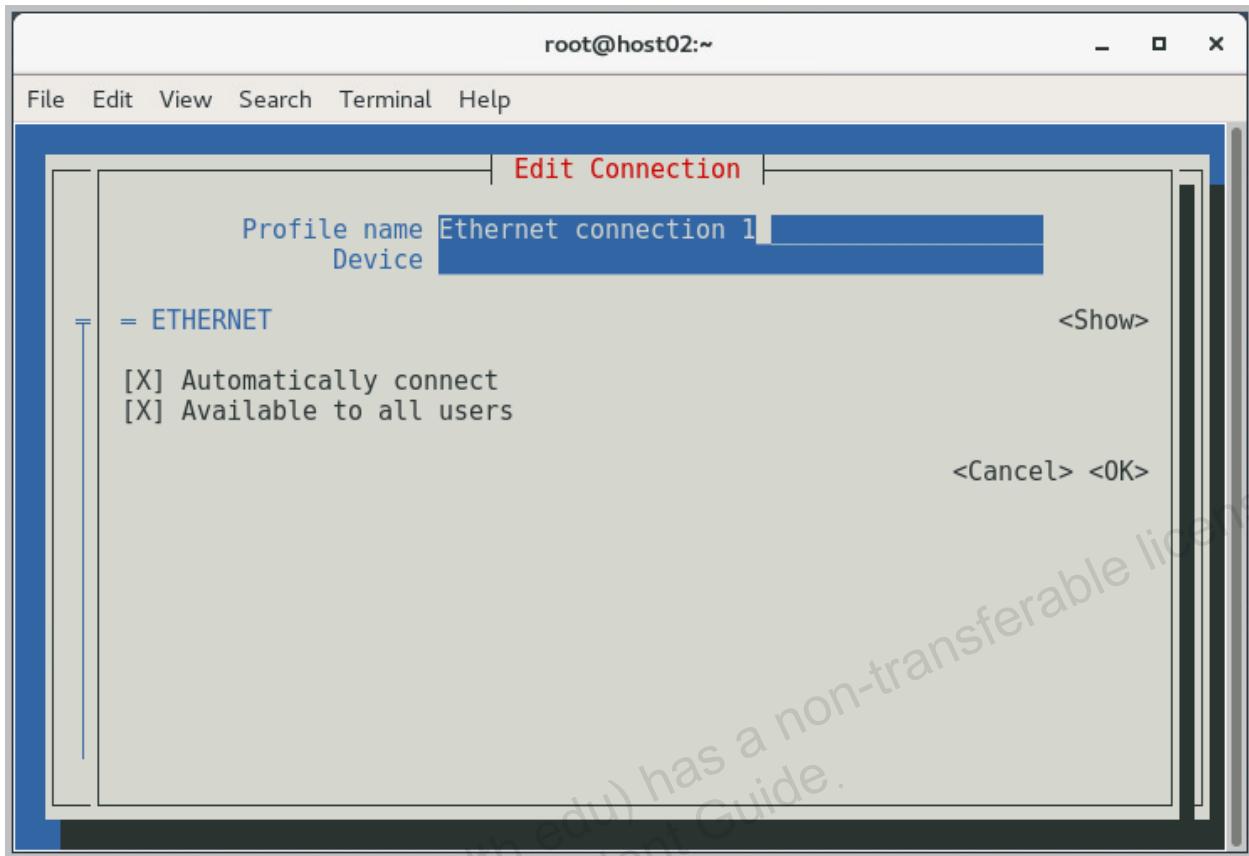
- d. Arrow down to select “Bond” to add a bonded interface.  
• Tab to select “Create” then press Enter.  
• The following window appears.  
• The default Connection name is “Bond connection 1.”  
• The default interface name is nm-bond.



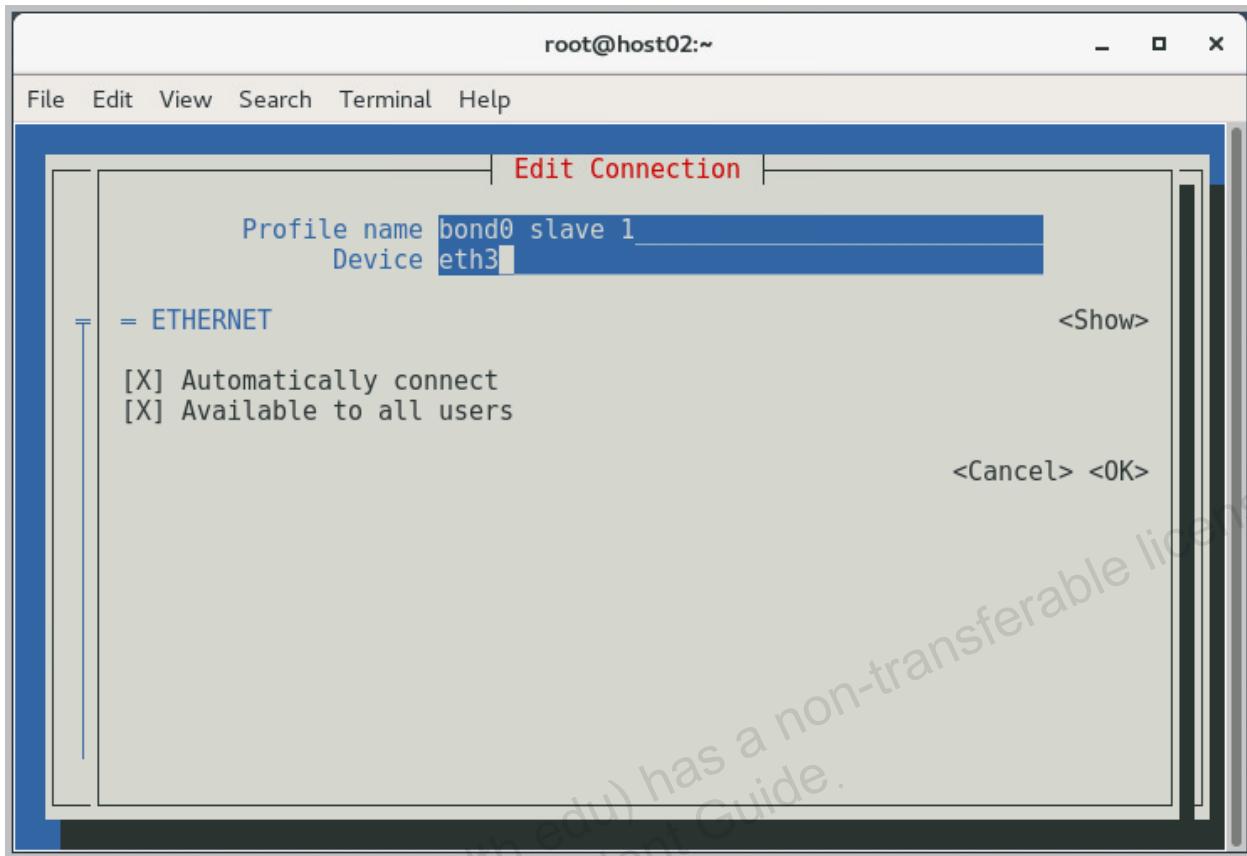
- e. Edit the Device name to rename to bond0.
- f. Tab to select the "Add" button then press Enter to add the slave interfaces to the bond.
  - The following window appears.



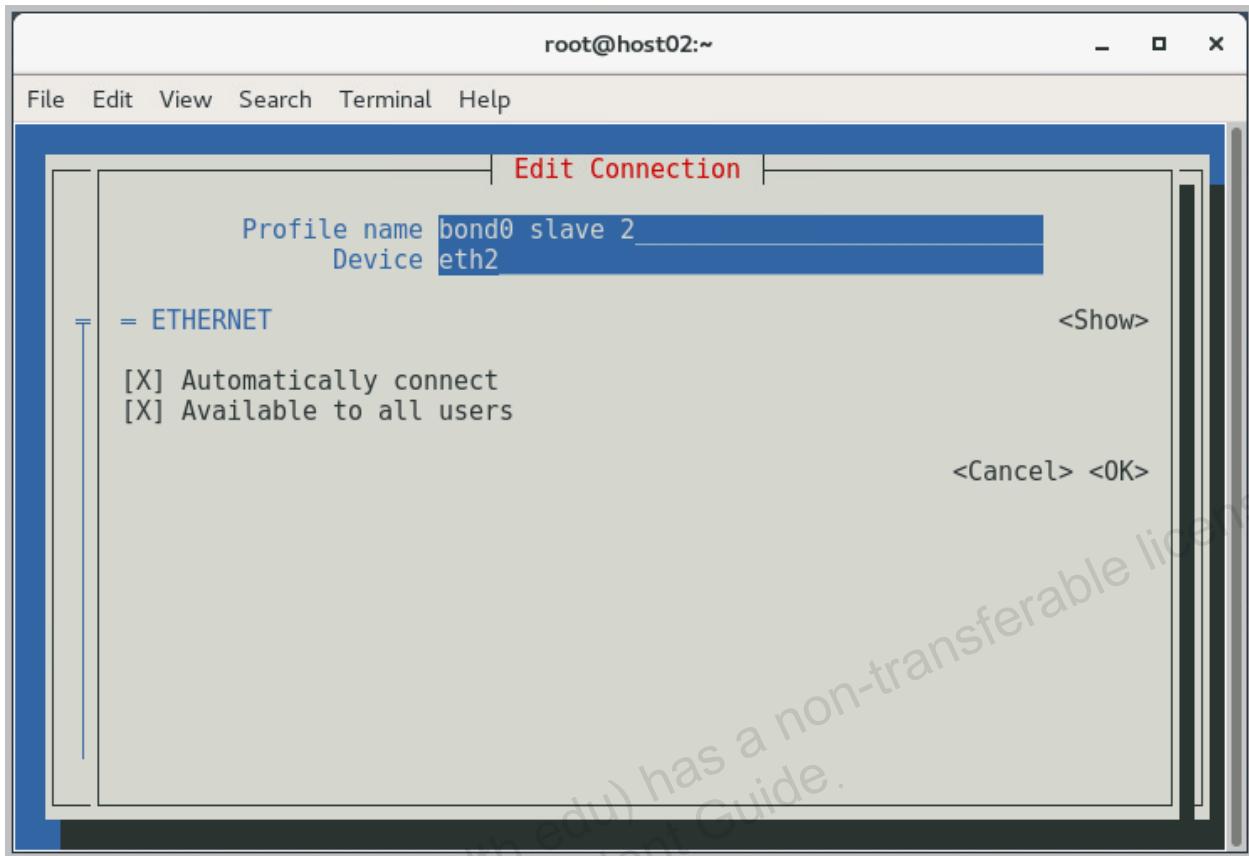
- g. Accept the default “Ethernet” selection. Tab to select “Create” then press Enter to display the following window.



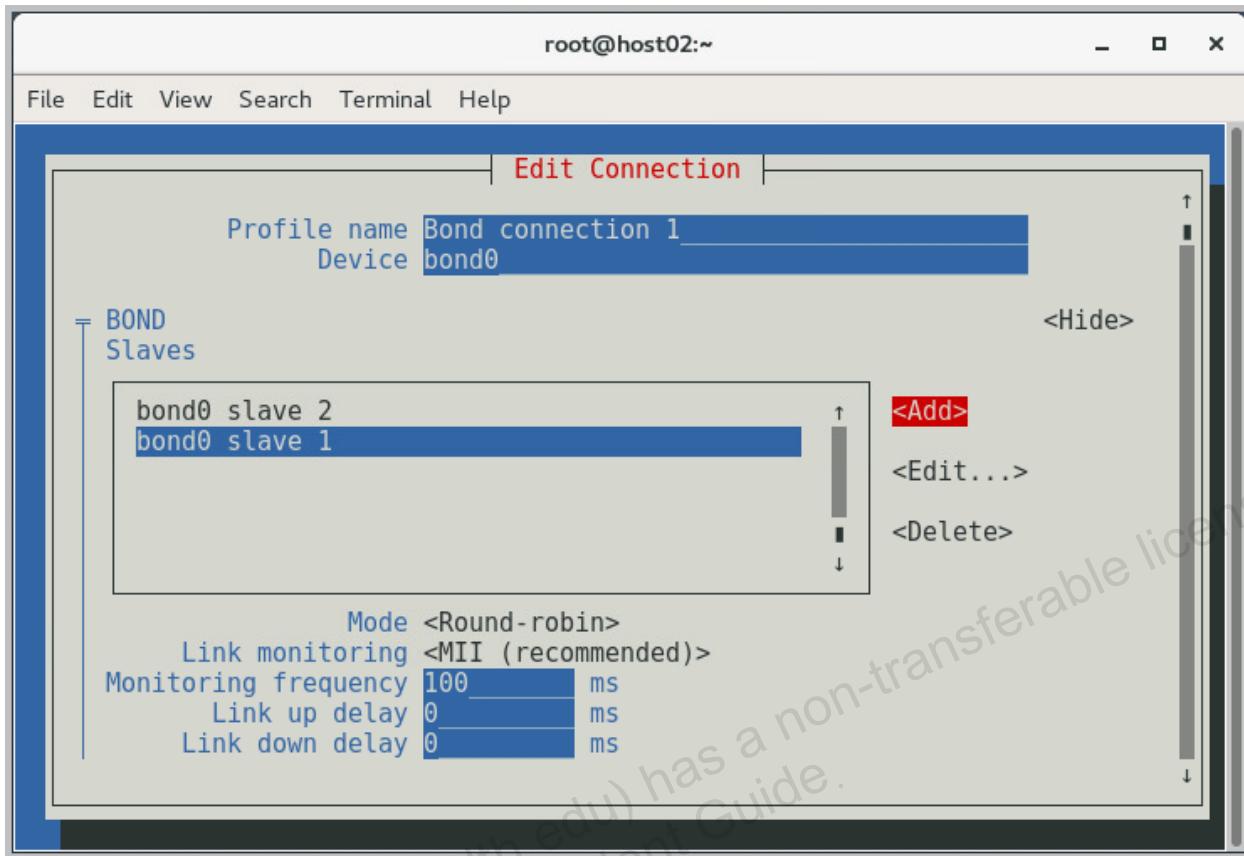
- h. Edit the Profile name field to bond0 slave 1.
- i. Enter the eth3 device for the Device.
  - The window appears as shown.



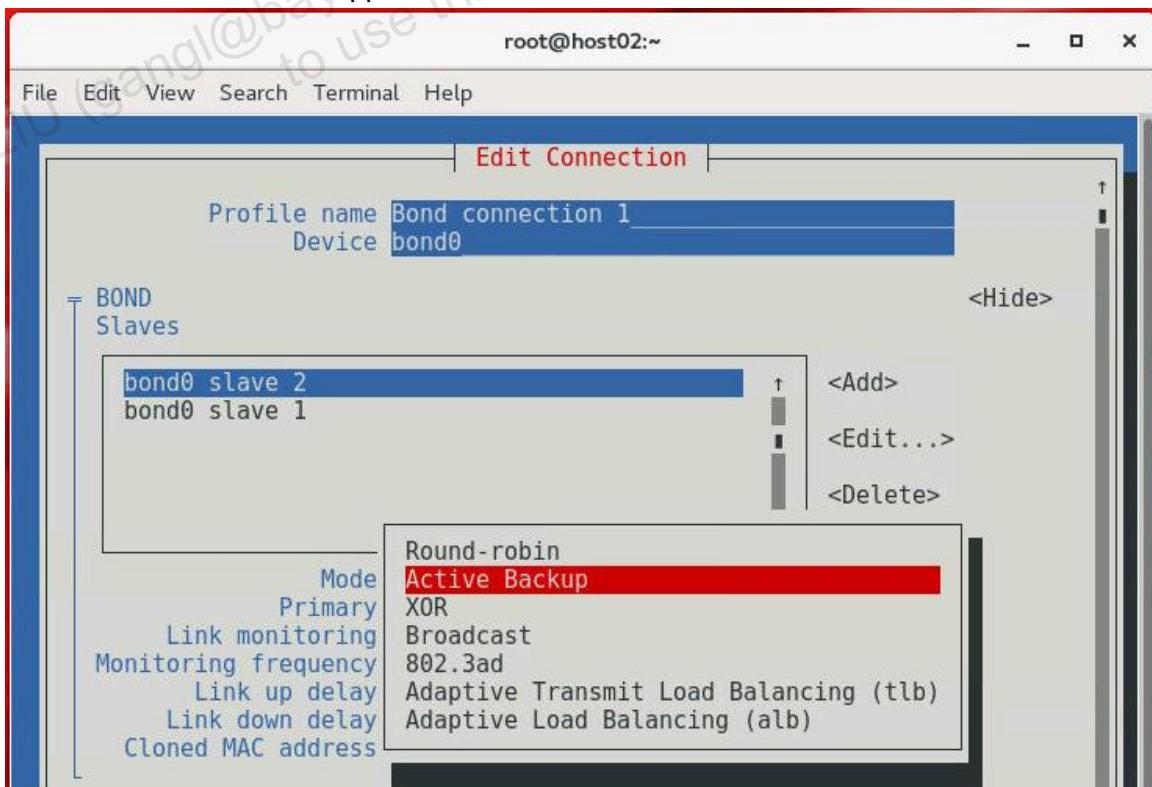
- j. Tab to select "<OK>" and press Enter.
- k. Repeat the steps to add the eth2 slave:
  - Select "Add"
  - Change Profile name to "bond0 slave 2"
  - Enter eth2 for Device.
  - The window appears as shown.



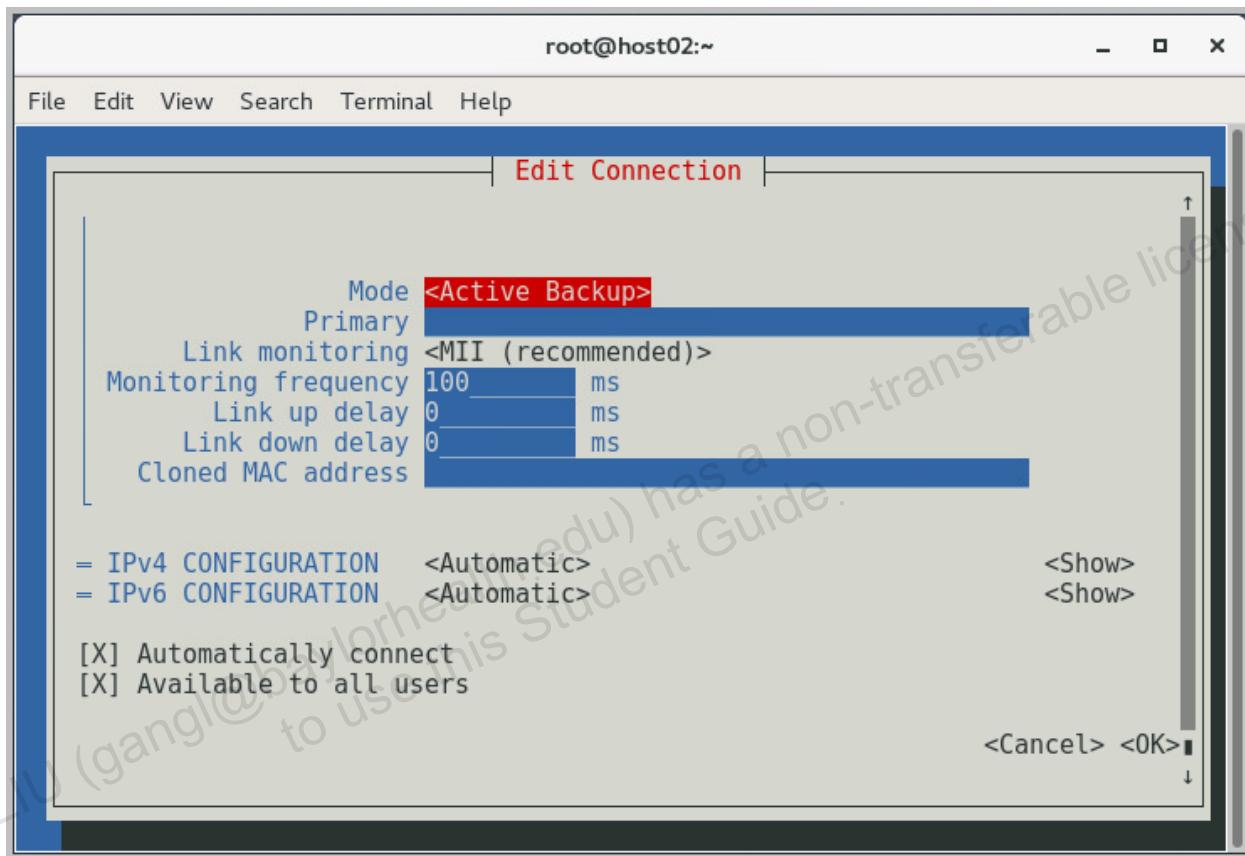
- I. Tab to select "<OK>" and press Enter.
  - The BOND Slaves appear as shown.



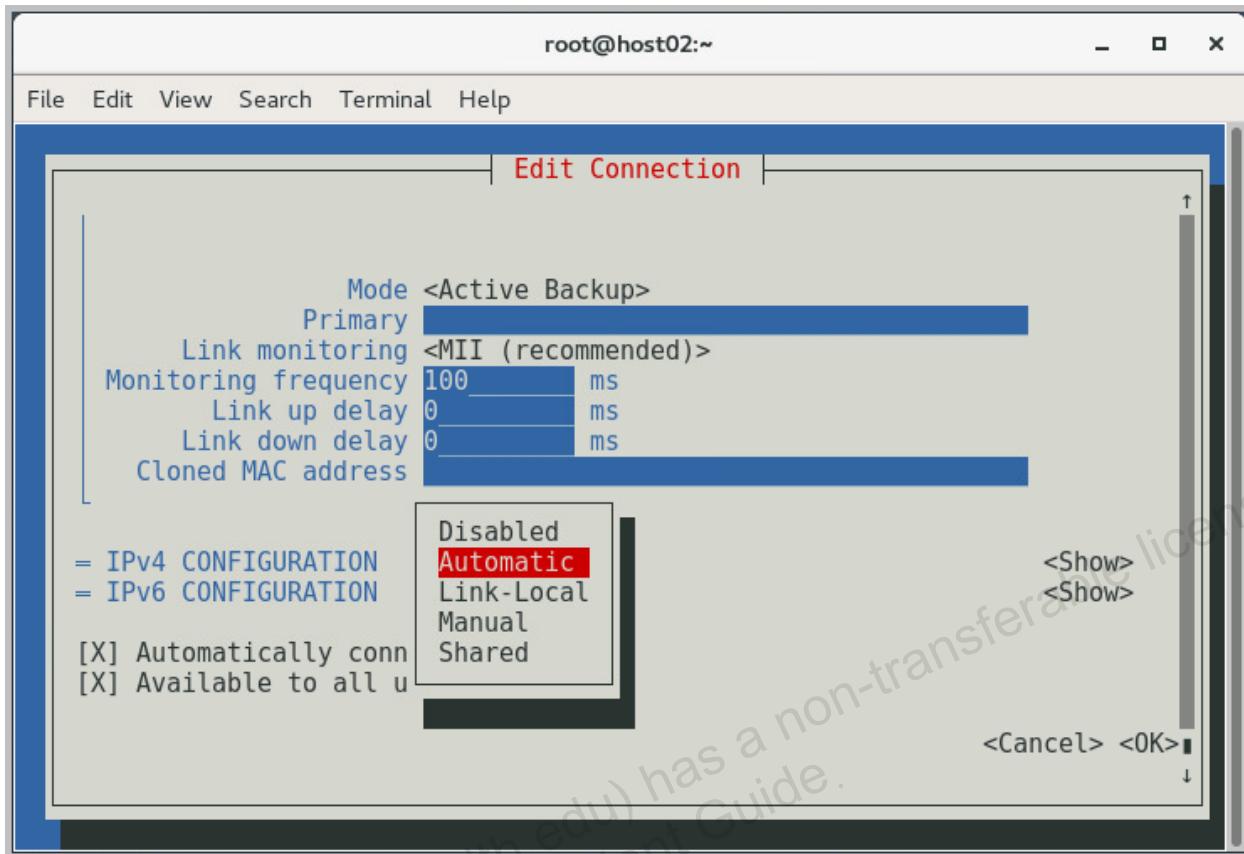
- m. Tab to select “Mode” button and press Enter to display the available modes.
- The list of modes appears.



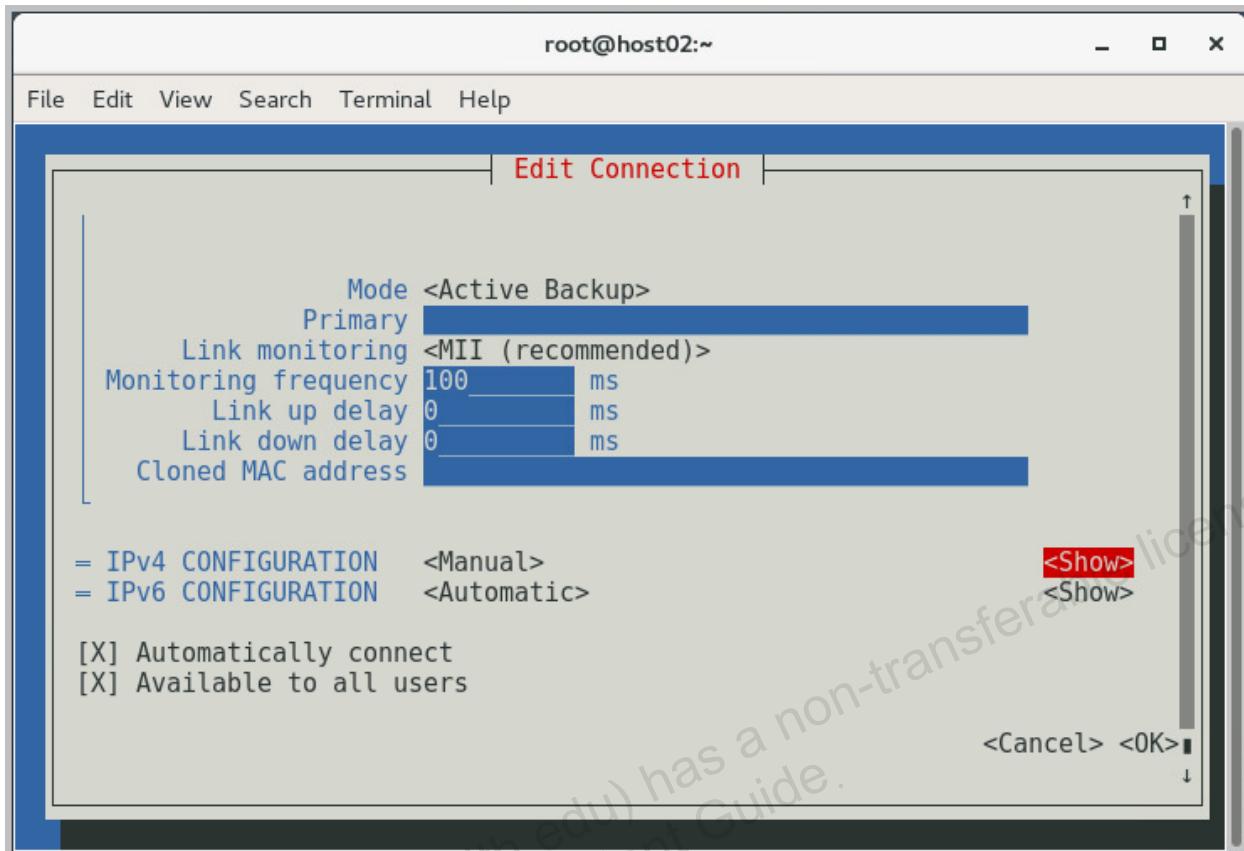
- n. Arrow down to select the “Active backup” option from the drop-down list and press Enter.
- The following window appears.
  - Note that a “Primary” prompt appears when “Active backup” is the selected Mode.
  - You can designate an interface as “Primary” to make it the active slave when it is available.
  - Do not specify a “Primary” interface for this exercise.



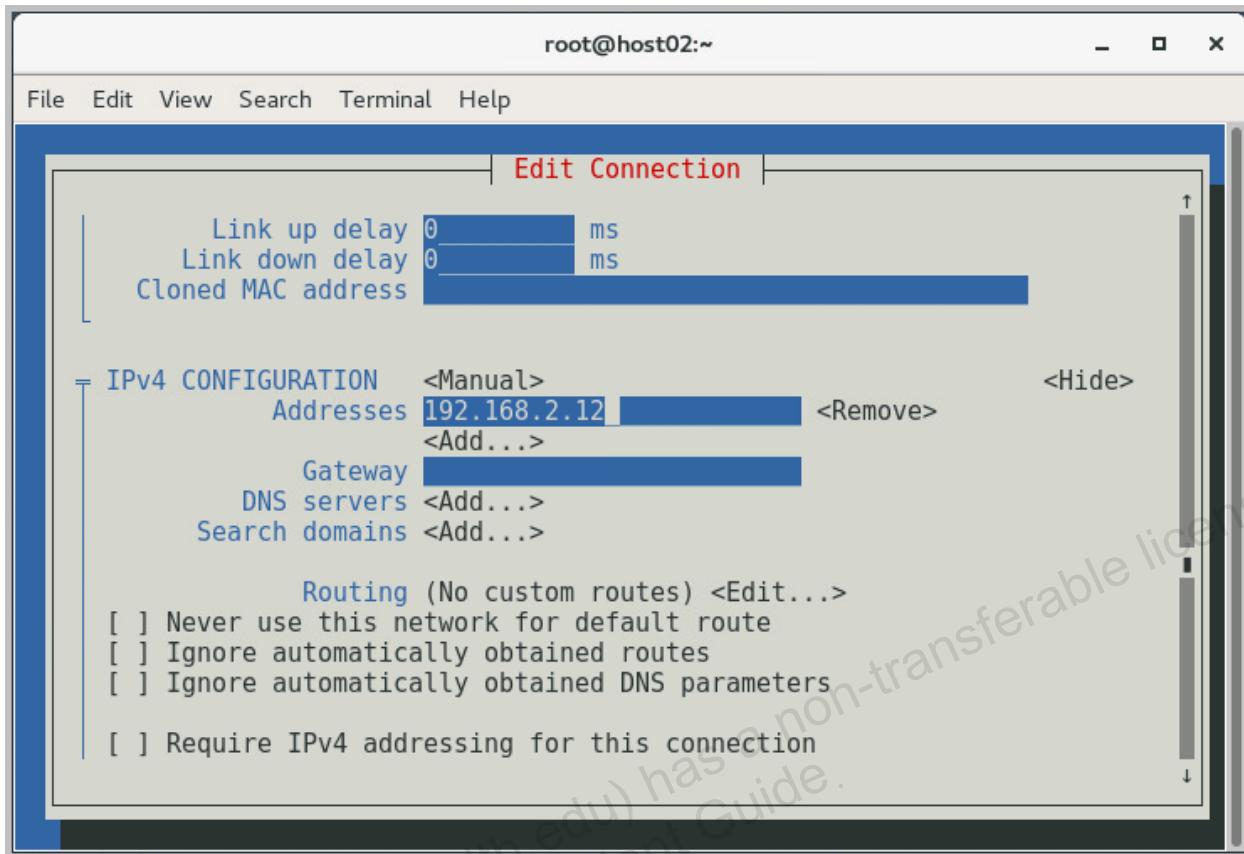
- o. Tab to the “IPv4 CONFIGURATION <Automatic>” and press Enter to display the pop-up menu.



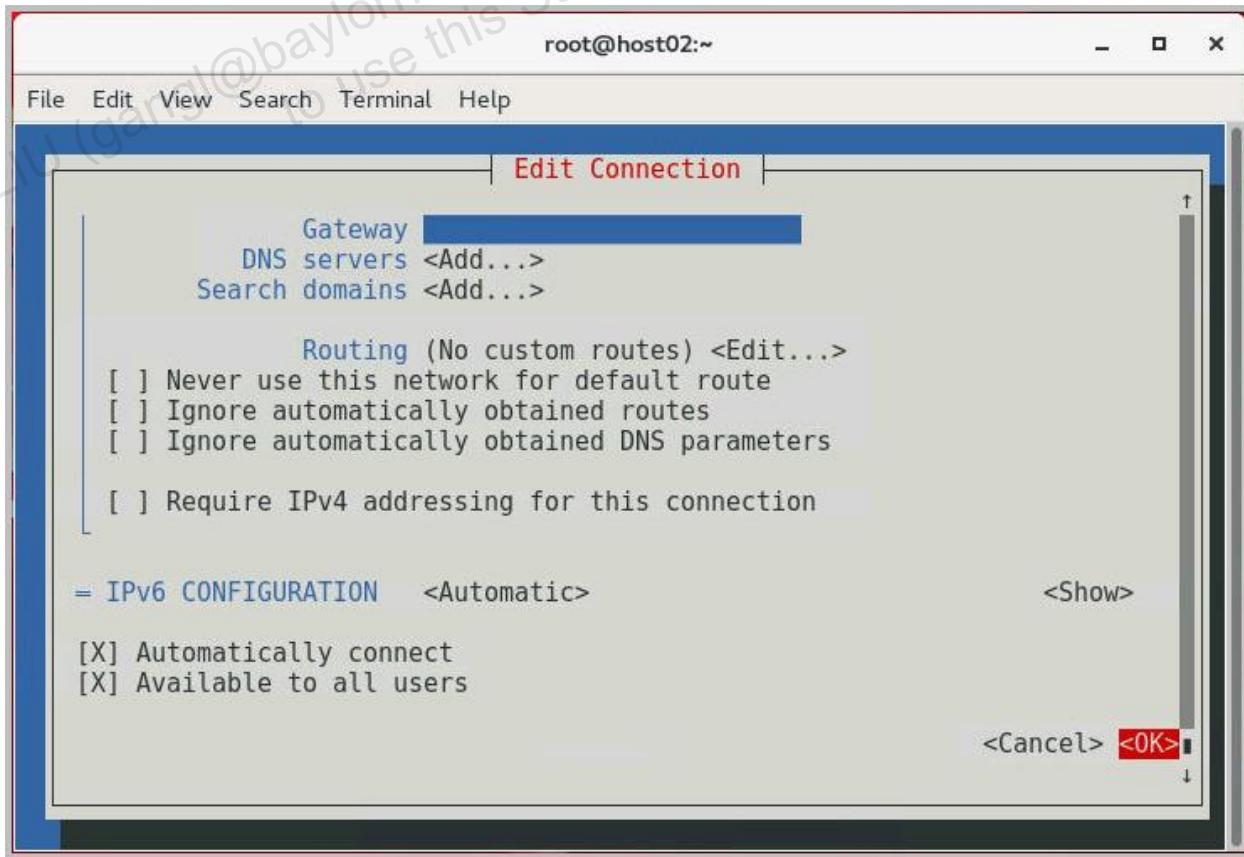
- p. Down arrow select “Manual” and press Enter.
- q. Tab to select “<Show>” to enter IPv4 address information.



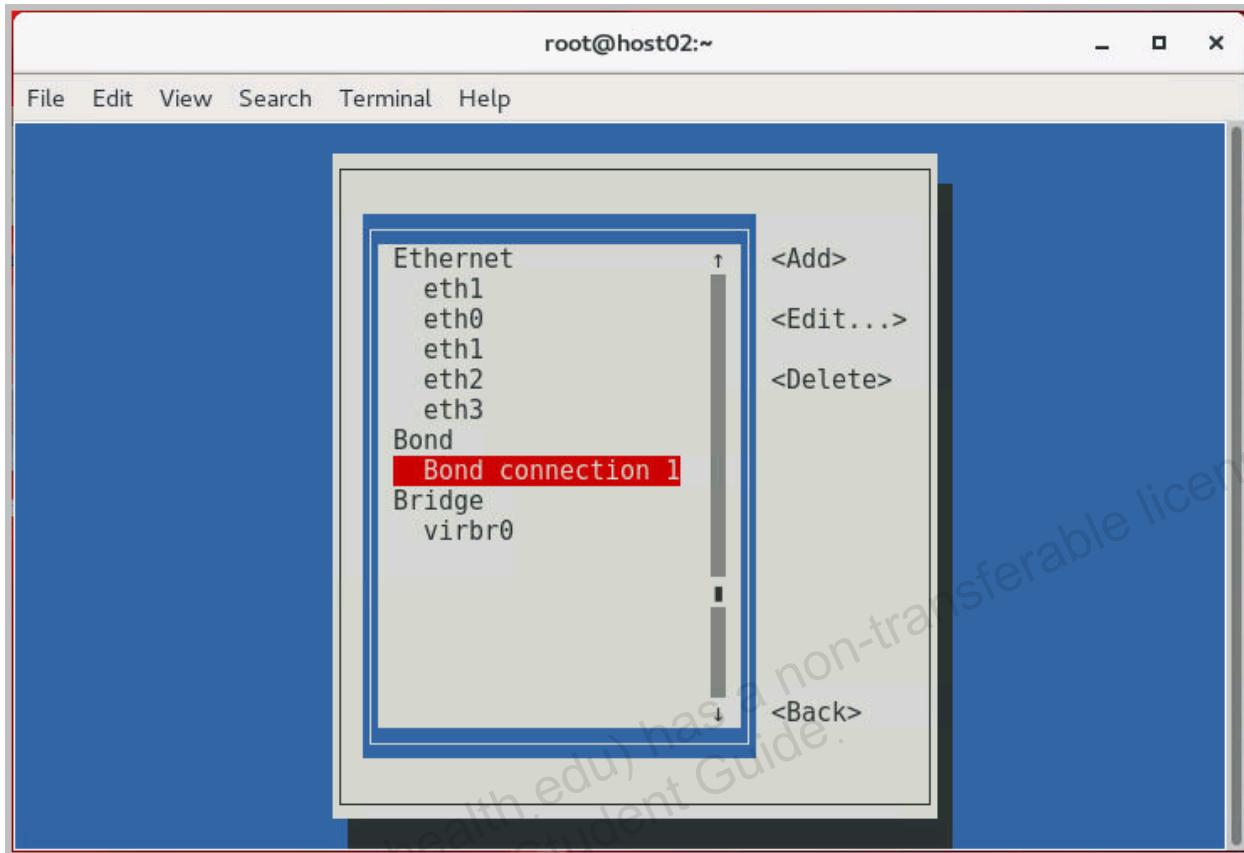
- r. Tab to select “Addresses <Add...>” and press Enter to add the IPv4 address.
- For Address, enter 192.168.2.12 as shown.



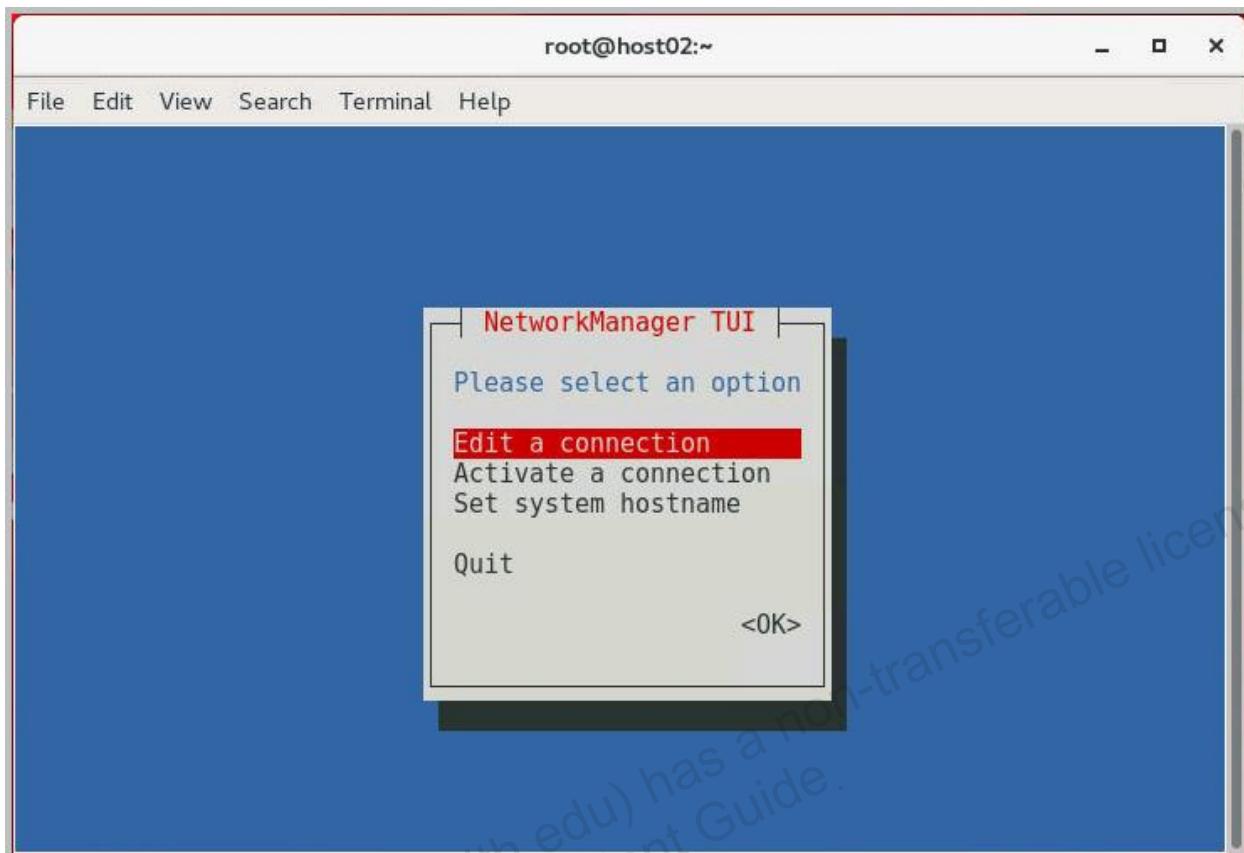
s. Tab to select "<OK>" and press Enter to complete configuring network bonding.



- The following screen appears.



- t. Tab to select "<Back>" and press Enter.
- The following screen appears.



- u. Down arrow select “<Quit>” and press Enter to exit the nmtui interface.
4. 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 UP”.
    - 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 qdisc pfifo_fast master bond0 state UP group default qlen 1000
 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 qdisc pfifo_fast master bond0 state UP group default qlen 1000
 link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
...
9: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500
 qdisc noqueue state UP group default
 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
 noprefixroute 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 ...
ifcfg-eth2 ...
ifcfg-eth3 ...
...
```

- 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`.”

```
cat /etc/sysconfig/network-scripts/ifcfg-Bond_connection_1
BONDING_OPTS="ad_select=stable all_slaves_active=0
arp_all_targets=any arp_validate=none downdelay=0
fail_over_mac=none lp_interval=1 miimon=100 min_links=0
mode=active-backup num_grat_arp=1 num_unsol_na=1
primary_reselect=always resend_igmp=1 updelay=0 use_carrier=1
xmit_hash_policy=layer2"
TYPE=Bond
BONDING_MASTER=yes
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.2.12
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME="Bond connection 1"
UUID=...
DEVICE=bond0
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
TYPE=Ethernet
NAME="bond0 slave 1"
UUID=...
ONBOOT=yes
SLAVE=yes
DEVICE=eth3
MASTER_UUID=...
```

- 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
TYPE=Ethernet
NAME="bond0 slave 2"
UUID=...
DEVICE=eth2
ONBOOT=yes
MASTER=bond0
SLAVE=yes
MASTER_UUID=...
```

- f. Use the `nmcli con` command to view the network connections.
- Note that the bond and slave connections are now shown.

NAME	UUID	TYPE	DEVICE
Bond connection 1	...	bond	bond0
bond0 slave 1	...	ethernet	eth3
bond0 slave 2	...	ethernet	eth2
eth0	...	ethernet	eth0
eth1	...	ethernet	eth1
...			
eth2	...	ethernet	--
eth3	...	ethernet	--

- Do not log off **host02**. You use it again in subsequent practices.

## Practice 9-2: 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 **gateway**.

### Tasks

1. From **gateway**, use the ssh command to connect to **host01**.

- If necessary, open a new terminal window on **gateway**.
  - Use the su – command to become the root user in this new terminal window.

```
[oracle@gateway ~]$ su -
Password:
Last login: ...
[root@gateway ~]#
```

- Use the ssh command to log on to **host01**.

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

2. View the network interfaces on **host01**.

- 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 state UNKNOWN group default
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
 ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 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 noprefixroute eth0
 ...
```

```

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UP group default qlen 1000
 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 dynamic eth1
 ...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
 state UP group default qlen 1000
 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 pfifo_fast
 state UP group default qlen 1000
 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.

NAME	UUID	TYPE	DEVICE
eth0	...	ethernet	eth0
eth1	...	ethernet	eth1
eth2	...	ethernet	--
eth3	...	ethernet	--

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 “`iface 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 iface 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.

NAME	UUID	TYPE	DEVICE
bond-slave-eth3	...	ethernet	eth3
bond-slave-eth2	...	ethernet	eth2
bond0	...	bond	bond0
eth0	...	ethernet	eth0
eth1	...	ethernet	eth1
eth2	...	ethernet	--
eth3	...	ethernet	--

- 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.
- f. 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 ...
...
```

- g. Use the `cat` command to view the contents of the `ifcfg-bond0` file.

```
cat /etc/sysconfig/network-scripts/ifcfg-bond0
BONDING_OPTS=mode=active-backup
TYPE=Bond
BONDING_MASTER=yes
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
```

```

IPADDR=192.168.2.11
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=bond0
UUID=...
DEVICE=bond0
ONBOOT=yes

```

- h. 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

```

- i. 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

```

- j. 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 UP”

```

ip addr
...
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master bond0 state UP group default qlen 1000

```

```
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 qdisc pfifo_fast master bond0 state UP group default qlen 1000
 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 qdisc noqueue state UP group default
 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 noprefixroute bond0
...

```

- Do not log off **host01**. You use it again in subsequent practices.

## Practice 9-3: 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=...
^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=...
^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`.
- 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 dormant lower_eth3 queues
address duplex mtu speed
addr_len flags name_assign_type
statistics
bonding gro_flush_timeout netdev_group subsystem
broadcast ifalias operstate
tx_queue_len
carrier ifindex phys_port_id type
carrier_changes iflink phys_port_name uevent
dev_id link_mode phys_switch_id
dev_port lower_eth2 power
```

- 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
DEVTYPE=bond
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 arp_all_targets 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
tlb_dynamic_lb
ad_partner_mac fail_over_mac num_unsol_na updelay
ad_select lacp_rate packets_per_slave
use_carrier
```

```
all_slaves_active lp_interval 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.

- Bringing one slave up also brings the bond and other slave up.

```
[host01]# nmcli con up bond-slave-eth2
Connection successfully activated (d-Bus active ...)
```

- e. 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:00:04:01
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 kernel: bond0: link status definitely down
for interface eth2, disabling it
<date_time> host01 kernel: bond0: making interface eth3 the new
active one
```

- 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

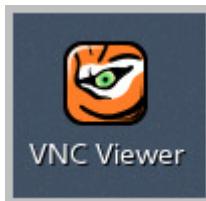
Slave Interface: eth2
MII Status: down
Speed: Unknown
Duplex: Unknown
Link Failure Count: 1
Permanent HW addr: 00:16:3e:00:03:01
Slave queue ID: 0
```

```
Slave Interface: eth3
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 00:16:3e:00:04:01
Slave queue ID: 0
```

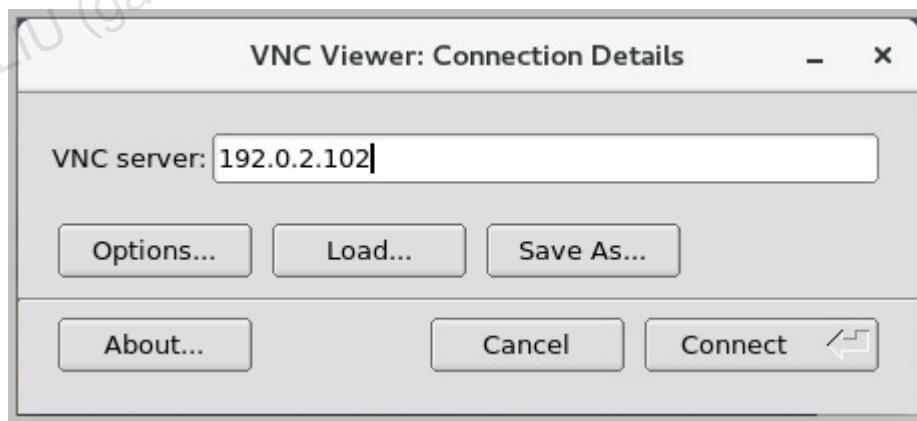
- h. Use the cat command to view the /sys/class/net/bond0/bonding/active\_slave file.
- i. This file also indicates that eth3 is the active slave.

```
[host01]# cat /sys/class/net/bond0/bonding/active_slave
eth3
```

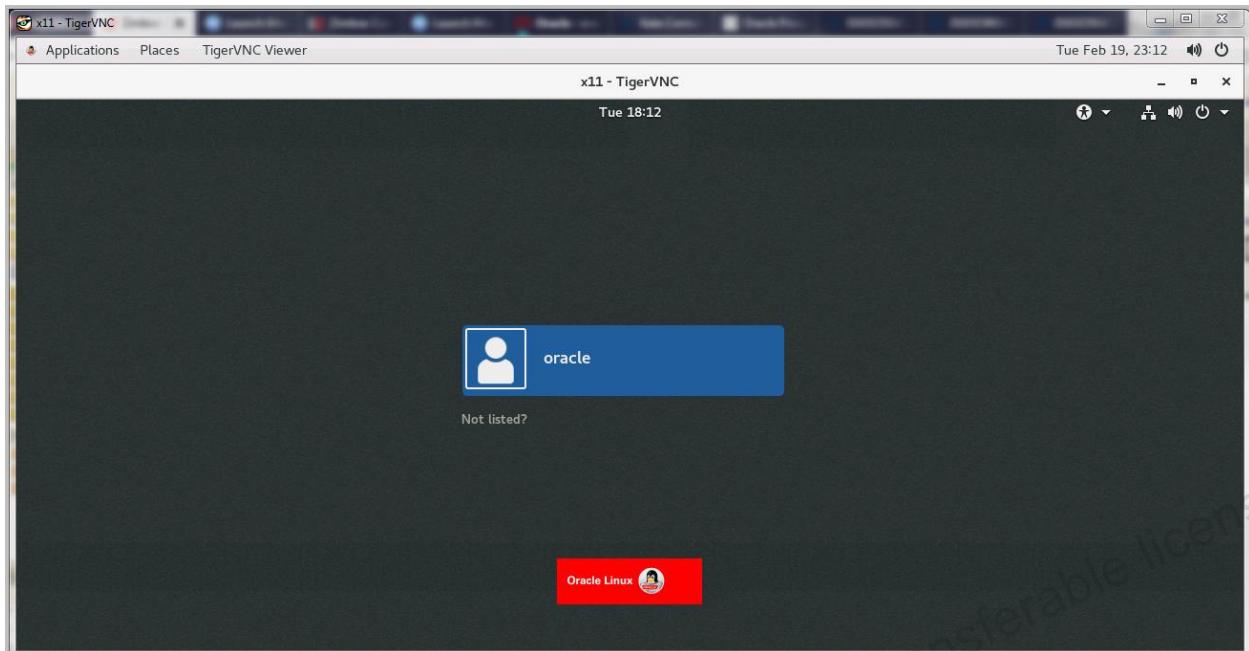
- 5. Remove bond and slave connections on host02.
  - Use the nm-connection-editor GUI interface on host02.
  - Use of this command requires you to connect to host02 by using VNC Viewer.
  - a. Double-click the “VNC Viewer” icon on the gateway desktop.



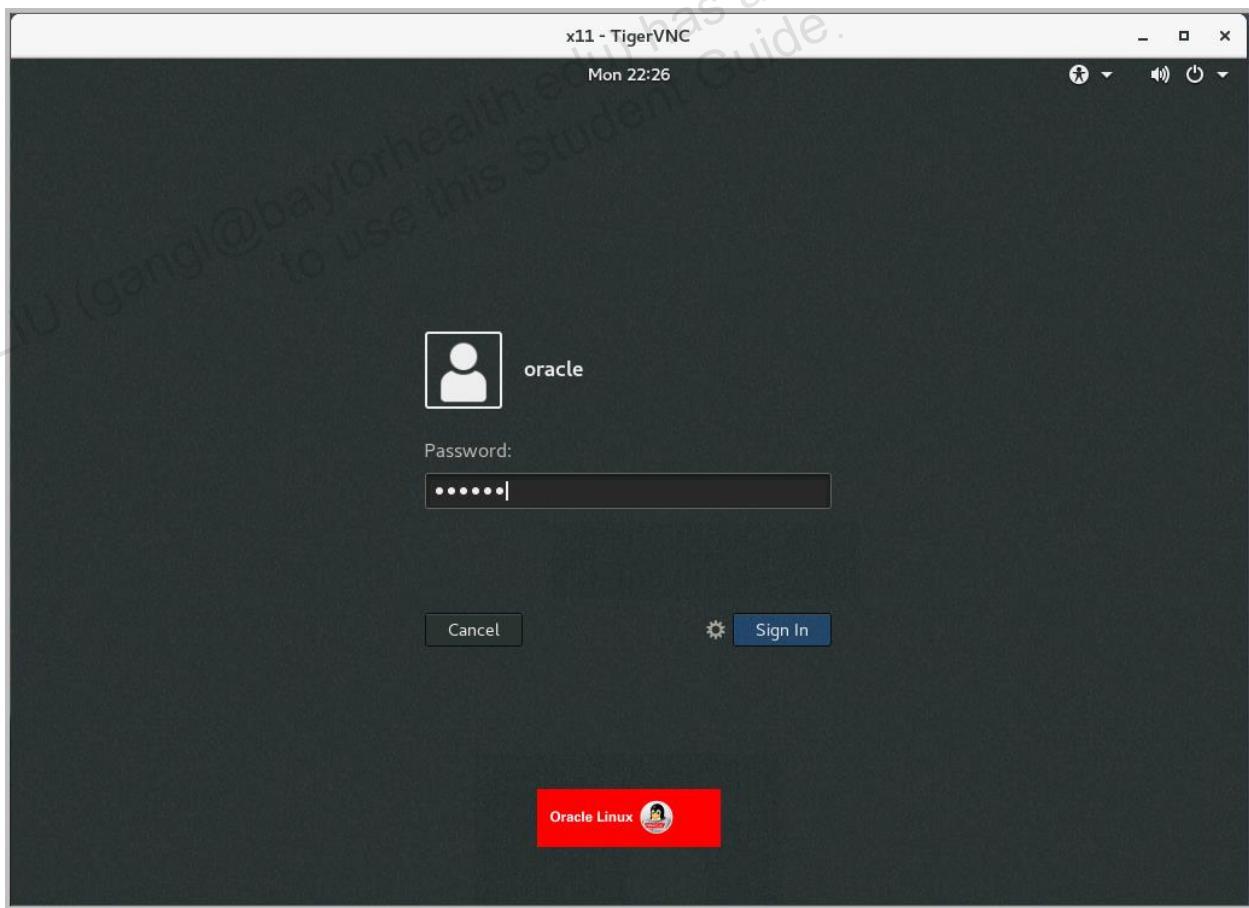
- b. Enter 192.0.2.102 as shown in the following window.
  - Click Connect.



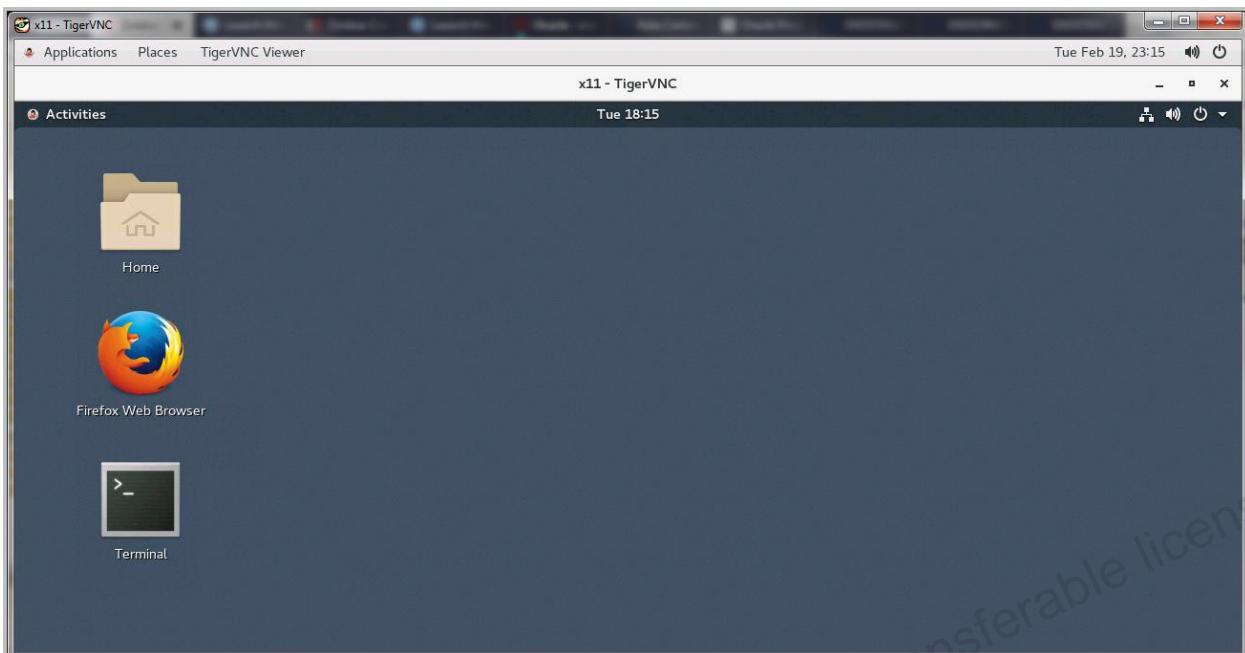
- The login window appears.
- Click in the window to see the oracle user.



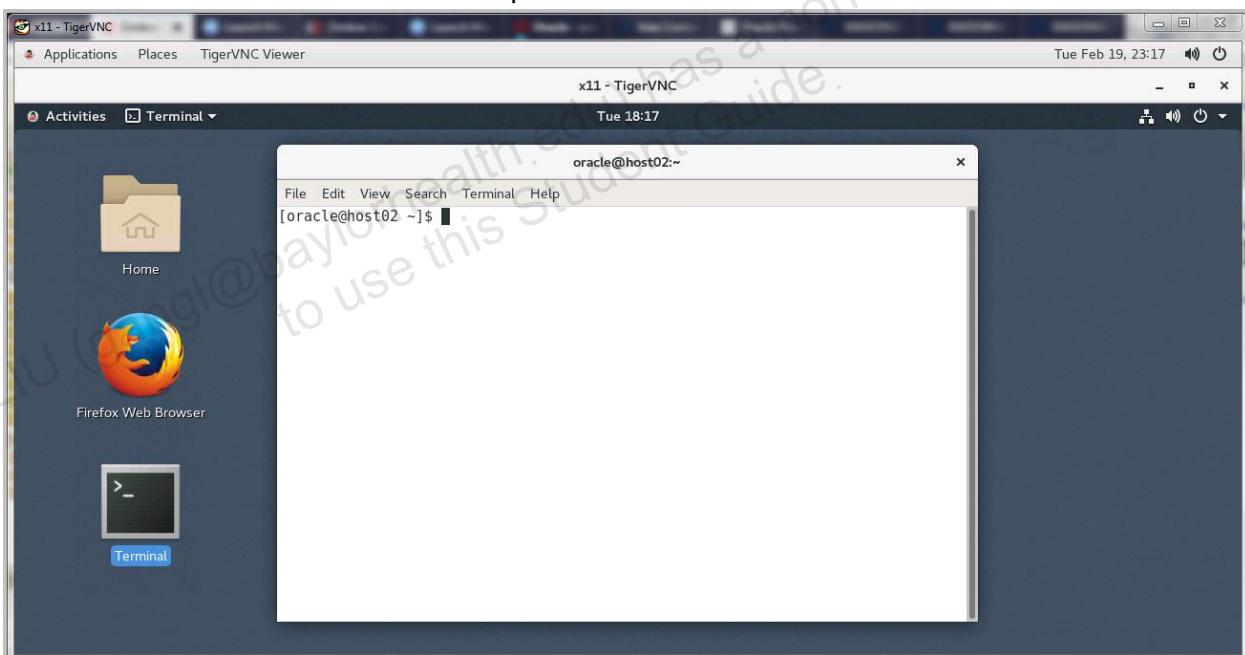
- c. Click the `oracle` user and login.
  - Password is `oracle`.



- The desktop appears as shown.



- d. Double-click the Terminal icon to open a terminal window as shown.



- e. In the terminal window, use the `su -` command to become the root user on host02.

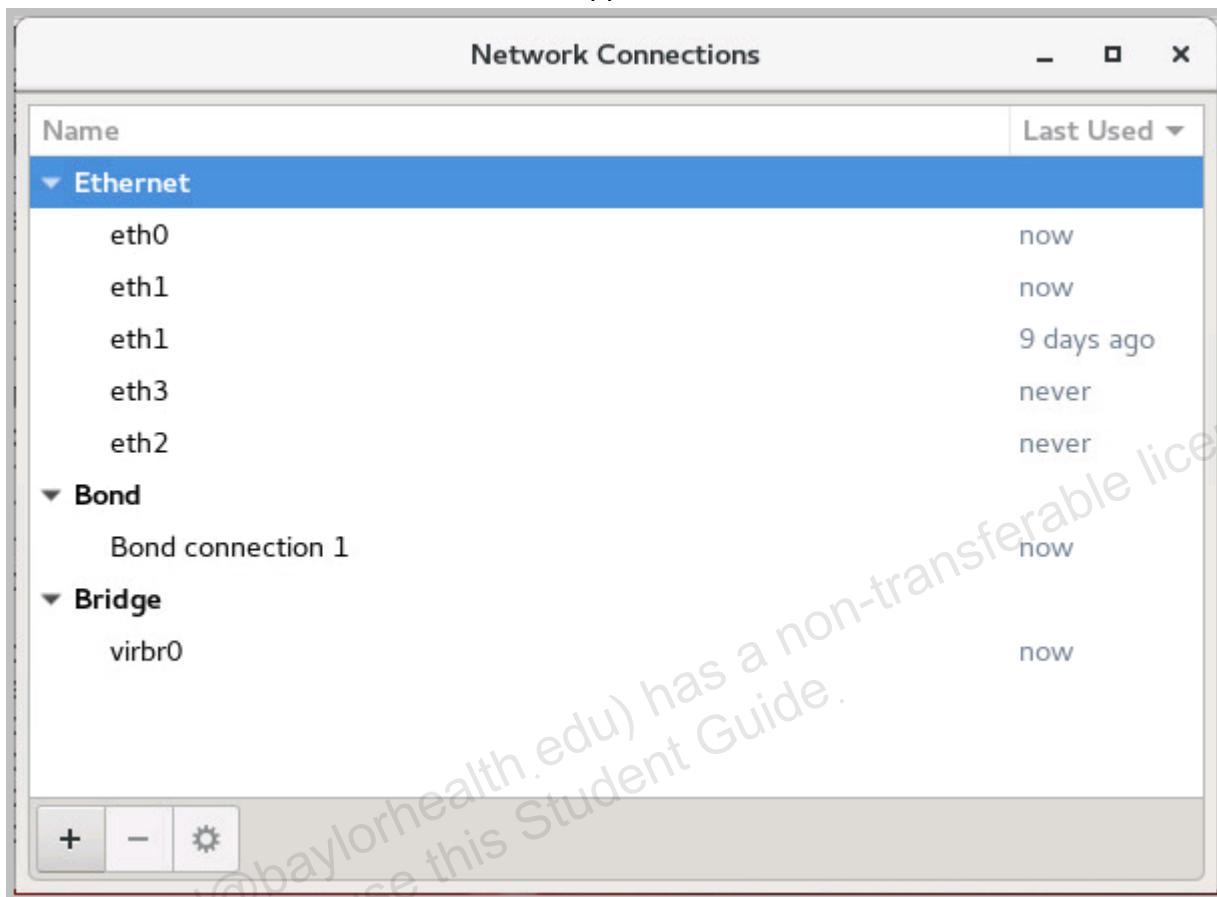
```
oracle@host02 ~] $ su -
Password:
Last login: ...
[root@host02 ~] #
```

- f. On **host02**, use the `nm-connection-editor` command to bring up the “Network Connections” window.

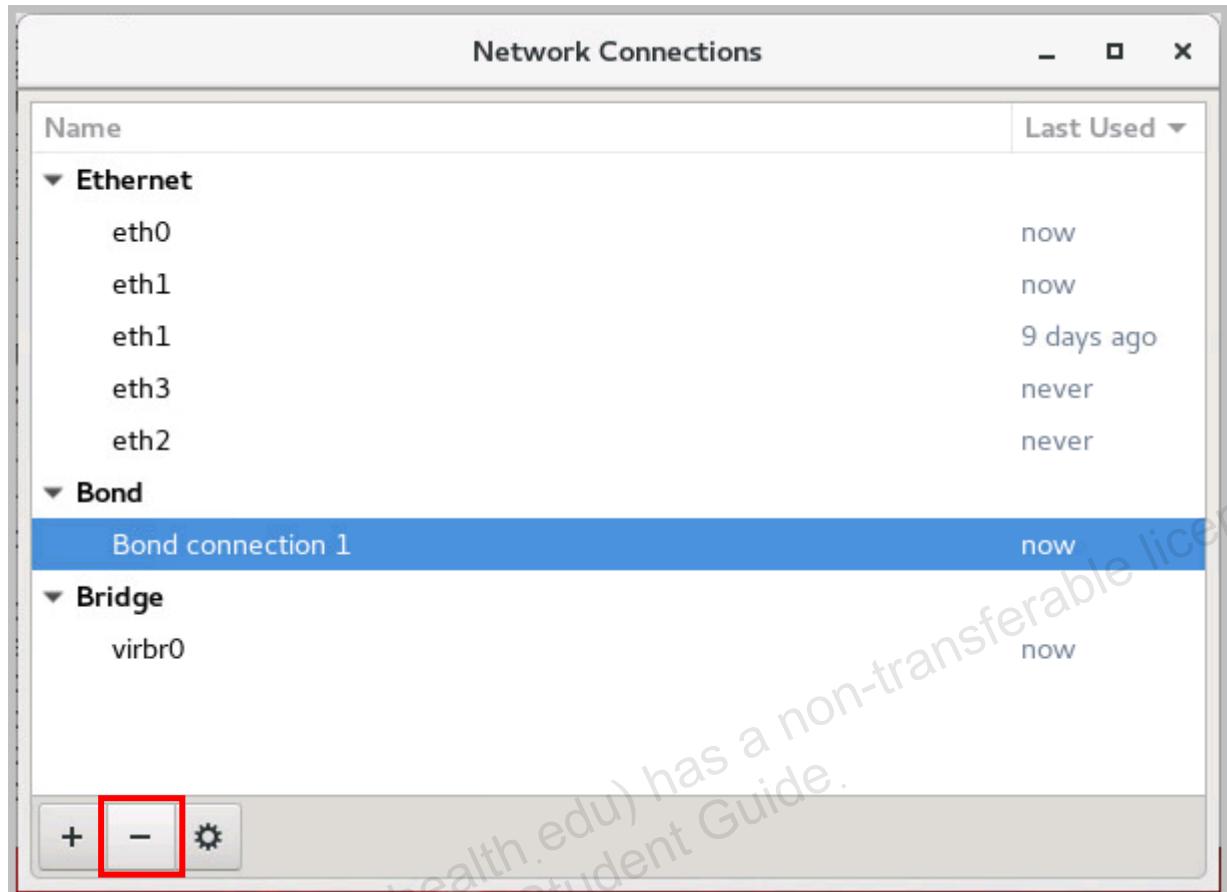
- From the terminal window on the GNOME desktop, run the `nm-connection-editor` command.

```
nm-connection-editor
```

- The “Network Connections” window appears.



- g. Click the “Bond connection 1” entry then click the delete button (minus sign) as shown:



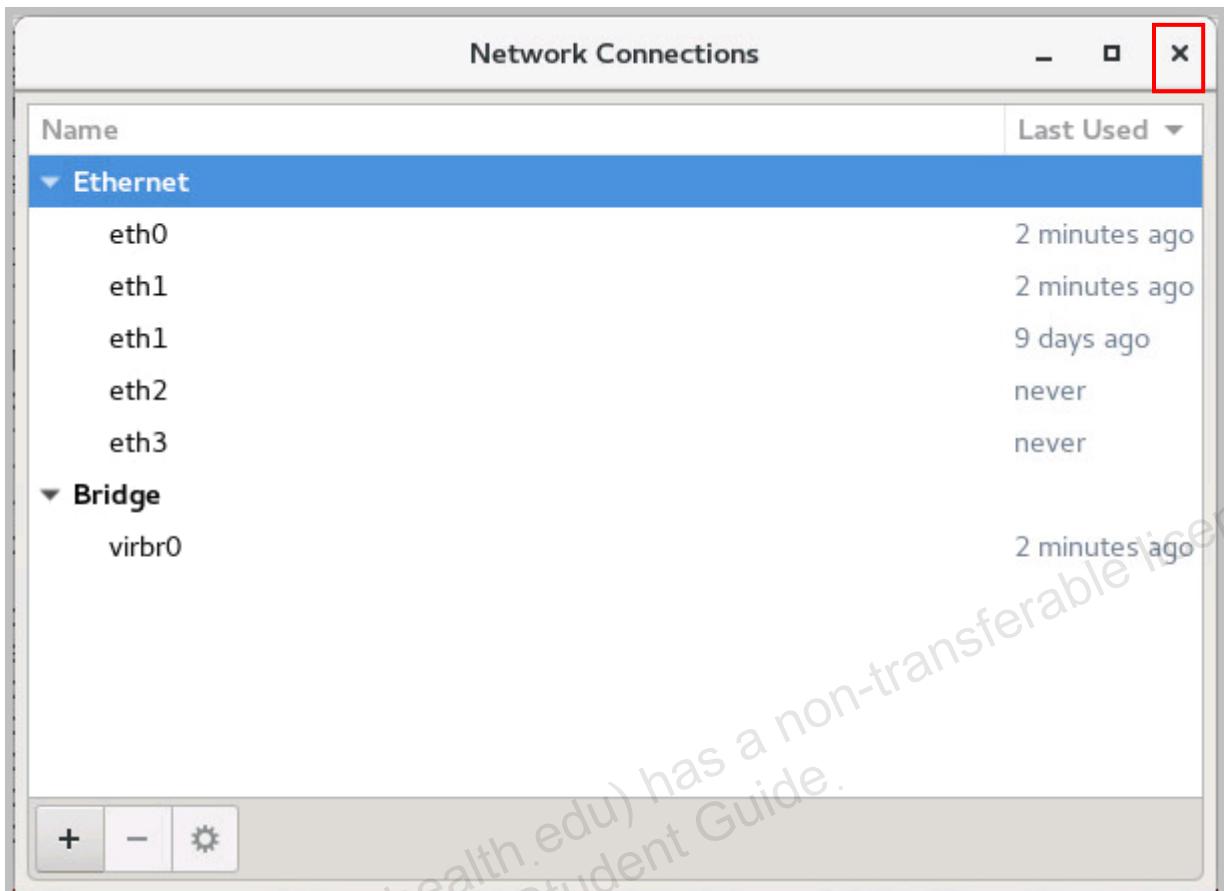
h. The following confirmation box appears.

- Click the Delete button.



- Note that the Bond connection is deleted.

i. Click the “x” in the top-right corner to close the “Network Connections” window.



Do not close the VNC connection to **host02**. You use this in a subsequent practice.

- j. 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
```

- k. 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
```

- l. 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
```

- m. 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
eth0 ... ethernet eth0
eth1 ... ethernet eth1
...
```

eth2	...	ethernet	--
eth3	...	Ethernet	--

- n. 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 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
 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 pfifo_fast state UP mode DEFAULT group default qlen 1000
 link/ether 00:16:3e:00:04:02 brd ff:ff:ff:ff:ff:ff
...
```

- o. 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 were removed.

```
[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
bond-slave-eth3 ... ethernet eth3
bond-slave-eth2 ... ethernet eth2
bond0 ... bond bond0
eth0 ... ethernet eth0
eth1 ... ethernet eth1
eth2 ... ethernet --
eth3 ... ethernet --
```

- 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
eth0 ... ethernet eth0
eth1 ... ethernet eth1
eth2 ... ethernet --
eth3 ... ethernet --
```

- 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 ...
...
```

- 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 9-4: 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 Connections” GUI to configure VLAN tagging on **host02**
- Review network configuration on **host02**

### Assumptions

- You are the `root` user on **host02**.
- You are connected to **host02** from **gateway** by using VNC Viewer.

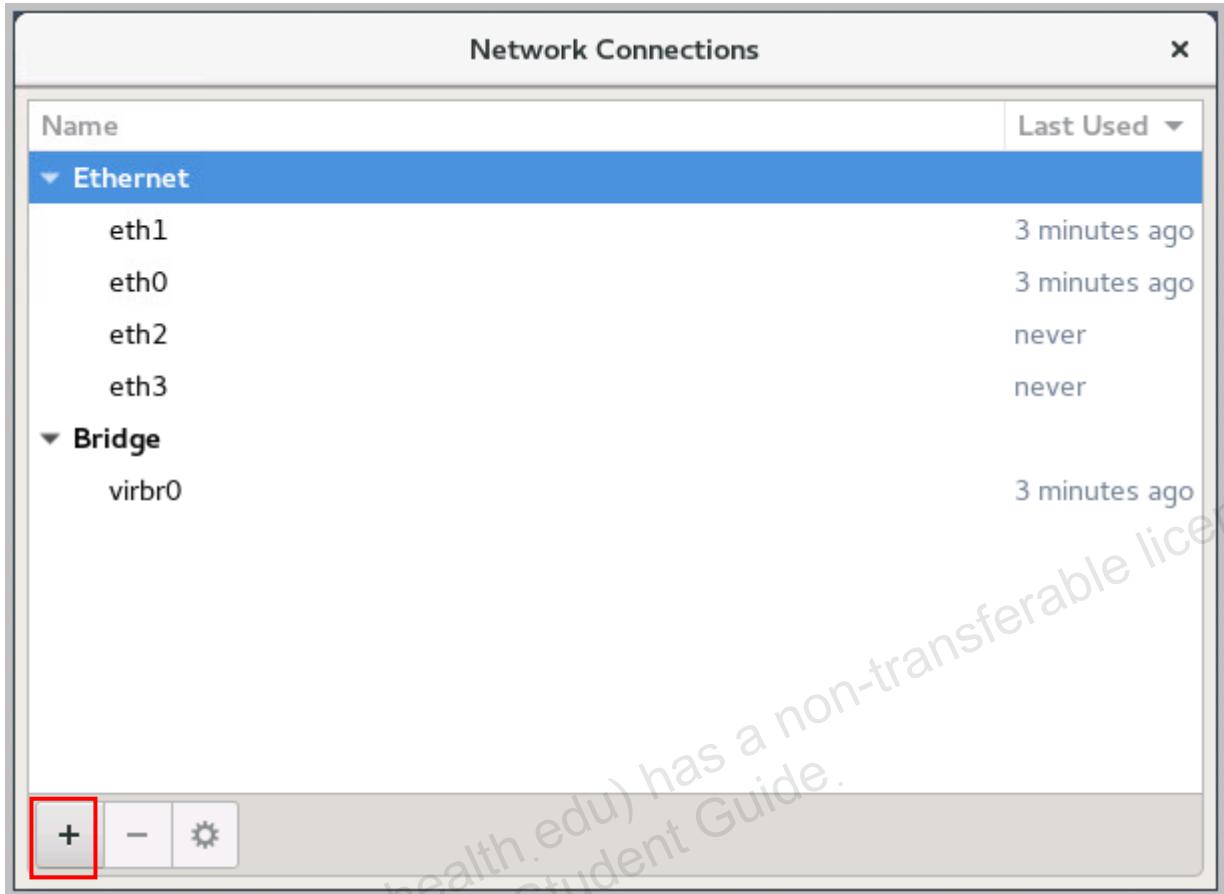
### Tasks

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 32768 0
...
```
2. Use the `nm-connection-editor` command to configure VLAN tagging.
  - a. Open a terminal window from **host02** GNOME desktop and run the command.

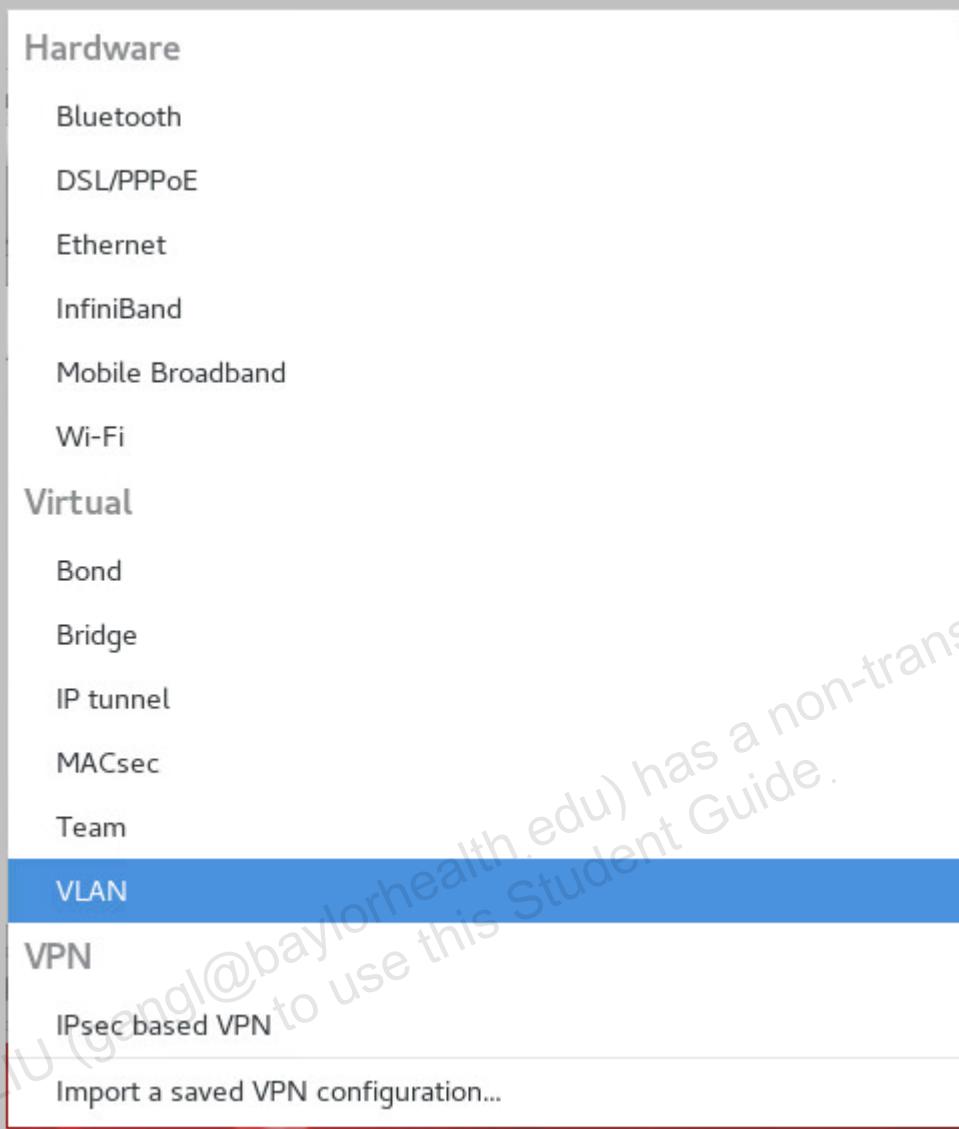
```
nm-connection-editor
```
  - b. The “Network Connections” window appears.



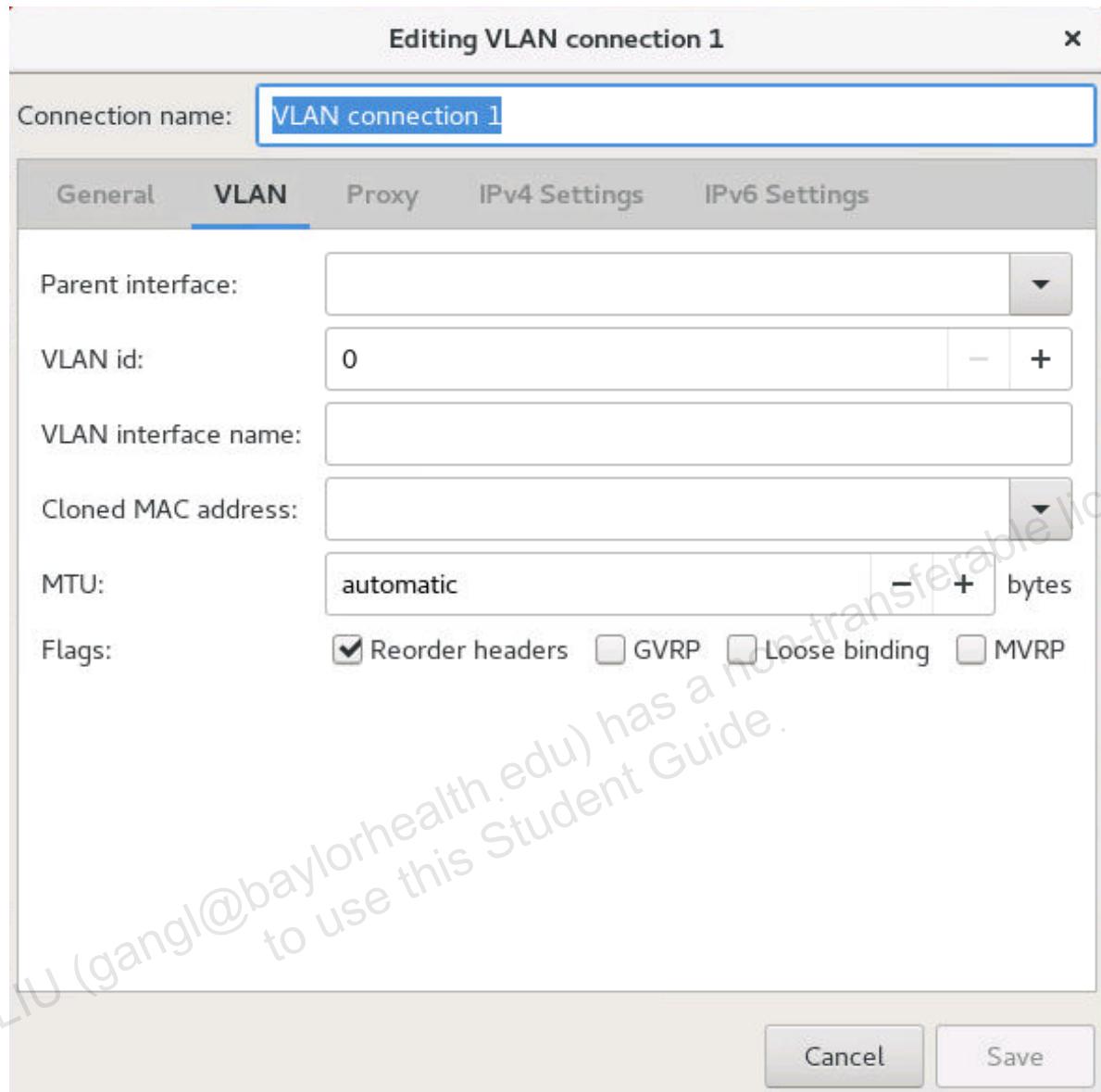
- c. Click the “+” button to add a new connection type.  
• The following window appears:



- d. Select “VLAN” from the drop-down menu.

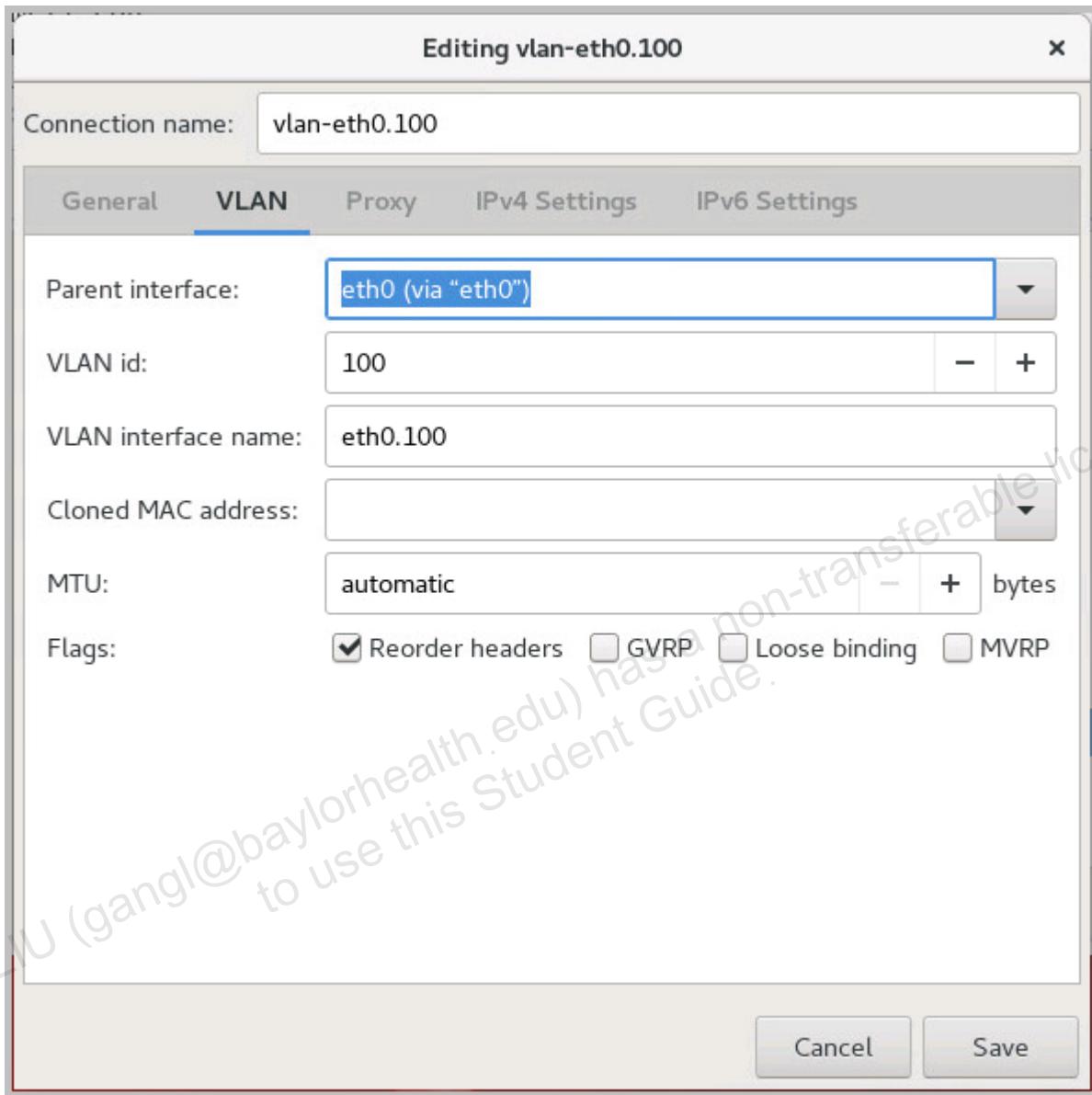


- e. With VLAN selected, click the “Create...” button to add a VLAN connection.
  - The following window appears.
  - The default Connection name is “VLAN connection 1.”



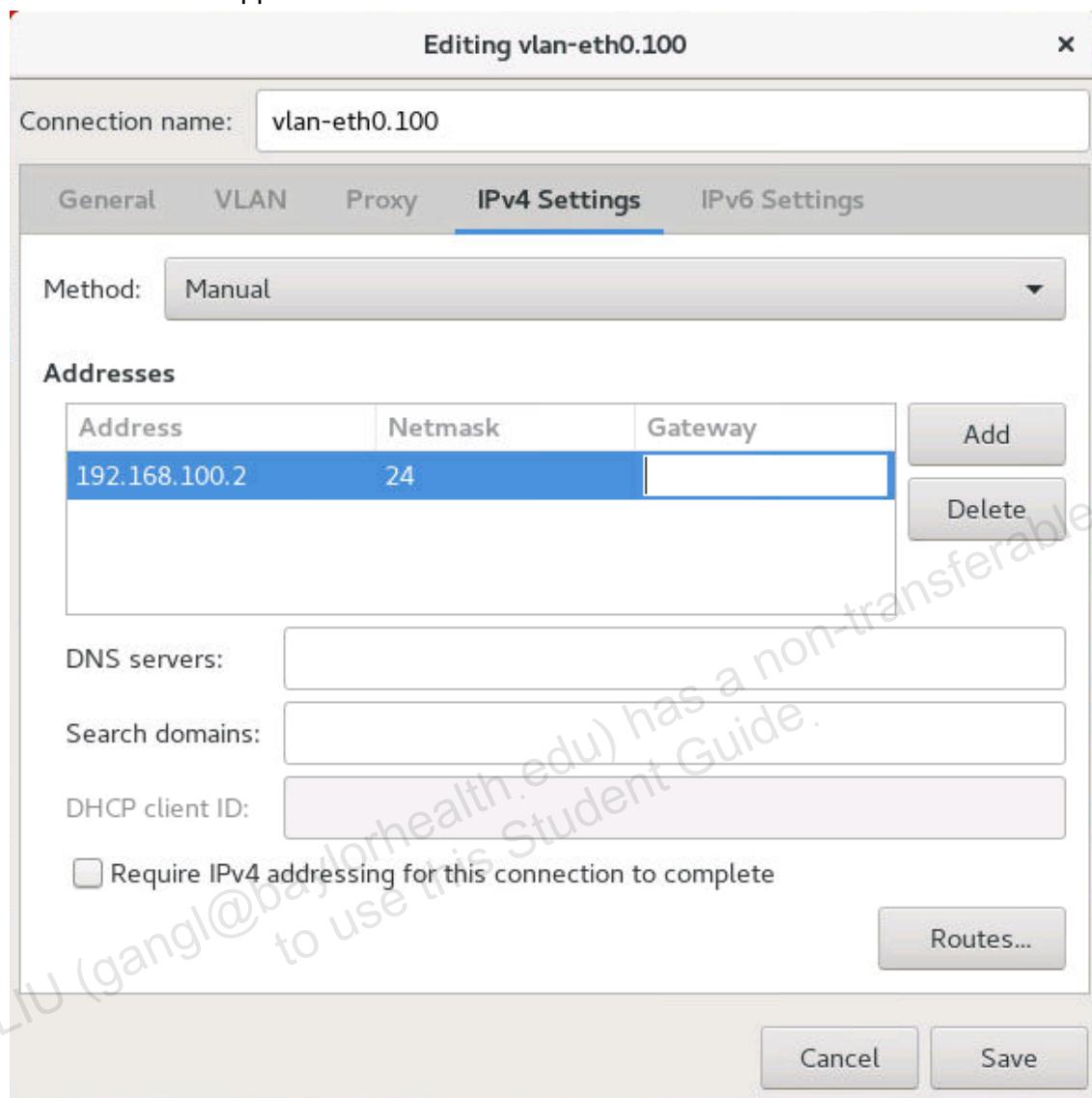
- f. Update the screen as follows.
- Change “Connection name:” to `vlan-eth0.100`.
  - Click the “Parent interface:” down arrow and select `eth0` (via “`eth0`”).
  - Change “VLAN id:” to `100`.
  - Change “VLAN interface name:” to `eth0.100`.

- The window appears as follows.



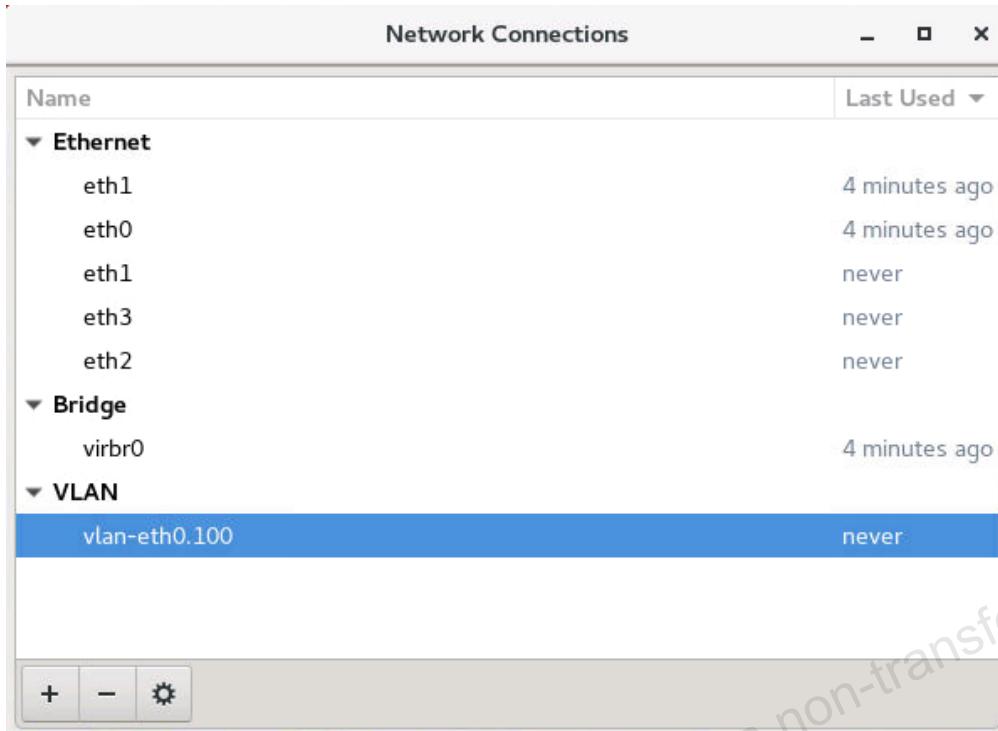
- Click the "IPv4 Settings" tab to assign an IPv4 address to the VLAN interface.
- Change the "Method" to "Manual."
- Click "Add" to add the following Address information:
  - Address: 192.168.100.2
  - Netmask: 24
  - Gateway: <empty>

- h. The window appears as follows.



- i. Click "Save" to complete configuring VLAN tagging.

- The "VLAN (eth0.100)" interface now appears in the "Network Connections" window.



- j. 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: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 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 noprefixroute eth0
...
10: eth0.100@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
 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 global noprefixroute 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 wlan-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 ... wlan eth0.100
eth0 ... ethernet eth0
eth1 ... 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`.

```
cat /etc/sysconfig/network-scripts/ifcfg-vlan-eth0.100
VLAN=yes
TYPE=Vlan
PHYSDEV=...
VLAN_ID=100
REORDER_HDR=yes
GVRP=no
MVRP=no
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.100.2
PREFIX=24
DEFROUTE=yes
```

```
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_PRIVACY=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=vlan-eth0.100
UUID=...
DEVICE=eth0.100
ONBOOT=yes
```

## Practice 9-5: 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 32768 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 wlan-eth0.100`” argument to specify the name of the new VLAN connection.
    - Use the “`iface 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 wlan-eth0.100 iface eth0.100
dev eth0 id 100 ip4 192.168.100.1/24
Connection 'wlan-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: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 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 noprefixroute eth0
8: eth0.100@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
 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 global noprefixroute 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
eth1 ethernet connected eth1
eth0.100 vlan connected wlan-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 ... wlan eth0.100
eth0 ... 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`.

```
cat /etc/sysconfig/network-scripts/ifcfg-vlan-eth0.100
VLAN=yes
TYPE=Vlan
PHYSDEV=eth0
VLAN_ID=100
REORDER_HDR=yes
GVRP=no
MVRP=no
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.100.1
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=vlan-eth0.100
UUID=...
DEVICE=eth0.100
ONBOOT=yes
```

## Practice 9-6: 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=...
^C
...
```

1. 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
...
```

1. From **host01**, use the `ping` command to communicate to the VLAN interface on **host02**.

- d. 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=...
^C
...
```

- e. 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.

```
[host02]$ su -
Password:
[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.
  - Sample output is shown.

```
... 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.
- h. 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 ...
```

- i. 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
...
```

- j. 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).
  - Sample output is shown:

```
... 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
...
```

- k. 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
```

- l. 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 dormant lower_eth0 power
address duplex mtu queues
addr_len flags name_assign_type speed
broadcast gro_flush_timeout netdev_group
statistics
carrier ifalias operstate subsystem
carrier_changes ifindex phys_port_id
tx_queue_len
dev_id iflink phys_port_name type
dev_port link_mode phys_switch_id uevent
```

- 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 37
 total bytes received 2772
 Broadcast/Multicast Rcvd 0

 total frames transmitted 47
 total bytes transmitted 3998
Device: eth0
INGRESS priority mappings: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0
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 ... 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 ... 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
```

- i. Log off **host01**.

```
[host01]# exit
logout
Connection to host01 closed.
[root@gateway ~]#
```

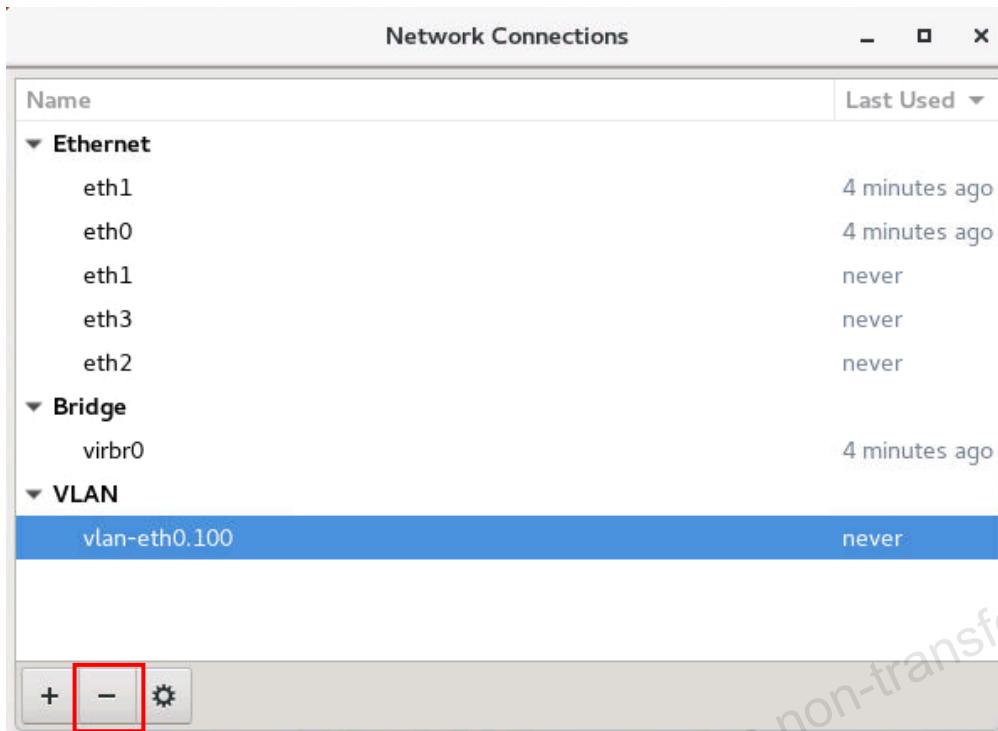
6. Remove VLAN interface on **host02**.

- a. On **host02**, use the `nm-connection-editor` command to bring up the “Network Connections” window.

- From the terminal window on the **host02** desktop, run the `nm-connection-editor` command.

```
nm-connection-editor
```

- The “Network Connections” window appears.



- Click the “vlan-eth0.100” entry.
- Click the “–” button below to remove the VLAN interface.
  - The dialog window appears as follows.
- Click the “Delete” button.
- Click the “x” in the top right corner to close the “Network Connections” window.
- 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 ... ethernet eth0
...
```

- 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
```

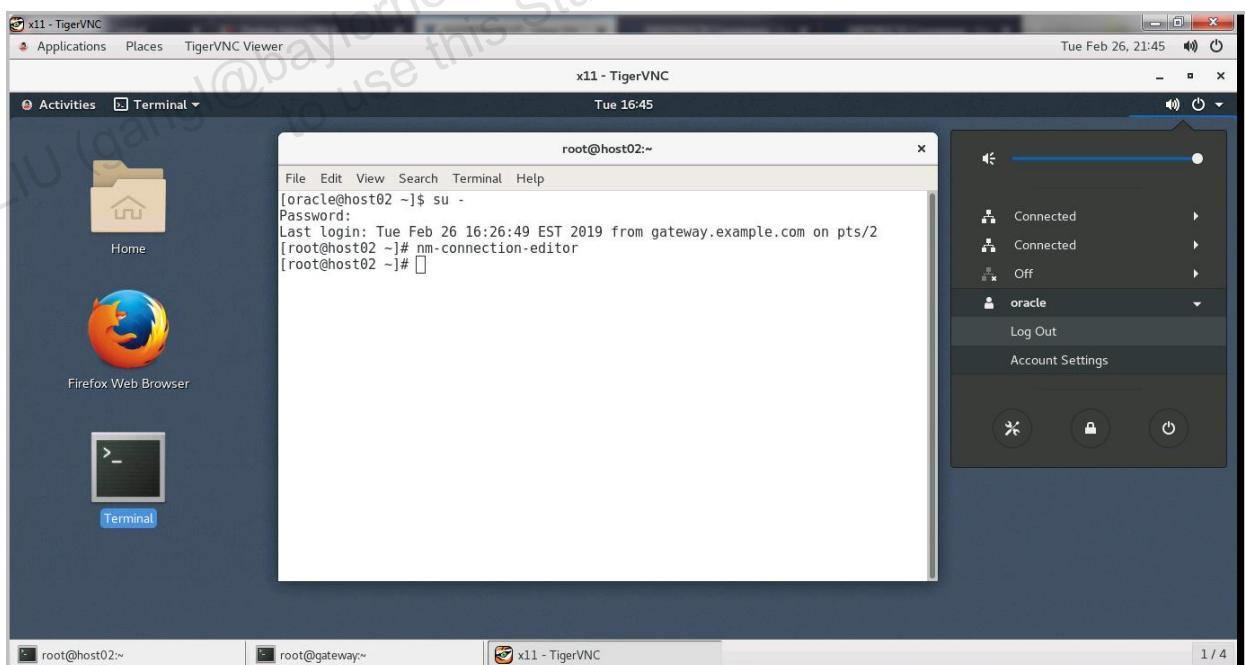
k. Use the `cat` command to view the `config` file.

- 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
```

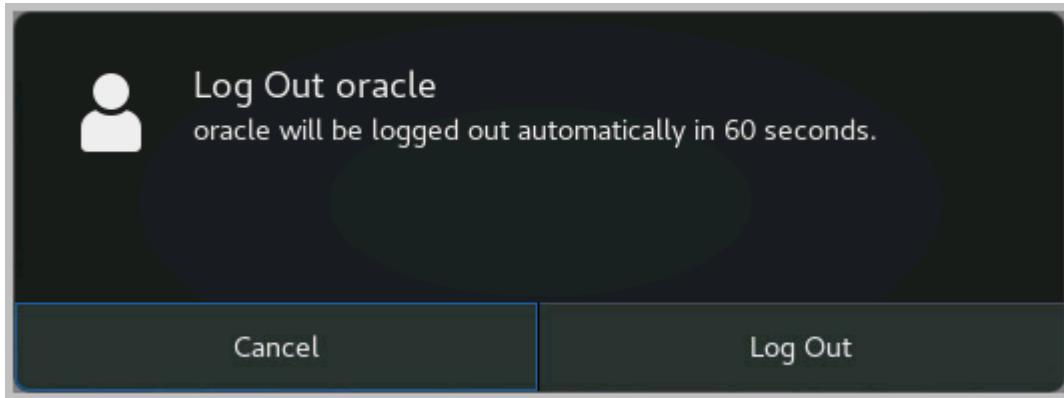
7. Log out and close the VNC connection to **host02**.

a. Click the down arrow in the top-right corner of the window to display the drop-down menu as shown.

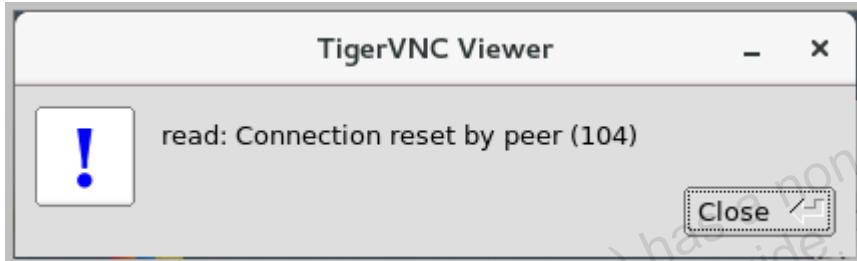


b. Click the `oracle` user to view the `Log Out` option as shown.

c. Click `Log Out` as shown.



- The **host02** desktop closes.
- The following window might appear. Close the window by clicking the X in the upper-right corner.



- If you are connected to **host02** from **gateway** by using the `ssh` command, log out from this window also.

```
[host02]# exit
logout
Connection to host02 closed.
[root@gateway ~]#
```

## Practice 9-7: Configuring a Site-to-Site VPN

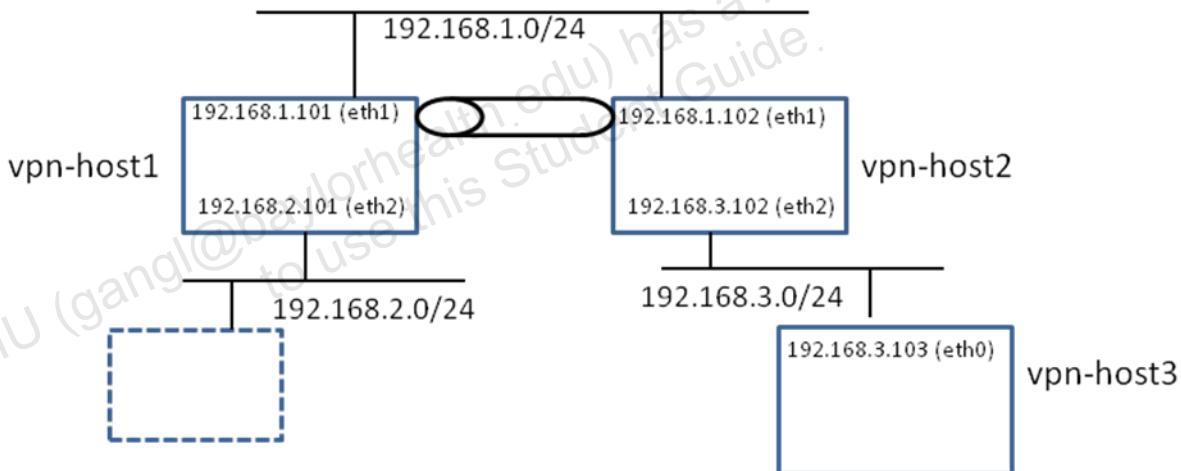
### Overview

In this practice, you:

- Log in to **vpn-host1** and **vpn-host2** and view the network configurations
- 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 **gateway**.
- The following describes the **vpn\*** virtual machines network configuration.



- You create a VPN tunnel from **vpn-host1** to **vpn-host2**.
- There is no **vpn-host3** in your environment.
- The **vpn-host3** in the diagram describes a possible host on the 192.168.3 subnet.

### Tasks

1. From a terminal window on **gateway**, use the `ssh` command to log in to **vpn-host1** as the `root` user:

```
[root@gateway ~]# ssh root@vpn-host1
root@vpn-host1's password:
Last login: ...
[root@vpn-host1 ~]#
```

2. Use the `ip addr` command to view the network configuration of **vpn-host1**.

- Note the `eth0` addresses:
  - MAC address: 00:16:3e:01:01:01
  - IP address: 192.0.2.107
- Note the `eth1` addresses:
  - MAC address: 00:16:3e:01:02:01
  - IP address: 192.168.1.101
- Note the `eth2` addresses:
  - MAC address: 00:16:3e:01:03:01
  - IP address: 192.168.2.101

```
[root@vpn-host1 ~]# ip addr
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 link/ether 00:16:3e:01:01:01 brd ff:ff:ff:ff:ff:ff
 inet 192.0.2.107/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 link/ether 00:16:3e:01:02:01 brd ff:ff:ff:ff:ff:ff
 inet 192.168.1.101/24 brd 192.168.1.255 scope global eth1
...
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 link/ether 00:16:3e:01:03:01 brd ff:ff:ff:ff:ff:ff
 inet 192.168.2.101/24 brd 192.168.2.255 scope global eth2
...
```

3. From a 2<sup>nd</sup> terminal window on **gateway**, use the `ssh` command to log in to **vpn-host2** as the **root** user:

```
[root@gateway ~]# ssh root@vpn-host2
root@vpn-host2's password:
Last login: ...
[root@vpn-host2 ~]#
```

4. Use the `ip addr` command to view the network configuration of **vpn-host2**.

- Note the `eth0` addresses:
  - MAC address: 00:16:3e:01:01:02
  - IP address: 192.0.2.108
- Note the `eth1` addresses:
  - MAC address: 00:16:3e:01:02:02
  - IP address: 192.168.1.102

- Note the `eth2` addresses:

- MAC address: `00:16:3e:01:03:02`
- IP address: `192.168.3.102`

```
[root@vpn-host2 ~]# ip addr
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 link/ether 00:16:3e:01:01:02 brd ff:ff:ff:ff:ff:ff
 inet 192.0.2.108/24 brd 192.0.2.255 scope global eth0
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 link/ether 00:16:3e:01:02:02 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 pfifo_fast state UP group default qlen 1000
 link/ether 00:16:3e:01:03:02 brd ff:ff:ff:ff:ff:ff
 inet 192.168.3.102/24 brd 192.168.3.255 scope global eth2
...
```

5. On **vpn-host1**, run the following command to initialize the NSS database in `/etc/ipsec.d`.

```
[vpn-host1]# ipsec initnss --nssdir /etc/ipsec.d
Initializing NSS database
```

6. 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 --nssdir /etc/ipsec.d --output
/etc/ipsec.d/www.example.com.secrets
Generated RSA key pair with CKAID ... was stored in the NSS
database
```

7. On **vpn-host2**, run the following command to initialize the NSS database in `/etc/ipsec.d`.

```
[vpn-host2]# ipsec initnss --nssdir /etc/ipsec.d
Initializing NSS database
```

8. Generate a new RSA authentication key for **vpn-host2**.

- Use the `ipsec newhostkey` command to generate the key.

```
[vpn-host2]# ipsec newhostkey --nssdir /etc/ipsec.d --output
/etc/ipsec.d/www.example.com.secrets
```

Generated RSA key pair with CKAID ... was stored in the NSS database

9. 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 the following command to display all generated keys on **vpn-host1**.

- Sample output is shown.

- In this example, the RSA key is `AwEAAboBh`. Your key might be different.

```
[vpn-host1]# ipsec showhostkey --list
```

< 1> RSA keyid: AwEAAboBh ckaid: ...

- c. Use the following command to display the host key on left host, **vpn-host1**.

- Substitute your RSA key as the argument to the `--rsaid` option. Your key appears in the previous command.
- Sample output is shown.

```
[vpn-host1]# ipsec showhostkey --rsaid AwEAAboBh --left
```

```
rsakey AwEAAboBh
leftrsasigkey=0sAwEAAboBhqXZqWP/5tXPI2xXqR/qq8TPyGUoUQ+rCkHy+WK
q14MrCcmPaHDVZfMIoRAN4Mot2k2535sHnc+SkWxaDyjueGKczTndALmck0eXXWa
WgcfNS94rH9wtleQuZXmT1nSQvW8kiHO1N1o22NrCRYZF8zrpQTNFC1WNAiO2qxW
ZSgdJn2q9iW6MFq0804AsNKI9QrrpC1n7xXyDrWhi+v5B73C01y4/uYeNIotyK9C
ImM713QK3MUpZOSNnRiACIQYw8aX+YEKSgjPU3+nEHp243QeUVraIf5LE0cKtTQu
S3Ur1cgZfQZCFX1rGyHqD/ZtUyzL9Fvo5j04kjnZgJTywr4f0Tmw7a+2QJPIQQ52
iOv1jnV5WzbKB2zpDICsCzRZ7yVaK7MXrDxvbNss8gjXjK5BXgFLcv1Fh/eJgcji
/AUK0S1vqXdYiJjWtZpjznRTDyE7+jqgLsSi0jY5y7i4dYhD+I0RujzTuv6z7ObD
+yLYpa/DoXQFMrFjB3kz9L+uqz7TtmwCthNdCJVJjnKL0jBIZ7IfVqBvIJoS5nra
WYbF/thUq7C6ziHML8AL2tUcx5wIne28ijJOT2LfjeU=
```

- d. 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 --rsaid AwEAAboBh --left
```

```
rsakey AwEAAboBh
leftrsasigkey=0sAwEAAboBhqXZqWP/5tXPI2xXqR/qq8TPyGUoUQ+rCkHy+WK
q14MrCcmPaHDVZfMIoRAN4Mot2k2535sHnc+SkWxaDyjueGKczTndALmck0eXXWa
WgcfNS94rH9wtleQuZXmT1nSQvW8kiHO1N1o22NrCRYZF8zrpQTNFC1WNAiO2qxW
ZSgdJn2q9iW6MFq0804AsNKI9QrrpC1n7xXyDrWhi+v5B73C01y4/uYeNIotyK9C
ImM713QK3MUpZOSNnRiACIQYw8aX+YEKSgjPU3+nEHp243QeUVraIf5LE0cKtTQu
S3Ur1cgZfQZCFX1rGyHqD/ZtUyzL9Fvo5j04kjnZgJTywr4f0Tmw7a+2QJPIQQ52
iOv1jnV5WzbKB2zpDICsCzRZ7yVaK7MXrDxvbNss8gjXjK5BXgFLcv1Fh/eJgcji
```

```
/AUK0S1vqXdYiJjWtZpjznRTDyE7+jggLssI0jY5y7i4dYhD+I0RuJzTuv6z7ObD
+yLYpa/DoXQFMrFjB3kz9L+uqz7TtmwCthNdCJVJjnKL0jbIZ7IfVqBvIJoS5nra
WYbf/thUq7C6ziHML8AL2tUcx5wIne28ijJOT2LfjeU=
```

- e. Use the `vi` editor to edit the `ipsec.conf` file.

```
[vpn-host1]# vi /etc/ipsec.conf
```

- f. Paste the contents of the buffer at the end of the file.

- Position your 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.

- g. On `vpn-host2`, use the following command to display all generated keys on `vpn-host2`.

- Sample output is shown.
- In this example, the RSA key is `AwEAAbJEs`. Your key might be different.

```
[vpn-host2]# ipsec showhostkey --list
< 1> RSA keyid: AwEAAbJEs ckaid:
```

- h. Use the following command to display the host key on right host, `vpn-host2`.

- Substitute your RSA key as the argument to the `--rsaid` option. Your key appears in the previous command.
- Sample output is shown.

```
[vpn-host2]# ipsec showhostkey --rsaid AwEAAbJEs --right
 # rsakey AwEAAbJEs
rightrsasigkey=0sAQPKXwWB4r62JUqcItOtIps5GIkOxOe0n51jz/09Sra5Qth
h1c0WaapVjycZIgDj3tVE4h/UCpBGZbE1Mz7u8DRZjrcv3aXF2CSESJcW8w0hoOD
9SUh3ZvDt1OE5bBwtM7moeJ2iY9rM0OqigRfIMeMKw0ZFdg1xGGmuvfWtJrd886c
GYUFTP3K3+1zblg9v1coOGdfb5jy03jAHgBC2waC1YYAZFQOcHp9XBGVzPq8VkXZ
AnECA8VtPuyExBxt/GBGUGJ0drLjG/HHtweLlqgB3hmy5NzhYiyS8UVpC7RBLpWG
OotjmM2dupw+voGP38bWy8K51T8wfRQbfsbUd84Ga6R7676ZKSZBSMyDsLrsW16
e1tf9sShJ9E6YZ3ZqSt1FsR8zM1ArQhE2gfp+InlQAp1Q7v8TUODy0z1bih407o0
nsYGFxwB9izXGNGrvxoKgvzgleRj7ROP6DA1s/8axdir0N0que975Rc01YM2o0sj
nWwQq124YvenLn1RCbH5fq5NF6V29U7+B5q/2afL6hCvfmQ==
```

- i. 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 --rsaid AwEAAbJEs --right
 # rsakey AwEAAbJEs
rightrsasigkey=0sAQPKXwWB4r62JUqcItOtIps5GIkOxOe0n51jz/09Sra5Qth
h1c0WaapVjycZIgDj3tVE4h/UCpBGZbE1Mz7u8DRZjrcv3aXF2CSESJcW8w0hoOD
9SUh3ZvDt1OE5bBwtM7moeJ2iY9rM0OqigRfIMeMKw0ZFdg1xGGmuvfWtJrd886c
GYUFTP3K3+1zblg9v1coOGdfb5jy03jAHgBC2waC1YYAZFQOcHp9XBGVzPq8VkXZ
AnECA8VtPuyExBxt/GBGUGJ0drLjG/HHtweLlqgB3hmy5NzhYiyS8UVpC7RBLpWG
```

```
OotjmM2dupw+voGP38bWy8K51T8wfRQbfbsbUd84Ga6R7676ZKSZXBSMyDsLrsWl6
e1tf9sShJ9E6YZ3ZqSt1FsR8zM1ArQhE2gfp+InlQAp1Q7v8TUODY0z1bih407o0
nsYGFxwB9izXGNGrvxoKgvzgleRj7ROP6DA1s/8axdir0N0que975Rc01YM2o0sj
nWwQq124YvenLn1RCbH5fq5NF6V29U7+B5q/2afL6hCvfmQ==
```

- j. Use the `vi` editor to create a temporary file named `right`.
- k. 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.
- m. From **vpn-host2**, use the `scp` command to copy the `right` file into the `/etc` directory on **vpn-host1**.
  - Answer `yes` when prompted.
  - Enter the `root` user’s password.

```
[root@vpn-host2 ~]# scp right vpn-host1:/etc
The authenticity of host 'vpn-host1 (192.0.2.107)' can't be
established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vpn-host1,192.0.2.107' (ECDSA) to
the list of known hosts.
root@vpn-host1's password:
right 100% ...
```

- n. 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.
  - Commands shown assume that you are in the `/etc` directory.

```
[vpn-host1]# pwd
/etc
[vpn-host1]# mv ipsec.conf ipsec.BAK
[vpn-host1]# cat ipsec.BAK right > ipsec.conf
```

- o. 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/qq8TPyGUoUQ+rCkHy+WK
q14MrCcmPaHDVZfM1oRAN4Mot2k2535sHnc+SkWxaDyjueGKczTndALmck0eXXWa
WgcfNS94rH9wtleQuZXmTlnSQvW8kiHO1N1o22NrCRYZF8zrpQTNFC1WNAiO2qxW
ZSgdJn2q9iW6MFq0804AsNKI9QrrpC1n7xXyDrWhi+v5B73C0ly4/uYeNIotyK9C
ImM713QK3MUpZOSnRiACIQYw8aX+YEKSgjPU3+nEHp243QeUVraIf5LE0cKtTQu
S3Ur1cgZfQZCFX1rGyHqD/ZtUyzL9Fvo5j04kjnZgJTywr4f0Tmw7a+2QJPIQQ52
iOv1jnV5WzbKB2zpDICsCzRZ7yVaK7MXrDxvbNss8gjXjK5BXgFLcv1Fh/eJgcji
/AUK0S1vqXdYiJjWtZpjznRTDyE7+jqgLsSi0jY5y7i4dYhD+I0RujzTuv6z7ObD
+yLYpa/DoXQFMrFjB3kz9L+uqz7TtmwCthNdCJVJjnKL0jbIZ7IfVqBvIJoS5nra
WYbF/thUq7C6ziHML8AL2tUcx5wIne28ijJOT2LfjeU=
rightrsasigkey=0sAQPXXwWB4r62JUqcItOtIps5GIkOxOe0n51jZ/09Sra5Qth
hlc0WaapVjycZIgDj3tVE4h/UCpBGZbE1MZ7u8DRZjrcv3aXF2CSESJcW8w0hoOD
9SUh3ZvDt1OE5bBWtM7moeJ2iY9rM0OqigRfIMeMKw0ZFdglxGGmuvfWtJrd886c
GYUFTP3K3+1zblg9vlcoOGdfb5jy03jAHgBC2waC1YYAZFQOcHp9XBGVzPq8VkXZ
AnECA8VtPuyExBxt/GBGUGJ0drLjG/HHtweLlqgB3hmy5NzhYiyS8UVpC7RBLpWG
Ootjm2dupw+voGP38bWy8K51T8wfRQbfbsbUd84Ga6R7676ZKSZXBMyDsLrsW16
e1tf9sShJ9E6YZ3ZqSt1FsR8zM1ArQhE2gfp+In1QAp1Q7v8TUODy0z1bih407o0
nsYGFxwB9izXGNGrvxoKgvzgleRj7ROP6DA1s/8axdir0N0que975Rc01YM2o0sj
nWwQq124YvenLn1RCbH5fq5NF6V29U7+B5q/2afL6hCvfmQ==
```

- Complete the “sitetosite” connection configuration in the /etc/ipsec.conf file on **vpn-host1**.

- 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 9e.

```
[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
 lefrsasigkey=...
...
```

- 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.
11. Use the `scp` command to copy the /etc/ipsec.conf file from **vpn-host1** to **vpn-host2**.

- Answer yes when prompted.
- Enter the `root` user’s password.

```

[vpn-host1]# scp /etc/ipsec.conf vpn-host2:/etc
...
root@vpn-host2's password:
ipsec.conf 100% ...

```

12. 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

```

13. 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
```

14. 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.
- If the `netstat` command is not found, use the `yum` command to install the `net-tools` package. Answer `y` to “Is this ok.”

```
[vpn-host1]# netstat -rn
-bash: netstat: command not found

[vpn-host1]# yum install net-tools
...
Is this ok [y/d/N]: y
...
Complete!

[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.
- If the `netstat` command is not found, use the `yum` command to install the `net-tools` package. Answer `y` to “Is this ok.”

```
[vpn-host2]# netstat -rn
-bash: netstat: command not found

[vpn-host2]# yum install net-tools
...
Is this ok [y/d/N]: y
```

```
...
Complete!
```

```
[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
```

15. 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
```

16. 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.

- e. 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
```

- f. 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...
...
```

#### 17. Log out of **vpn-host1** and **vpn-host2**.

```
[vpn-host1]# exit
logout
Connection to vpn-host1 closed.

[vpn-host2]# exit
logout
Connection to vpn-host2 closed.
```

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.

## **Practices for Lesson 10: XFS File System**

## Practices for Lesson 10: Overview

---

### Overview

In these practices, you:

- Create and mount an XFS file system
- Set quotas on an XFS file system
- Backup and restore an XFS file system

## Practice 10-1: Creating an XFS File System

### Overview

In this practice, you:

- View the installed XFS software packages
- Partition /dev/xvdd
- Create an XFS file system
- Mount the file system and copy files to the file system
- Attempt to resize an XFS file system
- Change and view the parameters of an XFS file system

### Assumptions

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

### Tasks

#### 1. Log in to **host03**.

- Use `ssh` to log on to **host03**.

```
ssh host03
root@host03's password:
Last login: ...
```

#### 2. Use the `rpm -qa | grep xfs` command to view the installed XFS software packages.

```
rpm -qa | grep xfs
xfsprogs-4.5.0-15.0.1.el7.x86_64
xfsdump-3.1.7-1.el7.x86_64
```

- Note that both the `xfsprogs` and `xfsdump` packages are installed.

#### 3. Create a new partition on /dev/xvdd.

- Use the `fdisk` command to display the available devices on your system.

```
fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
 /dev/xvda1 * 2048 2099199 1048576 83 Linux
 /dev/xvda2 2099200 41943039 19921920 8e Linux
 LVM
Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
 sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
 sectors
```

- b. Use the `parted` `mkpart` command to create a new primary partition on `/dev/xvdd`.
- The new partition uses all the space on the device.
  - If you get the error shown, “Error: `/dev/xvdd`: unrecognised disk label”, run the `mktabe msdos` command as shown, then continue with the `mkpart` command,

```
parted /dev/xvdd
GNU Parted 3.1
Using /dev/xvdd
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Error: /dev/xvdd: unrecognised disk label
(parted) mktabe msdos
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? ENTER
Start? 0
End? 100%
Warning: The resulting partition is not properly aligned for
best performance.
Ignore/Cancel? i
(parted) quit
Information: You may need to update /etc/fstab.
```

- c. Use the `fdisk` command to display the partition table on all available devices.

```
fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvda1 * 2048 2099199 1048576 83 Linux
/dev/xvda2 2099200 41943039 19921920 8e Linux
LVM

Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvdd1 1 41943039 20971519+ 83 Linux
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

- Note that the `/dev/xvdd1` partition is shown.

#### 4. Create an XFS file system.

- a. Use the `mkfs.xfs` command without any arguments or options to display the usage.

```
mkfs.xfs
No device name given in argument list
Usage: mkfs.xfs
```

```
/* blocksize */ [-b log=n|size=num]
/* metadata */ [-m crc=[0|1]
/* data subvol */ [-d agcount=n,agsize=n,file,name=xxx...
/* force overwrite */ [-f]
/* inode size */ [-i log=n|perblock=n|size=num,maxpct...
/* no discard */ [-K]
/* log subvol */ [-l agnum=n,internal,size=num,logdev...
/* label */ [-L label (maximum 12 characters)]
...
```

- b. Use the `mkfs.xfs` command to create an XFS file system on `/dev/xvdd1`.

- Use the `-L XFS` option to specify a label of XFS.
- Use the `-b size=512` option to specify a block size of 512 bytes.

```
mkfs.xfs -L XFS -b size=512 /dev/xvdd1
meta-data=/dev/xvdd1 isize=256 agcount=4, agsize=...
 = sectsz=512 attr=2, projid32bit=1
 = crc=0 finobt=0, sparse=0
data = bsize=512 blocks=10485758, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0
log =internal log bsize=512 blocks=20480, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
```

- Note that the block size of the `data` and the `log` sections is 512 bytes. The default is 4 KB.
  - Note that the block size of the `naming` (file system directory) section is 4 KB.
- c. Use the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
blkid | grep xfs
...
/dev/xvdd1: LABEL="XFS" UUID=... TYPE="xfs"
...
```

5. Mount the XFS file system on the `/Test` mount point and verify usability.

- a. Create the `/Test` mount point.

```
mkdir /Test
```

- b. Use the `vi` editor to add the following entry at the end of `/etc/fstab` to mount the file system:

```
vi /etc/fstab
...
LABEL=XFS /Test xfs defaults 0 0
```

- c. Use the `mount -a` command to mount the file systems in `/etc/fstab`.

```
mount -a
```

- d. Use the `df -h` command to display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/xvdd1 20G 4.1M 20G 1% /Test
```

- e. Use the `mount` command to view the `/dev/xvdd1` partition.

```
mount | grep /dev/xvdd1
/dev/xvdd1 on /Test type xfs (rw,relatime,seclabel,attr2,...)
```

- f. Use the `cp` command to copy some files to `/Test`. Use the `ls` command to list the contents of the `/Test` directory.

- Copy `/boot/vmlinuz*` to `/Test`.

```
cp /boot/vmlinuz* /Test
ls /Test
vmlinuz-0-rescue-8e93dd201df146c38412b6ec7dbb2821
vmlinuz-3.10.0-862.el7.x86_64
vmlinuz-4.1.12-112.16.4.el7uek.x86_64
```

- This confirms that the XFS file system is usable.

6. Attempt to increase the size of an XFS file system.

- Use the `xfs_growfs` command to expand the data section to use all available space.

```
xfs_growfs -d /Test
meta-data=/dev/xvdd1 isize=256 agcount=4, agsize=...
 = sectsz=512 attr=2, projid32bit=1
 = crc=0 finobt=0 spinodes=0
data = bsize=512 blocks=10485758, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0
log =internal log bsize=512 blocks=20480, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
data size unchanged, skipping
```

- Notice the “data size unchanged, skipping” message.
- The data section was not changed because there is no space available on the underlying device, `/dev/xvdd1`.
- The `xfs_growfs` command is most often used with logical volumes.

7. Change and view the parameters of an XFS file system.

- a. Use the `xfs_admin` command to display the file system label and the UUID.

```
xfs_admin -l /dev/xvdd1
label = "XFS"
xfs_admin -u /dev/xvdd1
UUID = ...
```

- b. Use the `xfs_admin` command to change the file system label to “`xfs`” (all lowercase).

```
xfs_admin -L xfs /dev/xvdd1
xfs_admin: /dev/xvdd1 contains a mounted filesystem
fatal error - couldn't initialize XFS library
```

- Note that the command failed because the file system is mounted.
- c. Use the `umount /Test` command to unmount the file system.

```
umount /Test
```

- d. Repeat the previous step to change the file system label to “`xfs`”.

```
xfs_admin -L xfs /dev/xvdd1
Writing all SBs
new label = "xfs"
```

- Note that the command is successful when the file system is not mounted.
- e. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
mount -a
mount: can't find LABEL=XFS
```

- The `/etc/fstab` entry mounts the file system by label.
- Because the label changed, the entry in `/etc/fstab` is no longer valid.
- f. Use the `vi` editor to edit the following entry in `/etc/fstab` to mount the file system:

- Change `LABEL=XFS` to `/dev/xvdd1`.

```
vi /etc/fstab
...
LABEL=XFS /Test xfs defaults 0 0 (old entry)
/dev/xvdd1 /Test xfs defaults 0 0 (new value)
```

- g. Use the `mount -a` command to mount the file systems in `/etc/fstab`.

```
mount -a
```

- h. Use the `blkid` command to display the new label on the XFS file system.

```
blkid | grep xfs
...
/dev/xvdd1: LABEL="xfs" UUID=... TYPE="xfs"
...
```

- Note that the label is now in lowercase.

## Practice 10-2: Setting Disk Quotas on an XFS File System

---

### Overview

In this practice, you:

- Enable user quotas on an XFS file system
- Use the `xfs_quota` command in interactive mode
- Set limits for the `oracle` user and view quota information
- Verify the quota settings limit disk space to the `oracle` user
- Remove disk quotas

### Assumptions

- You are the `root` user on `host03`.

### Tasks

1. Enable quotas for users on an XFS file system.

- This step directs you to edit the `/etc/fstab` file to enable quotas.
- Alternatively, you could use the `mount` command and include XFS quota mount options.
- a. Use the `mount` command to view the mount options for the XFS file system.

```
mount | grep " xfs "
...
/dev/xvdd1 on /Test type xfs (rw,relatime,seclabel,...,noquota)
...
```

- Notice the “`noquota`” mount option.
- b. Use the `umount` command to unmount `/Test`.

```
umount /Test
```

- c. Use the `vi` editor to change the mount option for the XFS file system.
- Change “`defaults`” to “`rw, quota`” for the `/dev/xvdd1` entry.

```
vi /etc/fstab
...
/dev/xvdd1 /Test xfs defaults 0 0 (old value)
/dev/xvdd1 /Test xfs rw,quota 0 0 (new value)
```

- d. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
mount -a
```

- e. Use the `mount` command to view the new mount options for the XFS file system.

```
mount | grep " xfs "
...
/dev/xvdd1 on /Test type xfs (rw,relatime,seclabel,...,usrquota)
```

- Notice the new “`usrquota`” mount option.

- f. Use the `xfs_quota` command to report overall state information.

```
xfs_quota -x -c state
User quota state on /Test (/dev/xvdd1)
 Accounting: ON
 Enforcement: ON
 Inode: #38 (1 blocks, 1 extents)
Group quota state on /Test (/dev/xvdd1)
 Accounting: OFF
 Enforcement: OFF
 Inode: #0 (0 blocks, 0 extents)
Project quota state on /Test (/dev/xvdd1)
 Accounting: OFF
 Enforcement: OFF
 Inode: #0 (0 blocks, 0 extents)
Blocks grace time: [7 days]
Inodes grace time: [7 days]
Realtime Blocks grace time: [7 days]
```

- Note that accounting and enforcement are ON for user quota but OFF for group and project quota.

2. Use the `xfs_quota` command in interactive mode.

- a. Enter the `xfs_quota` command without any options or arguments.

- The `xfs_quota>` prompt is displayed.

```
xfs_quota
xfs_quota>
```

- b. Enter `help` at the `xfs_quota>` prompt to display online help for the commands.

```
xfs_quota> help
df [-bir] [-hn] [-f file] -- show free and used counts for
blocks and inodes
help [command] -- help for one or all commands
print -- list known mount points and projects
quit -- exit the program
quota [-bir] [-g|-p|-u] [-hnNv] [-f file] [id|name]... -- show
usage and limits

Use 'help commandname' for extended help.
```

- c. Enter `help quota` at the `xfs_quota>` prompt to display extended help for the quota command.

```
xfs_quota> help quota
quota [-bir] [-g|-p|-u] [-hnNv] [-f file] [id|name]... -- show
usage and limits

display usage and quota information
```

```

-g -- display group quota information
-p -- display project quota information
-u -- display user quota information
-b -- display number of blocks used
-i -- display number of inodes used
-r -- display number of realtime blocks used
-h -- report in a human-readable format
-n -- skip identifier-to-name translations, just report IDs
-N -- suppress the initial header
-v -- increase verbosity in reporting (also dumps zero values)
-f -- send output to a file
The (optional) user/group/project can be specified either by
name or by
number (i.e. uid/gid/projid).

```

- d. Enter `quota -b` at the `xfs_quota>` prompt to display the number of blocks used.

```

xfs_quota> quota -b
Disk quotas for User root (0)
Filesystem Blocks Quota Limit Warn/Time
Mounted on
/dev/xvdd1 18321 0 0 00 [-----]
---] /Test

```

- e. Enter `quota -i` at the `xfs_quota>` prompt to display the number of inodes used.

```

xfs_quota> quota -i
Disk quotas for User root (0)
Filesystem Files Quota Limit Warn/Time
Mounted on
/dev/xvdd1 6 0 0 00 [-----]
---] /Test

```

- f. Enter `quit` at the `xfs_quota>` prompt to exit interactive mode.

```
xfs_quota> quit
```

3. Set limits for the `oracle` user and view quota information.

- a. Use the `xfs_quota` command to set a soft limit of 5 MB and a hard limit of 6 MB on the XFS file system for the `oracle` user.

```
xfs_quota -x -c 'limit -u bsoft=5m bhard=6m oracle' /Test
```

- b. Use the `xfs_quota` command to print all paths with devices and identifiers.

```
xfs_quota -x -c print
Filesystem Pathname
...
/Test /dev/xvdd1 (uquota)
```

- c. Use the `xfs_quota` command to report file system usage for blocks.

- The **-h** option provides human-readable output.
- The **-b** option provides block information.

```
xfs_quota -x -c 'free -hb'
Filesystem Size Used Avail Use% Pathname
...
/dev/xvdd1 20.0G 21.9M 20.0G 0% /Test
```

- d. Use the `xfs_quota` command to report file system usage for inodes.

- The **-i** option provides inode information.

```
xfs_quota -x -c 'free -hi'
Filesystem Inodes Used Free Use% Pathname
...
/dev/xvdd1 21.0m 7 21.0m 0% /Test
```

- e. Use the `xfs_quota` command to report file system quota information in human-readable form.

```
xfs_quota -x -c report /Test
User quota on /Test (/dev/xvdd1)
 Blocks
User ID Used Soft Hard Warn/Grace

root 18321 0 0 00 [-----]
oracle 0 5120 6144 00 [-----]
```

#### 4. Test quota settings.

- a. Use the `chmod` command to give read, write, and execute permissions on the `/Test` directory. Use the `ls -d /Test` command to view the permissions on the `/Test` directory.

```
chmod 777 /Test
ls -ld /Test
drwxrwxrwx ... /Test
```

- b. Use the `su - oracle` command to become the `oracle` user.

```
su - oracle
[oracle@host03 ~]$
```

- c. As the `oracle` user, use the `cd` command to change to the `/Test` directory.

```
[oracle@host03 ~]$ cd /Test
[oracle@host03 Test]$
```

- d. As the `oracle` user, use the `dd if=/dev/zero of=bigfile bs=1K count=7168` command to attempt to create a 7 MB file.

- The command fails because the `oracle` user has a hard limit of 6 MB on the XFS file system.

```
[oracle@host03 Test]$ dd if=/dev/zero of=bigfile bs=1K
count=7168
dd: error writing \u2018bigfile\u2019: Disk quota exceeded
```

```
6145+0 records in
6144+0 records out
6291456 bytes (6.3 MB) copied, 0.0358975 s, 175 MB/s
```

- Notice the “Disk quota exceeded” error message.
- e. Use the `exit` command to log off as the `oracle` user. Use the `whoami` command to confirm that you are the `root` user.

```
[oracle@host03 Test]$ exit
logout
whoami
root
```

- f. Use the `xfs_quota` command to report file system quota information in human-readable form.

```
xfs_quota -x -c report /Test
User quota on /Test (/dev/xvdd1)
 Blocks
User ID Used Soft Hard Warn/Grace

root 18325 0 0 00 [-----]
oracle 6144 5120 6144 00 [7 days]
```

- Note that the `Used` column for the `oracle` user is 6M.
5. Disable XFS quotas and re-test.

- a. Use the `umount` command to unmount `/Test`.

```
umount /Test
```

- b. Use the `vi` editor to change the mount option for the XFS file system.

- Change “`rw, quota`” to “`defaults`” for the `/dev/xvdd1` entry.

```
vi /etc/fstab
...
/dev/xvdd1 /Test xfs rw,quota 0 0 (old value)
/dev/xvdd1 /Test xfs defaults 0 0 (new value)
```

- c. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
mount -a
```

- d. Use the `mount` command to view the new mount options for the XFS file system.

```
mount | grep " xfs "
...
/dev/xvdd1 on /Test type xfs (rw,relatime,seclabel,...,noquota)
```

- Note that the mount option shows “`noquota`”.

- e. Use the `su - oracle` command to become the `oracle` user.

```
su - oracle
[oracle@host03 ~]$
```

- f. As the `oracle` user, use the `cd` command to change to the `/Test` directory.

```
[oracle@host03 ~]$ cd /Test
[oracle@host03 Test]$
```

- g. As the oracle user, use the dd if=/dev/zero of=bigfile bs=1K count=7168 command to attempt to create a 7 MB file.

```
[oracle@host03 Test]$ dd if=/dev/zero of=bigfile bs=1K
count=7168
7168+0 records in
7168+0 records out
7340032 bytes (7.3 MB) copied, 0.321967 s, 22.8 MB/s
```

- The file is successfully created with no disk quotas.
- h. Use the exit command to log off as the oracle user. Use the whoami command to confirm that you are the root user.

```
[oracle@host03 Test]$ exit
logout
whoami
root
```

## Practice 10-3: Backing Up and Restoring XFS File Systems

### Overview

In this practice, you:

- Use `xfsdump` to back up an XFS file system
- Use `xfsdump` to back up a specific file in an XFS file system
- View the `xfsdump` inventory
- Use `xfsrestore` to restore from a backup

### Assumptions

- You are the `root` user on `host03`.

### Tasks

1. Use the `xfsdump` utility.

- a. Run the `xfsdump -h` command to display the usage.

```
xfsdump -h
xfsdump: version 3.1.7 (dump format 3.0)
xfsdump: usage: xfsdump [-a (dump DMF dualstate files as ...
 [-b <blocksize>]
 [-c <media change alert program>]
 [-d <dump media file size>]
 [-e (allow files to be excluded)]
 [-f <destination> ...]
 [-h (help)]
 [-l <level>]
 [-m (force usage of minimal rmt)]
 [-o (overwrite tape)]
 [-p <seconds between progress ...]
 [-q <use QIC tape settings>]
...
...
```

- b. Use the `xfsdump` command to back up the entire XFS file system on `/Test` to a local file, `/usr/tmp/backup`.

- Omitting the `-l <level>` option performs a full backup (level 0).
- Enter a session label of your choice when prompted.
- Enter a media label of your choice when prompted.
- These labels could be provided as options to the `xfsdump` command:
  - `(-L <session_label>)`
  - `(-M <media_label>)`

```
xfsdump -f /usr/tmp/backup /Test
xfsdump: using file dump (drive_simple) strategy
```

```
xfsdump: version 3.1.7 (dump format 3.0) - type ^C for status
and control

===== dump label dialog
=====

please enter label for this dump session (timeout in 300 sec)
-> Full backup of /Test on 1/21/2019
session label entered: "Full backup of /Test on 1/21/2019"

----- end dialog -----

xfsdump: level 0 dump of host03.example.com:/Test
xfsdump: dump date: Tue Feb 26 18:00:09 2019
xfsdump: session id: 21505e0f-2388-459e-903e-8bd388364da5
xfsdump: session label: "Full backup of /Test on 1/21/2019"
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: skipping (no pruning necessary)
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 26123328 bytes
xfsdump: /var/lib/xfsdump/inventory created

===== media label dialog
=====

please enter label for media in drive 0 (timeout in 300 sec)
-> Full backup to /usr/tmp/backup on 1/21/2019
media label entered: "Full backup to /usr/tmp/backup on
1/21/2019"

----- end dialog -----

xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 26133104 bytes
xfsdump: dump size (non-dir files) : 26109384 bytes
xfsdump: dump complete: 51 seconds elapsed
xfsdump: Dump Summary:
```

```
xfsdump: stream 0 /usr/tmp/backup OK (success)
xfsdump: Dump Status: SUCCESS
```

- c. Use the `xfsdump` command to back up only `/Test/bigfile` to `/usr/tmp/bigfile_backup`.
  - Provide a session label of your choice as a command-line argument.
  - Provide a media label of your choice as a command-line argument.

```
xfsdump -f /usr/tmp/bigfile_backup -L "session label" -M
"media label" -s bigfile /Test
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.1.7 (dump format 3.0) - type ^C for status
and control
xfsdump: level 0 dump of host03.example.com:/Test
xfsdump: dump date: Tue Feb 26 18:02:39 2019
xfsdump: session id: 514c9da2-5c2a-4cee-b846-41289c9ef9df
xfsdump: session label: "session label"
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: skipping (no pruning necessary)
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 7361152 bytes
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 7365232 bytes
xfsdump: dump size (non-dir files) : 7343384 bytes
xfsdump: dump complete: 0 seconds elapsed
xfsdump: Dump Summary:
xfsdump: stream 0 /usr/tmp/bigfile_backup OK (success)
xfsdump: Dump Status: SUCCESS
```

- Note that you were not prompted for a session label or a media label.
- d. Use the `xfsdump -I` command to display the inventory.
  - Only selected lines are shown from the output.

```
xfsdump -I
file system 0:
 fs id: 6d1e5ada-e6d9-4431-90c9-670eb25c1138
 session 0:
 mount point: host03.example.com:/Test
 device: host03.example.com:/dev/xvdd1
 time: Tue Feb 26 18:00:09 2019
 session label: "Full backup of /Test on 1/21/2019"
```

```
session id: 21505e0f-2388-459e-903e-8bd388364da5
level: 0
resumed: NO
subtree: NO
streams: 1
stream 0:
 pathname: /usr/tmp/backup
 start: ino 35 offset 0
 end: ino 40 offset 0
 interrupted: NO
 media files: 1
 media file 0:
 mfile index: 0
 mfile type: data
 mfile size: 26133104
 mfile start: ino 35 offset 0
 mfile end: ino 40 offset 0
 media label: "Full backup to
/usr/tmp/backup on 1/21/2019"
 media id: 27e31f2f-5ecb-4eeb-a468-
94c7b406e3e1
session 1:
 mount point: host03.example.com:/Test
 device: host03.example.com:/dev/xvdd1
 time: Tue Feb 26 18:02:39 2019
 session label: "session label"
 session id: 514c9da2-5c2a-4cee-b846-41289c9ef9df
 level: 0
 resumed: NO
 subtree: YES
 streams: 1
 stream 0:
 pathname: /usr/tmp/bigfile_backup
 start: ino 39 offset 0
 end: ino 40 offset 0
 interrupted: NO
 media files: 1
 media file 0:
 mfile index: 0
 mfile type: data
 mfile size: 7365232
 mfile start: ino 39 offset 0
 mfile end: ino 40 offset 0
```

```

 media label: "media label"
 media id: c2d42359-7585-47a8-8089-
6f01078d43bd
xfsdump: Dump Status: SUCCESS

```

- Note that there are two sessions:

- One session for the full backup of the /Test directory
- One session for the backup of the single file, bigfile, in the /Test directory

- Make a change to the bigfile file in the /Test directory. Use the ls -l command before and after making the change to note the difference in file size.
- The change to bigfile is made by copying /etc/fstab and overwriting the original bigfile.

```

cd /Test
ls -l bigfile
-rw-rw-r-- ... 7340032 ... bigfile
cp /etc/fstab bigfile
cp: overwrite 'bigfile'? y
ls -l bigfile
-rw-rw-r-- ... 517 ... bigfile

```

- In this example, the original size of bigfile was 7340032 bytes. The new size is 517 bytes.
- Use the xfsdump command to perform a level 1 backup of the entire XFS file system on /Test to a local file, /usr/tmp/level1\_backup.
- Provide a session label of your choice as a command-line argument.
- Provide a media label of your choice as a command-line argument.

```

xfsdump -1 1 -f /usr/tmp/level1_backup -L "level 1 session" -M
"level 1 media" /Test
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.1.7 (dump format 3.0) - type ^C for status
and control
xfsdump: level 1 incremental dump of host03.example.com:/Test
based on level 0 dump begun Tue Feb 26 18:00:09 2019
xfsdump: dump date: Tue Feb 26 18:06:06 2019
xfsdump: session id: 70c1b0ec-affe-43e5-9f6c-d9be130cd971
xfsdump: session label: "level 1 session"
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: pruning unneeded subtrees
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 22144 bytes
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories

```

```

xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 22904 bytes
xfsdump: dump size (non-dir files) : 1056 bytes
xfsdump: dump complete: 0 seconds elapsed
xfsdump: Dump Summary:
xfsdump: stream 0 /usr/tmp/level1_backup OK (success)
xfsdump: Dump Status: SUCCESS

```

- g. Use the `xfsdump -I` command to display the inventory.

- Only selected lines are shown from the output.

```

xfsdump -I
file system 0:
 fs id: 6d1e5ada-e6d9-4431-90c9-670eb25c1138
 session 0:
 mount point: host03.example.com:/Test
 device: host03.example.com:/dev/xvdd1
 time: Tue Feb 26 18:00:09 2019
 session label: "Full backup of /Test on 1/21/2019"
 session id: 21505e0f-2388-459e-903e-8bd388364da5
 level: 0
 resumed: NO
 subtree: NO
 streams: 1
 stream 0:
 pathname: /usr/tmp/backup
 start: ino 35 offset 0
 end: ino 40 offset 0
 interrupted: NO
 media files: 1
 media file 0:
 mfile index: 0
 mfile type: data
 mfile size: 26133104
 mfile start: ino 35 offset 0
 mfile end: ino 40 offset 0
 media label: "Full backup to
/usr/tmp/backup on 1/21/2019"
 media id: 27e31f2f-5ecb-4eeb-a468-
94c7b406e3e1
 session 1:
 mount point: host03.example.com:/Test
 device: host03.example.com:/dev/xvdd1

```

```
time: Tue Feb 26 18:02:39 2019
session label: "session label"
session id: 514c9da2-5c2a-4cee-b846-41289c9ef9df
level: 0
resumed: NO
subtree: YES
streams: 1
stream 0:
 pathname: /usr/tmp/bigfile_backup
 start: ino 39 offset 0
 end: ino 40 offset 0
 interrupted: NO
 media files: 1
 media file 0:
 mfile index: 0
 mfile type: data
 mfile size: 7365232
 mfile start: ino 39 offset 0
 mfile end: ino 40 offset 0
 media label: "media label"
 media id: c2d42359-7585-47a8-8089-
6f01078d43bd
session 2:
 mount point: host03.example.com:/Test
 device: host03.example.com:/dev/xvdd1
 time: Tue Feb 26 18:06:06 2019
 session label: "level 1 session"
 session id: 70c1b0ec-affe-43e5-9f6c-d9be130cd971
 level: 1
 resumed: NO
 subtree: NO
 streams: 1
 stream 0:
 pathname: /usr/tmp/level1_backup
 start: ino 39 offset 0
 end: ino 40 offset 0
 interrupted: NO
 media files: 1
 media file 0:
 mfile index: 0
 mfile type: data
 mfile size: 22904
 mfile start: ino 39 offset 0
```

```

 mfile end: ino 40 offset 0
 media label: "level 1 media"
 media id: ea29a746-a76d-4090-a385-
2b215a01b3c9
xfsdump: Dump Status: SUCCESS

```

- Note that the last backup is a level 1 backup.
2. Use the `xfsrestore` utility.
- Run the `xfsrestore -h` command to display the usage.

```

xfsrestore -h
xfsrestore: version 3.1.7 (dump format 3.0)
xfsrestore: usage: xfsrestore [-a <alt. workspace dir> ...]
 [-b <blocksize>]
 [-c <media change alert ...]
 [-e (don't overwrite existing ...]
 [-f <source> ...]
 [-h (help)]
 [-i (interactive)]
 [-m (force usage of minimal ...]
 [-n <file> (restore only if ...]
 [-o (restore owner/group even ...]
 [-p <seconds between progress ...]
 [-q <use QIC tape settings>]
...

```

- Use the `ls -l /Test` command to view the contents and file sizes of the XFS file system before initiating an `xfsrestore`.

```

ls -l /Test
-rw-rw-r-- ... 517 ... bigfile
-rwxr-xr-x ... 6383280 ... vmlinuz-0-rescue-...
-rwxr-xr-x ... 6383280 ... vmlinuz-3.10.0-862.el7.x86_64
-rwxr-xr-x ... 5993952 ... vmlinuz-4.1.12-112.16.4.el7uek.x86_64

```

- Note that the size of `bigfile` is 517 bytes.
- c. Use the `xfsrestore` command to restore from `/usr/tmp/backup` to the `/Test` directory.
- `/usr/tmp/backup` is a level 0 backup.

```

xfsrestore -f /usr/tmp/backup /Test
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.7 (dump format 3.0) - type ^C for status
and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: host03.example.com

```

```

xfsrestore: mount point: /Test
xfsrestore: volume: /dev/xvdd1
xfsrestore: session time: Tue Feb 26 18:00:09 2019
xfsrestore: level: 0
xfsrestore: session label: "Full backup of /Test on 1/21/2019"
xfsrestore: media label: "Full backup to /usr/tmp/backup on
1/21/2019"
xfsrestore: file system id: 6d1e5ada-e6d9-4431-90c9-670eb25c1138
xfsrestore: session id: 21505e0f-2388-459e-903e-8bd388364da5
xfsrestore: media id: 27e31f2f-5ecb-4eeb-a468-94c7b406e3e1
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 1 directories and 4 entries processed
xfsrestore: directory post-processing
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 0 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore: stream 0 /usr/tmp/backup OK (success)
xfsrestore: Restore Status: SUCCESS

```

- d. Use the `ls -l /Test` command to view the contents and file sizes of the XFS file system.

```

ls -l /Test
-rw-rw-r-- ... 7340032 ... bigfile
-rwxr-xr-x ... 6383280 ... vmlinuz-0-rescue-...
-rwxr-xr-x ... 6383280 ... vmlinuz-3.10.0-862.el7.x86_64
-rwxr-xr-x ... 5993952 ... vmlinuz-4.1.12-112.16.4.el7uek.x86_64

```

- Note that the size of `bigfile` is now 7340032 bytes.
- Recall that the level 0 backup was done before changing the size of `bigfile` in the previous step.

- e. Use the `rm` command to remove the `vmlinuz-3.10.0-693.el7.x86_64` file in the `/Test` directory.

```

rm /Test/vmlinuz-3.10.0-862.el7.x86_64
rm: remove regular file '/Test/vmlinuz-3.10.0-862.el7.x86_64'? y

```

- f. Use the `xfsrestore` command to restore from `/usr/tmp/level1_backup` to the `/Test` directory.

- `/usr/tmp/level1_backup` is a level 1 backup.

```

xfsrestore -f /usr/tmp/level1_backup /Test
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.7 (dump format 3.0) - type ^C for status
and control
xfsrestore: searching media for dump

```

```

xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: host03.example.com
xfsrestore: mount point: /Test
xfsrestore: volume: /dev/xvdd1
xfsrestore: session time: Tue Feb 26 18:06:06 2019
xfsrestore: level: 1
xfsrestore: session label: "level 1 session"
xfsrestore: media label: "level 1 media"
xfsrestore: file system id: 6d1e5ada-e6d9-4431-90c9-670eb25c1138
xfsrestore: session id: 70c1b0ec-affe-43e5-9f6c-d9be130cd971
xfsrestore: media id: ea29a746-a76d-4090-a385-2b215a01b3c9
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 1 directories and 4 entries processed
xfsrestore: directory post-processing
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 0 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore: stream 0 /usr/tmp/level1_backup OK (success)
xfsrestore: Restore Status: SUCCESS

```

- g. Use the `ls -l /Test` command to view the contents and file sizes of the XFS file system.

```

ls -l /Test
-rw-rw-r-- ... 517 ... bigfile
-rwxr-xr-x ... 6383280 ... vmlinuz-0-rescue-...
-rwxr-xr-x ... 5993952 ... vmlinuz-4.1.12-112.16.4.el7uek.x86_64

```

- Recall that the `level 1 xfsdump` to `/usr/tmp/level1_backup` (performed in the previous step) backed up only the file that changed (`bigfile`) since the previous backup.
- Therefore, only the `bigfile` (517 bytes) was restored.
- The `vmlinuz-3.10.0-862.el7.x86_64` file was not restored because it was not backed up in the `level 1 xfsdump`.

- h. Use the `xfsrestore` utility to restore from `/usr/tmp/backup`, and then restore from `/usr/tmp/level1_backup`.

- Use the `ls -l` command after each restore to display the contents of `/Test`.

```

xfsrestore -f /usr/tmp/backup /Test
...
xfsrestore: Restore Status: SUCCESS

ls -l /Test

```

```

-rw-rw-r-- ... 7340032 ... bigfile
-rwxr-xr-x ... 6383280 ... vmlinuz-0-rescue-...
-rwxr-xr-x ... 6383280 ... vmlinuz-3.10.0-862.el7.x86_64
-rwxr-xr-x ... 5993952 ... vmlinuz-4.1.12-12.16.4.el7uek.x86_64

xfsrestore -f /usr/tmp/level1_backup /Test
...
xfsrestore: Restore Status: SUCCESS

ls -l /Test
-rw-rw-r-- ... 517 ... bigfile
-rwxr-xr-x ... 6383280 ... vmlinuz-0-rescue-...
-rwxr-xr-x ... 6383280 ... vmlinuz-3.10.0-862.el7.x86_64
-rwxr-xr-x ... 5993952 ... vmlinuz-4.1.12-112.16.4.el7uek.x86_64

```

- All files are restored, including the latest copy of `bigfile` (517 bytes).
3. Clean up before proceeding to the next lesson.
- Use the `umount` command to unmount `/Test`.
    - Use the `cd` command to ensure that you are not in the `/Test` directory.

```

cd
umount /Test

```

- Remove the `/Test` directory.

```
rmdir /Test
```

- Use the `vi` editor to delete the following entry from `/etc/fstab`:

```

vi /etc/fstab
/dev/xvdd1 /Test xfs defaults 0 0 (delete this line)

```

- Use the `rm` command to remove the `xfsdump` files in `/usr/tmp`.

```

cd /usr/tmp
rm backup bigfile_backup level1_backup
rm: remove regular file 'backup'? y
rm: remove regular file 'bigfile_backup'? y
rm: remove regular file 'level1_backup'? y

```

- Delete the `/dev/xvdd1` partition using `fdisk`.

```

fdisk /dev/xvdd
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

```

```
Command (m for help): p
```

```
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0007778c
```

Device	Boot	Start	End	Blocks	Id	System
/dev/xvdd1		1	41943039	20971519+	83	Linux

```
Command (m for help): d
Selected partition 1
Partition 1 is deleted
```

```
Command (m for help): p
```

```
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0007778c
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

```
Command (m for help): w
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
Syncing disks.
```

#### 4. Log off **host03**.

- Use the **exit** command to log off of **host03**.

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

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.

## **Practices for Lesson 11: Btrfs File System**

## Practices for Lesson 11: Overview

---

### Practices Overview

In these practices, you:

- Verify that the `btrfs-progs` package is installed
- Create Btrfs file systems
- Resize a Btrfs file system
- Create and mount Btrfs subvolumes and snapshots
- Take a snapshot of a file in a Btrfs subvolume
- Recover a corrupted Btrfs file system

## Practice 11-1: Creating a Btrfs File System

### Overview

In this practice, you:

- Create a single-disk Btrfs file system with different specifications
- Mount the file system, copy a file to it, and display the file system information
- Resize a Btrfs file system
- Create a two-disk Btrfs file system with different specifications
- Mount the file system, copy a file to it, and display the file system information

### Assumptions

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

### Tasks

1. Log in to `host03` and view the files provided by the `btrfs-progs` package.

- a. Use `ssh` to log in to `host03`, providing the root password.

```
ssh host03
root@host03's password:
Last login: ...
```

- b. Use the `rpm -qf` command to view the contents of the `btrfs-progs` package.

```
rpm -qf btrfs-progs
/usr/lib/udev/rules.d/64-btrfs-dm.rules
/usr/lib64/libbtrfs.so.0
/usr/lib64/libbtrfs.so.0.1
/usr/sbin/btrfs
/usr/sbin/btrfs-convert
/usr/sbin/btrfs-debug-tree
/usr/sbin/btrfs-find-root
/usr/sbin/btrfs-image
/usr/sbin/btrfs-map-logical
/usr/sbin/btrfs-select-super
/usr/sbin/btrfs-zero-log
/usr/sbin/btrfsck
/usr/sbin/btrfstune
/usr/sbin/fsck.btrfs
/usr/sbin/mkfs.btrfs
/usr/share/doc/btrfs-progs-4.9.1
/usr/share/doc/btrfs-progs-4.9.1/COPYING
/usr/share/man/man5/btrfs.5.gz
...
/usr/share/man/man8/mkfs.btrfs.8.gz
```

2. Create a single-disk Btrfs file system.

- a. Use the `fdisk -l` command to display the available devices on your system.

```
fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvda1 * 2048 2099199 1048576 83 Linux
/dev/xvda2 2099200 41943039 19921920 8e Linux
LVM
Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

- The `/dev/xvdc` and `/dev/xvdd` devices have no partitions. We will use the entire devices in these practices.

- b. Use the `mkfs.btrfs` command to make a Btrfs file system on `/dev/xvdc`.

```
mkfs.btrfs -L Btrfs /dev/xvdc
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.

Detected a SSD, turning off metadata duplication. Mkfs with -m
dup if you want to force metadata duplication.
Label: Btrfs
UUID: ...
Node size: 16384
Sector size: 4096
Filesystem size: 20.00GiB
Block group profiles:
 Data: single 8.00MiB
 Metadata: single 8.00MiB
 System: single 4.00MiB
SSD detected: yes
Incompat features: extref
Number of devices: 1
Devices:
 ID SIZE PATH
 1 20.00GiB /dev/xvdc
```

- In this example, metadata duplication is turned off.
  - “Detected a SSD” is technically not a bug. The SSD detection code in Btrfs is looking for rotational disks, which a Xen-backed image is not. You can check this by looking at `/sys/class/block/xvdc/queue/rotational`. It will be set to 0. If you want Btrfs to think the disk is rotational, set this to 1. This happens on all HVM with PV and PVM virtual machines using the `xvd` block devices.
- c. Repeat the previous command but include the `-m dup` option.
- Also include the `-f` option to force overwrite of an existing filesystem.

```
mkfs.btrfs -f -m dup -L Btrfs /dev/xvdc
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.

Label: Btrfs
UUID: ...
Node size: 16384
Sector size: 4096
Filesystem size: 20.00GiB
Block group profiles:
 Data: single 8.00MiB
 Metadata: DUP 1.00GiB
 System: DUP 8.00MiB
SSD detected: yes
Incompat features: extref
Number of devices: 1
Devices:
 ID SIZE PATH
 1 20.00GiB /dev/xvdc
```

- d. Run the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
blkid | grep btrfs
/dev/xvdc: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
```

- e. Use the `mkdir` command to make the `/btrfs` directory. This directory is used as the mount point.

```
mkdir /btrfs
```

- f. Use the `vi` editor to add the following entry in `/etc/fstab` to mount the file system:

```
vi /etc/fstab
...
/dev/xvdc /btrfs btrfs defaults 0 0
```

- You can use LABEL or UUID when mounting as well.

- g. Use the `mount` command to mount the file systems in `/etc/fstab`.

```
mount -a
```

- h. Run the `df -h` command to display the space information for `/btrfs`.

```
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 17M 18G 1% /btrfs
```

- i. Use the `cp` command to copy `/boot/vmlinuz-3.10*` to `/btrfs`.

```
cp /boot/vmlinuz-3.10* /btrfs
```

- j. Run the `df -h` command again to display the space information for `/btrfs`.

```
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 17M 18G 1% /btrfs
```

- Note that there is no difference in the amount of space used and available.

- k. Run the `sync` command to flush the file system buffers. Run the `df -h /btrfs` command again.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 23M 18G 1% /btrfs
```

- Note that the `sync` command resulted in a difference in the output of `df -h`.
- Running the `sync` command allows the `df -h` command to be somewhat truthful.
- However, `df` often shows incorrect free space. This inaccuracy issue is briefly discussed on the `btrfs` filesystem `df` page in the lesson titled “Implementing Btrfs File System.”
- It is not a bug. The main point is that `df` cannot be trusted on Btrfs volumes regardless of whether `sync` runs or not.
- Essentially, `df` knows nothing about how Btrfs duplicates and COWs data, so it is only accurate on a very small number of Btrfs volumes and rapidly becomes invalid.
- These practices use `df`, and it works, but these are small disks.
- For more about why you should not use `df` on a Btrfs volume, see:  
[https://btrfs.wiki.kernel.org/index.php/FAQ#Help.21\\_Btrfs\\_claims\\_1.27m\\_out\\_of\\_space.2C\\_but\\_it\\_looks\\_like\\_I\\_should\\_have\\_lots\\_left.21](https://btrfs.wiki.kernel.org/index.php/FAQ#Help.21_Btrfs_claims_1.27m_out_of_space.2C_but_it_looks_like_I_should_have_lots_left.21).

- I. Run the `btrfs file` command to display the available `btrfs` filesystem commands.

```
btrfs file
usage: btrfs filesystem [<group>] <command> [<args>]

btrfs filesystem df [options] <path>
 Show space usage information for a mount point
btrfs filesystem du [options] <path> [<path>..]
 Summarize disk usage of each file.
btrfs filesystem show [options] [<path>|<uid>|<device>|...
 Show the structure of a filesystem
...
```

- m. Use the `btrfs filesystem show` command to view the structure of the file system.

```
btrfs file show
Label: 'Btrfs' uuid: ...
Total devices 1 FS bytes used 6.46MiB
devid 1 size 20.00GiB used 2.02GiB path /dev/xvdc
```

- n. Use the `btrfs filesystem df` command to show the space used on the mount point.

```
btrfs file df /btrfs
Data, single: total=8.00MiB, used=6.34MiB
System, DUP: total=8.00MiB, used=16.00KiB
Metadata, DUP: total=1.00GiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

- Note that, by default, the metadata is duplicated even on a single-disk file system.

3. Create a single-disk Btrfs file system without duplicating the metadata.

- a. Use the `umount` command to unmount `/btrfs`.

```
umount /btrfs
```

- b. Use the `mkfs.btrfs -m single` command to make a Btrfs file system on `/dev/xvdc`.

```
mkfs.btrfs -f -m single -L Btrfs /dev/xvdc
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.

Label: Btrfs
UUID: ...
Node size: 16384
Sector size: 4096
Filesystem size: 20.00GiB
Block group profiles:
```

```

Data: single 8.00MiB
Metadata: single 8.00MiB
System: single 4.00MiB
SSD detected: yes
Incompat features: extref
Number of devices: 1
Devices:
ID SIZE PATH
1 20.00GiB /dev/xvdc

```

- c. Use the `mount` command to mount the file system.

```
mount -a
```

- d. Use the `cp` command to copy `/boot/vmlinuz-3.10*` to `/btrfs`.

```
cp /boot/vmlinuz-3.10* /btrfs
```

- e. Run the `sync` command, and then use the `df -h` command to display the space information for `/btrfs`.

```

sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 23M 20G 1% /btrfs

```

- f. Use the `btrfs filesystem df` command to show the space used on the mount point.

```

btrfs filesystem df /btrfs
Data, single: total=8.00MiB, used=6.15MiB
System, single: total=4.00MiB, used=16.00KiB
Metadata, single: total=264.00MiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B

```

- Note that the metadata is not duplicated.

4. Increase and decrease the size of a Btrfs file system.

- a. Use the `btrfs file resize` command to increase the size of the file system by 2 GB.

```

btrfs file resize +2G /btrfs
Resize '/btrfs' of '+2G'
ERROR: unable to resize '/btrfs': no enough free space

```

- The error indicates that you are unable to increase the size of the file system.
- The `resize` command does not change the size of the underlying partition.
- If you want to enlarge a file system, you must expand the partition first.

- b. Use the `btrfs file resize` command to reduce the size of the file system by 2 GB.

```

btrfs file resize -2G /btrfs
Resize '/btrfs/' of '-2G'

```

- Note that you are able to reduce the size of the file system.
- c. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 18G 23M 18G 1% /btrfs
```

- Note that the size is reduced by 2 GB, from 20 GB down to 18 GB.
- d. Use the `btrfs file resize` command to increase the size of the file system by 1 GB.

```
btrfs file resize +1G /btrfs
Resize '/btrfs/' of '+1G'
```

- Note that the resize was successful this time because space is available on the underlying partition.
- e. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 19G 23M 19G 1% /btrfs
```

- Note that the size is increased by 1 GB, from 18 GB up to 19 GB.
5. Add a disk to and remove a disk from a Btrfs file system.
- a. Use the `btrfs device add` command to add a 5 GB disk, `/dev/xvdd`, to the existing `/btrfs` file system.

```
btrfs device add /dev/xvdd /btrfs
/dev/xvdd appears to contain a partition table (dos).
Error: Use the -f option to force overwrite.
```

- b. Repeat the previous command but include the `-f` option.

```
btrfs device add -f /dev/xvdd /btrfs
```

- c. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 39G 23M 39G 1% /btrfs
```

- Note that the size is increased by 20 GB, from 19 GB up to 39 GB.
- d. Use the `btrfs file show` command to verify that the file system now contains two disks.

```
btrfs file show
Label: 'Btrfs' uuid: ...
 Total devices 2 FS bytes used 6.28MiB
 devid 1 size 19.00GiB used 276.00MiB path /dev/xvdc
 devid 2 size 20.00GiB used 0.00B path /dev/xvdd
```

- Note that all the data is on `/dev/xvdc` (276.00MiB) and that `/dev/xvdd` has 0.00.

- e. Run the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
blkid | grep btrfs
/dev/xvdc: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
/dev/xvdd: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
```

- Note that `/dev/xvdd` has the same label and UUID as `/dev/xvdc`.
- The two devices have different `UUID_SUB` values.

- f. Run the `btrfs filesystem balance` command on the file system.

```
btrfs file balance /btrfs
```

WARNING:

Full balance without filters requested. This operation is very intense and takes potentially very long. It is recommended to use the balance filters to narrow down the balanced data.

Use '`btrfs balance start --full-balance`' option to skip this warning. The operation will start in 10 seconds.

Use Ctrl-C to stop it.

10 9 8 7 6 5 4 3 2 1

Starting balance without any filters.

Done, had to relocate 4 out of 4 chunks

- After adding a device, it is recommended that you run this command on the file system to redistribute the chunks of the file system across all the devices.

- g. Use the `btrfs file show` command to show the effect of the balance command.

```
btrfs file show
```

Label: 'Btrfs' uuid: ...

Total devices 2 FS bytes used 6.53MiB

devid 1 size 19.00GiB used 32.00MiB path /dev/xvdc

devid 2 size 20.00GiB used 1.25GiB path /dev/xvdd

- Note that both devices now have some data on them.

- h. Use the `btrfs device delete` command to remove `/dev/xvdd` from the `/btrfs` file system.

```
btrfs device delete /dev/xvdd /btrfs
```

- i. Use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
df -h /btrfs
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvdc	19G	23M	19G	1%	/btrfs

- Note that the size is decreased by 20 GB, from 39 GB down to 19 GB.

6. Create a two-disk Btrfs file system with the default RAID level.

- a. Use the `umount` command to unmount `/btrfs`.

```
umount /btrfs
```

- b. Use the `mkfs.btrfs` command to make a Btrfs file system with two disks, `/dev/xvdc` and `/dev/xvdd`.

```
mkfs.btrfs -f -L Btrfs /dev/xvd[cd]
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.

Label: Btrfs
UUID: ...
Node size: 16384
Sector size: 4096
Filesystem size: 40.00GiB
Block group profiles:
 Data: RAID0 2.00GiB
 Metadata: RAID1 1.00GiB
 System: RAID1 8.00MiB
SSD detected: yes
Incompat features: extref
Number of devices: 2
Devices:
 ID SIZE PATH
 1 20.00GiB /dev/xvdc
 2 20.00GiB /dev/xvdd
```

- Note that the file system size is 40 GiB.
- c. Run the `blkid` command to display the block device attributes: UUID, label, and file system type.

```
blkid | grep btrfs
/dev/xvdc: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
/dev/xvdd: LABEL="Btrfs" UUID=... UUID_SUB=... TYPE="btrfs"
```

- Note that both devices are listed, each having the same `UUID` but different `UUID_SUB` values.
- d. Use the `mount` command to mount the file system.
- ```
# mount -a
```
- Note that the original `/etc/fstab` entry, which references only one device, `/dev/xvdc`, still works to mount the two-disk file system.
 - You need to reference only one device even when your file system contains multiple devices.
 - In this example, you can reference either device, `/dev/xvdc` or `/dev/xvdd`, when mounting the file system.
- e. Use the `cp` command to copy `/boot/vmlinuz-3.10*` to `/btrfs`.
- ```
cp /boot/vmlinuz-3.10* /btrfs
```

- f. Run the `sync` command, and then use the `df -h` command to display the space information for `/btrfs`.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 40G 24M 38G 1% /btrfs
```

- g. Use the `btrfs filesystem show` command to view the structure of the file system.

```
btrfs file show
Label: 'Btrfs' uuid: ...
 Total devices 2 FS bytes used 6.96MiB
 devid 1 size 20.00GiB used 2.01GiB path /dev/xvdc
 devid 2 size 20.00GiB used 2.01GiB path /dev/xvdd
```

- Note that the data is distributed evenly across the two devices.
- h. Use the `btrfs filesystem df` command to show the space used on the mount point.

```
btrfs file df /btrfs
Data, RAID0: total=2.00GiB, used=6.84MiB
System, RAID1: total=8.00MiB, used=16.00KiB
Metadata, RAID1: total=1.00GiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

- Note that, by default, the data is RAID0 and the metadata is RAID1.

7. Create a two-disk Btrfs file system with RAID-1 for both the data and metadata.

- a. Use the `umount` command to unmount `/btrfs`.

```
umount /btrfs
```

- b. Use the `mkfs.btrfs` command to make a Btrfs file system with two disks, `/dev/xvdc` and `/dev/xvdd`, and specify RAID-1 for data.
- Metadata is RAID1 by default.

```
mkfs.btrfs -f -L Btrfs -d raid1 /dev/xvd[cd]
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.
```

```
Label: Btrfs
UUID: ...
Node size: 16384
Sector size: 4096
Filesystem size: 40.00GiB
Block group profiles:
 Data: RAID1 1.00GiB
 Metadata: RAID1 1.00GiB
 System: RAID1 8.00MiB
SSD detected: yes
```

```
Incompat features: extref
Number of devices: 2
Devices:
 ID SIZE PATH
 1 20.00GiB /dev/xvdc
 2 20.00GiB /dev/xvdd
```

- c. Use the `mount` command to mount the file system.

```
mount -a
```

- d. Use the `cp` command to copy `/boot/vmlinuz-3.10*` to `/btrfs`.

```
cp /boot/vmlinuz-3.10* /btrfs
```

- e. Run the `sync` command and then use the `df -h` command to display the space information for `/btrfs`.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 23M 19G 1% /btrfs
```

- Note that the output shows that the file system has a size of 20 GB.
- This is accurate because this is a RAID-1 array (one disk mirrored to the other).

- f. Use the `btrfs` filesystem `df` command to show the space used on the mount point.

```
btrfs file df /btrfs
Data, RAID1: total=1.00GiB, used=6.59MiB
System, RAID1: total=8.00MiB, used=16.00KiB
Metadata, RAID1: total=1.00GiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

- Note that both the data and metadata are RAID1.
- You can also see that data is allocated in 1 GiB chunks.

## Practice 11-2: Working with Subvolumes and Snapshots

### Overview

In this practice, you:

- Create a Btrfs subvolume
- Create a snapshot of the subvolume
- List the subvolume and snapshot
- Mount the subvolume and snapshot
- Delete the snapshot
- Take a snapshot of a file by using the `cp --reflink` command

### Assumptions

- You are the `root` user on the **host03** VM.

### Tasks

#### 1. Create a Btrfs subvolume.

- a. Run the `btrfs sub` command to display the available `btrfs subvolume` commands.

```
btrfs sub
usage: btrfs subvolume <command> <args>

 btrfs subvolume create [-i <qgroupid>] [<dest>/]<name>
 Create a subvolume
 btrfs subvolume delete [options] <subvolume>
[<subvolume>...]
 Delete subvolume(s)
 btrfs subvolume list [options] [-G [+|-]value] [-C [+|-]
]value] [--sort=gen,ogen,rootid,path] <path>
 List subvolumes (and snapshots)
 btrfs subvolume snapshot [-r] [-i <qgroupid>] <source>
<dest>| [<dest>/]<name>
 Create a snapshot of the subvolume
 btrfs subvolume get-default <path>
 Get the default subvolume of a filesystem
 btrfs subvolume set-default <subvol-id> <path>
 Set the default subvolume of a filesystem
 btrfs subvolume find-new <path> <lastgen>
 List the recently modified files in a filesystem
 btrfs subvolume show <subvol-path>
 Show more information of the subvolume
 btrfs subvolume sync <path> [<subvol-id>...]
```

Wait until given subvolume(s) are completely removed from the filesystem.

manage subvolumes: create, delete, list, etc

- Use the `btrfs sub create` command to create a subvolume named `SV1`.

```
btrfs sub create /btrfs/SV1
Create subvolume '/btrfs/SV1'
```

- Use the `ls -l` command to display the contents of `/btrfs`.

```
ls -l /btrfs
drwxr-xr-x ... SV1
-rwxr-xr-x ... vmlinuz-3.10...
```

- Note that the `SV1` subvolume is created and appears as a regular directory.

- Use the `mv` command to move the `vmlinuz*` file to the `/btrfs/SV1` subvolume.

```
mv /btrfs/vmlinuz* /btrfs/SV1
```

- Use the `ls -lR` command to display the contents of `/btrfs`.

```
ls -lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-3.10...
```

- Note that the `vmlinuz*` file now resides in the `SV1` subvolume.

- Run the `sync` command, and then use the `df -h` command to display the disk space usage on the `/btrfs` file system.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 23M 19G 1% /btrfs
```

- Create a Btrfs snapshot.

- Use the `btrfs sub snapshot` command to create a snapshot of the `SV1` subvolume. Name the snapshot `SV1-snap`.

```
btrfs sub snapshot /btrfs/SV1 /btrfs/SV1-snap
Create a snapshot of '/btrfs/SV1' in '/btrfs/SV1-snap'
```

- Use the `ls -lR` command to display the contents of `/btrfs`.

```
ls -lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1
drwxr-xr-x ... SV1-snap
```

```
/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-3.10...

/btrfs/SV1-snap:
-rwxr-xr-x ... vmlinuz-3.10...
```

- Note that the new SV1-snap directory in /btrfs is created.
  - The SV1-snap snapshot is a point-in-time copy of the SV1 subvolume.
- c. Run the sync command, and then use the df -h command to display disk space usage on the /btrfs file system.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 23M 19G 1% /btrfs
```

- Note that the creation of the snapshot did not reduce the available disk space.
- d. Use the cp command to copy /boot/vmlinuz-4.1.12\* to the /btrfs/SV1 subvolume.

```
cp /boot/vmlinuz-4.1.12* /btrfs/SV1
```

- e. Use the ls -lR command to display the contents of /btrfs.

```
ls -lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1
drwxr-xr-x ... SV1-snap

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-3.10...
-rwxr-xr-x ... vmlinuz-4.1.12...

/btrfs/SV1-snap:
-rwxr-xr-x ... vmlinuz-3.10...
```

- Note that the SV1 subvolume now contains two vmlinuz\* files.
3. Mount the subvolume and the snapshot.
- a. Use the btrfs sub list command to list the subvolumes and snapshots.
- The output is sample only. The ID and gen numbers may be different.

```
btrfs sub list /btrfs
ID 258 gen 13 top level 5 path SV1
ID 259 gen 11 top level 5 path SV1-snap
```

- In this example, the ID of the SV1 subvolume is 258.
- The ID of the SV1-snap subvolume (snapshot) is 259.
- The IDs of your subvolumes might be different. Substitute your IDs as necessary in the following steps.

- b. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
btrfs sub get-default /btrfs
ID 5 (FS_TREE)
```

- Note that the ID of 5 indicates that the `root` subvolume is the default.

- c. Use the `btrfs sub set-default` command to set the subvolume ID to 258 (ID of the `SV1` subvolume).

```
btrfs sub set-default 258 /btrfs
```

- d. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
btrfs sub get-default /btrfs
ID 258 gen 13 top level 5 path SV1
```

- Note that the default subvolume ID is 258 for the `SV1` subvolume.

- e. Use the `umount` command to unmount the file system mounted on `/btrfs`.

```
umount /btrfs
```

- f. Use the `mount` command to mount the file system.

```
mount -a
```

- g. Use the `ls -lR` command to display the contents of `/btrfs`.

```
ls -lR /btrfs
/btrfs/:
-rwxr-xr-x ... vmlinuz-3.10...
-rwxr-xr-x ... vmlinuz-4.1.12...
```

- Note that the contents of the `SV1` subvolume are now mounted on `/btrfs`.

- h. Use the `btrfs sub set-default` command to set the subvolume ID to 259 (ID of the `SV1-snap` snapshot).

```
btrfs sub set-default 259 /btrfs
```

- i. Use the `btrfs sub get-default` command to get the default subvolume ID of the file system.

```
btrfs sub get-default /btrfs
ID 259 gen 11 top level 5 path SV1-snap
```

- Note that the default subvolume ID is 259 for the `SV1-snap` subvolume.

- j. Use the `umount` command to unmount the file system mounted on `/btrfs`.

```
umount /btrfs
```

- k. Use the `mount` command to mount the file system.

```
mount -a
```

- l. Use the `ls -lR` command to display the contents of `/btrfs`.

```
ls -lR /btrfs
/btrfs/:
```

```
...
-rwxr-xr-x ... vmlinuz-3.10...
```

- Note that the contents of the SV1-snap snapshot are now mounted on /btrfs.

4. Mount the root subvolume.

- Use the btrfs sub set-default command to set the subvolume ID to 5 (ID of the root subvolume).

```
btrfs sub set-default 5 /btrfs
```

- Use the btrfs sub get-default command to get the default subvolume ID of the file system.

```
btrfs sub get-default /btrfs
ID 5 (FS_TREE)
```

- Note that the default subvolume ID is 5 for the root subvolume.

- Use the umount command to unmount the file system mounted on /btrfs.

```
umount /btrfs
```

- Use the mount command to mount the file system.

```
mount -a
```

- Use the ls -lR command to display the contents of /btrfs.

```
ls -lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1
drwxr-xr-x ... SV1-snap

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-3.10...
-rwxr-xr-x ... vmlinuz-4.1.12...

/btrfs/SV1-snap:
-rwxr-xr-x ... vmlinuz-3.10...
```

- Note that the root subvolume is now mounted on /btrfs.
- You could also mount the root file system as follows, but this does not change the default subvolume ID.
- **DO NOT** run this command, this is information only.

```
mount -o subvolid=5 /dev/xvdc /btrfs
```

- ID of 0 can also be used to mount the root subvolume.

5. Delete the snapshot.

- Use the btrfs sub delete command to delete the snapshot.

```
btrfs sub delete /btrfs/SV1-snap
Delete subvolume (no-commit): '/btrfs/SV1-snap'
```

- b. Use the `btrfs sub list` command to list the subvolumes and snapshots.

- Rerun this command if an error is encountered.

```
btrfs sub list /btrfs
ID 258 gen 13 top level 5 path SV1
```

- Note that the `SV1-snap` snapshot is removed.

- c. Use the `ls -lR` command to display the contents of `/btrfs`.

```
ls -lR /btrfs
/btrfs/:
drwxr-xr-x ... SV1

/btrfs/SV1:
-rwxr-xr-x ... vmlinuz-3.10...
-rwxr-xr-x ... vmlinuz-4.1.12...
```

- Note that the `SV1-snap` directory is removed.

6. Take a snapshot of a file.

- a. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 29M 19G 1% /btrfs
```

- b. Use the `cd` command to change to the `/btrfs/SV1` directory.

```
cd /btrfs/SV1
```

- c. Use the `cp` command to copy the `vmlinuz-3.10*` file to `copy_of_10`.

```
cp vmlinuz-3.10* copy_of_10
```

- d. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 35M 19G 1% /btrfs
```

- Note that the “Used” amount increased from 29M to 35M.

- e. Use the `rm` command to remove the `copy_of_10` file.

```
rm copy_of_10
rm: remove regular file 'copy_of_10'? y
```

- f. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
sync
df -h /btrfs
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvdc	20G	29M	19G	1%	/btrfs

- Note that the “Used” amount decreased from 35M to 29M.

- g. Use the `cp --reflink` command to copy the `vmlinuz-3.10*` file to `copy_of_10`.

```
cp --reflink vmlinuz-3.10* copy_of_10
```

- h. Use the `ls -l` command to display the sizes of the original file and the copy.

```
ls -l
-rwxr-xr-x ... 6383280 ... copy_of_10
-rwxr-xr-x ... 6383280 ... vmlinuz-3.10...
-rwxr-xr-x ... 5993952 ... vmlinuz-4.1.12...
```

- Note that the sizes of the `vmlinuz-3.10*` and `copy_of_10` files are the same.

- i. Use the `diff` command to compare the original file and the copy.

```
diff vmlinuz-3.10* copy_of_10
```

- No output indicates that there are no differences in the two files.

- j. Run the `sync` command, and then use the `df -h` command to display disk space usage on the `/btrfs` file system.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 29M 19G 1% /btrfs
```

- Note that the “Used” amount remained at 29M even after making a copy.
- The data blocks are not duplicated when the `--reflink` option is used.
- This allows the copy to be almost instantaneous and also saves disk space.
- One restriction is that this operation works only within the boundaries of the same file system and within the same subvolume.

## Practice 11-3: Recovering from Data Corruption

---

### Overview

In this practice, you:

- Prepare the environment by re-creating a RAID-1 file system
- Use the `btrfs-corrupt-block` utility to induce data corruption
- Use the `btrfs scrub` command to recover from data corruption

### Assumptions

- You are the `root` user on the **host03** VM.

### Tasks

1. Prepare the environment.
  - a. From **host03**, use the `cd` command to change to the `root` user home directory, and then use the `umount` command to unmount `/btrfs`.

```
cd
umount /btrfs
```

- b. Use the `mkfs.btrfs` command to make a Btrfs file system with two disks, `/dev/xvdc` and `/dev/xvdd`, and specify RAID-1 for data.

```
mkfs.btrfs -f -L Btrfs -d raid1 /dev/xvd[cd]
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.
```

```
Label: Btrfs
UUID: ...
Node size: 16384
Sector size: 4096
Filesystem size: 40.00GiB
Block group profiles:
 Data: RAID1 1.00GiB
 Metadata: RAID1 1.00GiB
 System: RAID1 8.00MiB
SSD detected: yes
Incompat features: extref
Number of devices: 2
Devices:
 ID SIZE PATH
 1 20.00GiB /dev/xvdc
 2 20.00GiB /dev/xvdd
```

- c. Use the `mount` command to mount the file system.

```
mount -a
```

- d. Use the `cp` command to copy `/boot/vmlinuz-3.10*` to `/btrfs`.

```
cp /boot/vmlinuz-3.10* /btrfs
```

- e. Run the `sync` command, and then use the `df -h` command to display space information for `/btrfs`.

```
sync
df -h /btrfs
Filesystem Size Used Avail Use% Mounted on
/dev/xvdc 20G 23M 19G 1% /btrfs
```

- f. Use the `btrfs filesystem show` command to view the structure of the file system.

```
btrfs filesystem show
Label: 'Btrfs' uuid: ...
 Total devices 2 FS bytes used 6.71MiB
 devid 1 size 20.00GiB used 2.01GiB path /dev/xvdc
 devid 2 size 20.00GiB used 2.01GiB path /dev/xvdd
```

## 2. Induce data corruption on the Btrfs file system.

- The `btrfs-corrupt-block` command, which is in the `/stage` directory on `gateway`, is used in this practice.
- a. Use the `scp` command to copy `btrfs-corrupt-block` from the `/stage` directory on `gateway` to the `root` user's home directory on `host03`.

```
scp gateway:/stage/btrfs-corrupt-block ~
root@gateway's password:
btrfs-corrupt-block 100% 991KB
10.3MB/s 00:00
```

- b. Use the `cd` command to change to the `/btrfs` directory, and then run the `filefrag` command to report on file fragmentation on the `vmlinuz-3.10*` file.

```
cd /btrfs
filefrag -v vmlinuz-3.10*
Filesystem type is: 9123683e
File size of vmlinuz-3.10.0-862.el7.x86_64 is 6383280 (1559
blocks of 4096 bytes)
ext: logical_offset: physical_offset: length:
expected: flags:
 0: 0.. 1558: 269440.. 270998: 1559:
eof
vmlinuz-3.10.0-862.el7.x86_64: 1 extent found
```

- In this example, the file is on physical block 269440 and has a 4K (4096) block size.
- c. Use a calculator to multiply the physical block by the block size.
  - Multiply  $4096 * 269440 = 1103626240$ .
  - In this example, 1103626240 is the amount that you want to corrupt.

- d. Use the `cd` command to change to your home directory, and then use the `umount` command to unmount `/btrfs`.

```
cd
umount /btrfs
```

- e. Use the `btrfs-corrupt-block` tool to corrupt the first copy of that block.

- This is a very dangerous tool and is not part of the `btrfs-progs` RPM.
- Do not use this tool outside the parameters of this lesson.
- By default, this tool corrupts all the copies of the block.
- You want to corrupt only the first copy of the block.

```
./btrfs-corrupt-block -c 1 -1 1103626240 /dev/xvdc
mirror 1 logical 1103626240 physical 1083703296 device /dev/xvdd
corrupting 1103626240 copy 1
mirror 2 logical 1103626240 physical 1103626240 device /dev/xvdc
```

3. Mount the file system and scrub.

- a. Use the `mount` command to mount the file system.

```
mount -a
```

- b. Use the `btrfs scrub` command to start a file system scrub. If necessary, press the **Enter** key after the WARNING message, to get the prompt back.

```
btrfs scrub start /btrfs
scrub started on /btrfs/, fsid ... (pid=...)
WARNING: errors detected during scrubbing, corrected
```

- Press ENTER to get a prompt to enter the next command.

- c. Use the `btrfs scrub` command to get the status of the file system scrub.

```
btrfs scrub status /btrfs
scrub status for ...
scrub started at ... and finished after 00:00:00
total bytes scrubbed: 13.43MiB with 1 errors
error details: csum=1
corrected errors: 1, uncorrectable errors: 0, unverified
errors: 0
```

- Note that one error was found, a checksum error, and one error was corrected.

- d. Use the `dmesg` command to determine what happened.

- Sample output is shown.

```
dmesg
...
[...] BTRFS: checksum error at logical 1103626240 on dev
/dev/xvdd, sector 2116608, root 5, inode 257, offset 0, length
4096, links 1 (path: vmlinuz-3.10.0-862.el7.x86_64)
[...] BTRFS: bdev /dev/xvdd errs: wr 0, rd 0, flush 0, corrupt
1, gen 0
```

```
[...] BTRFS: fixed up error at logical 1103626240 on dev
/dev/xvdd
```

- Note that the error induced by the btrfs-corrupt-block command at logical 1103626240 was corrected by the scrub.

- e. Use the btrfs scrub command to start a file system scrub.

```
btrfs scrub start /btrfs
scrub started on /btrfs/, fsid ... (pid=...)
```

- f. Use the btrfs scrub command to get the status of the file system scrub.

```
btrfs scrub status /btrfs
Scrub status for ...
scrub started at ... and finished after 00:00:00
total bytes scrubbed: 13.43MiB with 0 errors
```

- Note that 0 errors were found.

4. Prepare for future practices.

- a. Remove the file in the /btrfs directory.

```
rm /btrfs/*
rm: remove regular file '/btrfs/vmlinuz-3.10.0-862.el7.x86_64'? y
```

- b. Use the umount command to unmount /btrfs.

```
umount /btrfs
```

- c. Remove the /btrfs directory.

```
rmdir /btrfs
```

- d. Use the vi editor to remove the following entry in /etc/fstab:

```
vi /etc/fstab
...
/dev/xvde /btrfs btrfs defaults 0 0 (delete this entry)
```

- e. Remove the btrfs-corrupt-block file from the root user's home directory.

```
cd
rm btrfs-corrupt-block
rm: remove regular file 'btrfs-corrupt-block'? y
```

- f. Use the exit command to log off host03.

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

## **Practices for Lesson 12: Storage Administration**

## **Practices for Lesson 12: Overview**

---

### **Practices Overview**

In these practices, you create and work with logical volumes and RAID devices.

## Practice 12-1: Creating Linux LVM Partitions

---

### Overview

In this practice, you create new partitions and change the system ID to Linux LVM.

### Assumptions

- You are the `root` user on **gateway**.
- There are no partitions on `/dev/xvdc` and `/dev/xvdd` on **host03**.
- There are no file systems mounted on `/dev/xvdc` and `/dev/xvdd` on **host03**.

### Tasks

1. Log in to **host03**.

Use the `ssh` command to log in to **host03** as the `root` user.

Provide the `root` password when prompted.

```
ssh host03
root@host03's password:
Last login: ...
```

2. Create two 1G partitions on `/dev/xvdc`.

- a. Use the `fdisk` command to create two 1G primary partitions on `/dev/xvdc`.

```
fdisk /dev/xvdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x8e7a9de5.

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 (2048-41943039, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default
41943039): +1G
Partition 1 of type Linux and of size 1 GiB is set
```

```

Command (m for help): n
Partition type:
 p primary (1 primary, 0 extended, 3 free)
 e extended
Select (default p): ENTER
Using default response p
Partition number (2-4, default 2): ENTER
First sector (2099200-41943039, default 2099200): ENTER
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-41943039, default
41943039): +1G
Partition 2 of type Linux and of size 1 GiB is set

```

- b. Use the “t” command to change the system ID on partition 2.

```

Command (m for help): t
Partition number (1,2, default 2): ENTER
Hex code (type L to list codes): L
 0 Empty 24 NEC DOS 81 Minix / old Lin ...
 1 FAT12 27 Hidden NTFS Win 82 Linux swap / So ...
 2 XENIX root 39 Plan 9 83 Linux ...
...
 8 AIX 4e QNX4.x 2nd part 8e Linux LVM ...
...
Hex code (type L to list codes): 8e
Changed system type of partition 'Linux' 'Linux LVM'

```

- c. Print the new partition table.

```

Command (m for help): p
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
...
 Device Boot Start End Blocks Id System
/dev/xvdc1 2048 2099199 1048576 83 Linux
/dev/xvdc2 2099200 4196351 1048576 8e Linux LVM

```

- d. Save the new partition table.

```

Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.

```

## 3. Create two 1G partitions on /dev/xvdd.

- a. Use the fdisk command to create two 1G primary partitions on /dev/xvdd.

```
fdisk /dev/xvdd
Welcome to fdisk (util-linux 2.23.2).
...
Command (m for help): n
Partition type:
 p primary partition (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 (2048-41943039, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default
41943039): +1G
Partition 1 of type Linux and of size 1 GiB is set

Command (m for help): n
Partition type:
 p primary partition (1 primary, 0 extended, 3 free)
 e extended
Select (default p): ENTER
Using default response p
Partition number (2-4, default 2): ENTER
First sector (2099200-41943039, default 2099200): ENTER
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-41943039, default
41943039): +1G
Partition 2 of type Linux and of size 1 GiB is set
```

- b. Use the "t" command to change the system ID on partition 2.

```
Command (m for help): t
Partition number (1,2, default 2): ENTER
Hex code (type L to list codes): 8e
Changed system type of partition 'Linux' 'Linux LVM'
```

- c. Print the new partition table.

```
Command (m for help): p
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
...
 Device Boot Start End Blocks Id System
/dev/xvdd1 2048 2099199 1048576 83 Linux
/dev/xvdd2 2099200 4196351 1048576 8e Linux LVM
```

- d. Save the new partition table.

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

## Practice 12-2: Creating a Logical Volume

---

### Overview

In this practice, you create physical volumes, a volume group, and a logical volume. You also use LVM utilities to display information about the LVM entities.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Initialize the new partitions for use by LVM (create physical volumes).

- a. List the partitions with the Linux LVM (**8e**) system ID.

```
fdisk -l | grep 8e
...
/dev/xvdc2 2099200 4196351 1048576 8e Linux LVM
/dev/xvdd2 2099200 4196351 1048576 8e Linux LVM
```

- b. Use the `pvccreate` command to create physical volumes on both partitions.

```
pvccreate -v /dev/xvdc2 /dev/xvdd2
Wiping internal VG cache
Wiping cache of LVM-capable devices
Wiping signatures on new PV /dev/xvdc2.
Wiping signatures on new PV /dev/xvdd2.
Set up physical volume for "/dev/xvdc2" with 2097152
available sectors.
Zeroing start of device /dev/xvdc2.
Writing physical volume data to disk "/dev/xvdc2".
Physical volume "/dev/xvdc2" successfully created.
Set up physical volume for "/dev/xvdd2" with 2097152
available sectors.
Zeroing start of device /dev/xvdd2.
Writing physical volume data to disk "/dev/xvdd2".
Physical volume "/dev/xvdd2" successfully created.
```

2. Display information about the physical volumes.

- a. Use the `pvdisplay` command to display attributes of the physical volumes.

```
pvdisplay
...
"/dev/xvdd2" is a new physical volume of "1.00 GiB"
--- NEW Physical volume ---
PV Name /dev/xvdd2
VG Name
PV Size 1.00 GiB
```

```
...
"/dev/xvdc2" is a new physical volume of "1.00 GiB"
--- NEW Physical volume ---
PV Name /dev/xvdc2
VG Name
PV Size 1.00 GiB
...
```

- b. Use the `pvs` command to report information in a more condensed form.

```
pvs
PV VG Fmt Attr PSize PFree
...
/dev/xvdc2 lvm2 --- 1.00g 1.00g
/dev/xvdd2 lvm2 --- 1.00g 1.00g
```

- c. Use the `pvscan` command to scan all disks for physical volumes.

```
pvscan
...
PV /dev/xvdd2 lvm2 [1.00 GiB]
PV /dev/xvdc2 lvm2 [1.00 GiB]
Total: ...
```

### 3. Create a volume group.

- Use the `vgcreate` command to create a volume group named `myvolg` from the `/dev/xvdc2` physical volume.

```
vgcreate -v myvolg /dev/xvdc2
Wiping internal VG cache
Wiping cache of LVM-capable devices
Wiping signatures on new PV /dev/xvdc2.
Adding physical volume '/dev/xvdc2' to volume group 'myvolg'
Archiving volume group "myvolg" metadata (seqno 0).
Creating volume group backup "/etc/lvm/backup/myvolg" (seqno 1).
Volume group "myvolg" successfully created
```

### 4. Display information about the volume group.

- a. Use the `vgdisplay` command to display attributes of the volume group.

```
vgdisplay
--- Volume group ---
VG Name myvolg
System ID
Format lvm2
Metadata Areas 1
Metadata Sequence No 1
VG Access read/write
```

```

VG Status resizable
MAX LV 0
Cur LV 0
Open LV 0
Max PV 0
Cur PV 1
Act PV 1
VG Size 1020.00 MiB
PE Size 4.00 MiB
Total PE 255
Alloc PE / Size 0 / 0
Free PE / Size 255 / 1020.00 MiB
VG UUID ...

```

- b. Use the `vgs` command to report information in a more condensed form.

```

vgs
 VG #PV #LV #SN Attr VSize VFree
myvolg 1 0 0 wz--n- 1020.00m 1020.00m
...

```

- c. Use the `vgscan` command to scan all disks for volume groups and rebuild caches.

```

vgscan
 Reading volume groups from cache.
 Found volume group "myvolg" using metadata type lvm2
...

```

- d. Display information about the physical volumes.

```

pvs
 PV VG Fmt Attr PSize PFree
...
 /dev/xvdc2 myvolg lvm2 a-- 1020.00m 1020.00m
 /dev/xvdd2 lvm2 --- 1.00g 1.00g

```

- Note that the `/dev/xvdc2` physical volume is assigned to the `myvolg` volume group.

## 5. Create a logical volume.

- Use the `lvcreate` command to create a 500 MB logical volume named `myvol` from the `myvolg` volume group.

```

lvcreate -v -L 500m -n myvol myvolg
 Archiving volume group "myvolg" metadata (seqno 1).
 Creating logical volume myvol
 Creating volume group backup "/etc/lvm/backup/myvolg" (seqno
2).
 Activating logical volume myvolg/myvol.

```

```

activation/volume_list configuration setting not defined:
Checking only host tags for myvolg/myvol.

Creating myvolg-myvol
Loading table for myvolg-myvol (249:2).
Resuming myvolg-myvol (249:2).
Wiping known signatures on logical volume "myvolg/myvol"
Initializing 4.00 KiB of logical volume "myvolg/myvol" with
value 0.
Logical volume "myvol" created.

```

6. Display information about the logical volume.

- a. Use the `lvdisplay` command to display attributes of the logical volume.

```

lvdisplay
--- Logical volume ---
LV Path /dev/myvolg/myvol
LV Name myvol
VG Name myvolg
LV UUID ...
LV Write Access read/write
LV Creation host, time host03.example.com, ...
LV Status available
open 0
LV Size 500.00 MiB
Current LE 125
Segments 1
Allocation inherit
Read ahead sectors auto
 - currently set to 8192
Block device 249:2
...

```

- b. Use the `lvs` command to report information in a more condensed form.

```

lvs
LV VG Attr LSize Pool Origin Data% Move Log...
myvol myvolg -wi-a---- 500.00m
...

```

- c. Use the `lvscan` command to scan all disks for logical volumes.

```

lvscan
ACTIVE '/dev/myvolg/myvol' [500.00 MiB] inherit
...

```

- d. Display information about the physical volumes.

```

pvs
PV VG Fmt Attr PSize PFree

```

```
...
/dev/xvdc2 myvolg lvm2 a-- 1020.00m 520.00m
/dev/xvdd2 lvm2 --- 1.00g 1.00g
```

- Note that the free space in the /dev/xvdc2 physical volume has been reduced.
- e. Display information about the volume group.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 1 1 0 wz--n- 1020.00m 520.00m
...
```

- Note that the free space in the myvolg volume group has also been reduced.

## Practice 12-3: Creating a File System and Mounting a Logical Volume

### Overview

In this practice, you create a file system on the logical volume and mount the logical volume.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Display the block device name that was automatically created.

- a. List the `/dev` entries for the **myvol** logical volume.

```
ls -l /dev/myvolg/myvol
lrwxrwxrwx. ... /dev/myvolg/myvol -> ../dm-2
ls -l /dev/mapper/myvolg-myvol
lrwxrwxrwx. ... /dev/mapper/myvolg-myvol -> ../dm-2
```

- Note that two entries were automatically created.
- Note that both entries are symbolic links to `/dev/dm-2`.

- b. List the `/dev/dm-2` entry.

```
ls -l /dev/dm-2
brw-rw----. 1 root disk 249, 2 ... /dev/dm-2
```

- Note that `/dev/dm-2` is a block device.

2. Create a file system on the logical volume.

- a. Create an **ext4** file system on the **myvol** logical volume.

```
mkfs.ext4 /dev/mapper/myvolg-myvol
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
128016 inodes, 512000 blocks
25600 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=34078720
63 block groups
8192 blocks per group, 8192 fragments per group
2032 inodes per group
Superblock backups stored on blocks:
 8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

- b. Create a /myvol mount point.

```
mkdir /myvol
```

- c. Mount the file system.

```
mount /dev/mapper/myvolg-myvol /myvol
```

- d. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-myvol 477M 2.3M 445M 1% /myvol
```

3. Update the file systems mount table.

- a. Use the vi editor to add the following line to /etc/fstab:

```
vi /etc/fstab
/dev/mapper/myvolg-myvol /myvol ext4 defaults 0 0
```

## Practice 12-4: Backing Up Volume Group Metadata

---

### Overview

In this practice, you view the LVM configuration file, view the automatically created metadata backups and archives, and manually create a volume group metadata backup.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. View the configuration of metadata backups and archiving.
  - a. Use the `less` command to view the logical volume configuration, `/etc/lvm/lvm.conf`.

```
less /etc/lvm/lvm.conf
This is an example configuration file for the LVM2 system.
...
```

- b. Use the search character, `/`, and search for the word `backup`.

```
/backup
```

- The search results in displaying the following:

```
Configuration section backup.
How LVM metadata is backed up and archived.
In LVM, a 'backup' is a copy of the metadata for the current system,
and an 'archive' contains old metadata configurations. They are
stored in a human readable text format.
backup {

 # Configuration option backup/backup.
 # Maintain a backup of the current metadata configuration.
 # Think very hard before turning this off!
 backup = 1

 # Configuration option backup/backup_dir.
 # Location of the metadata backup files.
 # Remember to back up this directory regularly!
 backup_dir = "/etc/lvm/backup"

 # Configuration option backup/archive.
 # Maintain an archive of old metadata configurations.
 # Think very hard before turning this off.
 archive = 1

 # Configuration option backup/archive_dir.
 # Location of the metadata archive files.
 # Remember to back up this directory regularly!
}
```

- Note that metadata backups are enabled (`backup = 1`) and backups are stored in the `/etc/lvm/backup` directory.

- Also note that archives are enabled (`archive = 1`) and the archives are stored in the `/etc/lvm/archive` directory.
- Press `q` to quit the `less` command.

```
q
#
```

- You can also use the `lvm dumpconfig` command to view the configuration settings.

```
lvm dumpconfig
config {
 checks=1
 abort_on_errors=0
 profile_dir="/etc/lvm/profile"
}
...
backup {
 backup=1
 backup_dir="/etc/lvm/backup"
 archive=1
 archive_dir="/etc/lvm/archive"
 retain_min=10
 retain_days=30
}
...
```

## 2. View the metadata backup and archive files.

- Use the `cd` command to change to the `/etc/lvm/backup` directory. Use the `ls` command to display the contents of the directory.

```
cd /etc/lvm/backup
ls
myvolg ...
```

- Note that the backup of the `myvolg` volume group was created automatically.
- Use the `less` command to view the contents of the `myvolg` metadata backup file.

```
less myvolg
...
description = "Created *after* executing 'lvcreate -v -L 500...
...
myvolg {
...
 physical_volumes {
 pv0 {
...
 device = "/dev/xvdc2" # Hint only
...
...
```

```

logical_volumes {
 myvol {
 ...

```

- Note that the description states the backup file was created “after” executing the `lvcreate` command string.
  - Also note that the metadata backup file includes information on the physical and logical volumes.
- c. Press `q` to quit the `less` command.
  - d. `q`
  - e. `#`
  - f. Use the `cd` command to change to the `/etc/lvm/archive` directory. Use the `ls` command to display the contents of the directory.

```

cd /etc/lvm/archive
ls
myvolg_00000-...vg myvolg_00001-...vg

```

- Note that the archive files for the `myvolg` volume group were created automatically.
- g. Use the `less` command to view the contents of the `myvolg-00000*` archive file.

```

less myvolg_00000*
...
description = "Created *before* executing 'vgcreate -v myvolg...
...
myvolg {
...
 physical_volumes {
 pv0 {
...
 device = "/dev/xvdc2" # Hint only
...

```

- Note that the description states the archive file was created “before” executing the `vgcreate` command string.
- h. Press `q` to quit the `less` command.
  - i. `q`
  - j. `#`
  - k. Use the `less` command to view the contents of the `myvolg-00001*` archive file.

```

less myvolg_00001*
...
description = "Created *before* executing 'lvcreate -v -L 500...
...
myvolg {
...
 physical_volumes {

```

```
pv0 {
...
 device = "/dev/xvdc2" # Hint only
...}
```

- Note that the description states the archive file was created “before” executing the `lvcreate` command string.
  - I. Press `q` to quit the `less` command.
  - m. `q`
  - n. `#`
- 3. Create a metadata backup of the `myvolg` volume group.
  - a. Use the `vgcfgbackup` command to back up the metadata for the `myvolg` volume group.
    - Include the `-f file_today` argument to the `vgcfgbackup` command.
  - b. Use the `diff` command to display the differences in the newly created backup file and the existing `myvolg` metadata backup file.
    - Note that the newly created backup was created in the current directory.

```
vgcfgbackup -f file_today myvolg
Volume group "myvolg" successfully backed up.
```

```
diff file_today /etc/lvm/backup/myvolg
...
< description = "vgcfgbackup -f file_today myvolg"

> description = "Created *after* executing 'lvcreate -v -L ...
..."
```

- Note that only the description and the creation time information are different.

## Practice 12-5: Creating a Logical Volume Snapshot

---

### Overview

In this practice, you create a snapshot volume, mount the snapshot, and remove the snapshot volume.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Create a snapshot volume.

- a. Copy `/boot/init*` to the mounted logical volume, `/myvol`.

```
cp /boot/init* /myvol
ls /myvol
initramfs-0-rescue-...img
initramfs-3.10.0-862.el7.x86_64.img
initramfs-4.1.12-112.16.4.el7uek.x86_64.img
...
```

- b. Create a 500 MB snapshot named `myvol-snapshot` of the **myvol** logical volume.

```
lvcreate -L 500m -s -n myvol-snapshot myvolvg/myvol
Logical volume "myvol-snapshot" created.
```

- c. Use the `ls -l` command to list the contents of the `/etc/lvm/backup` directory and the contents of the `/etc/lvm/archive` directory.

```
ls -l /etc/lvm/backup
-rw-----. . . <date_time> myvolg
...
ls -l /etc/lvm/archive
-rw-----. . . <date_time> file_today
-rw-----. . . <date_time> myvolg_00000-...vg
-rw-----. . . <date_time> myvolg_00001-...vg
-rw-----. . . <date_time> myvolg_00002-...vg
...
```

- Note that a new `myvolg` backup file was automatically created when the snapshot was created (note the time stamp).
- Also note that a new archive file was automatically created (note the time stamp on the `myvolg_00002*` file).

d. List the logical volumes.

```
lvs
 LV VG Attr LSize Pool Origin Data% ...
 myvol myvolg owi-aos--- 500.00m
 myvol-snapshot myvolg swi-a-s--- 500.00m myvol 0.01
```

- Note that the new myvol-snapshot logical volume exists.

e. List the contents of the /dev/myvolg and /dev/mapper directories. Your output may differ.

```
ls -l /dev/myvolg
lrwxrwxrwx. ... myvol -> ../dm-2
lrwxrwxrwx. ... myvol-snapshot -> ../dm-5
ls -l /dev/mapper
crw-rw----. ... control
lrwxrwxrwx. ... myvolg-myvol -> ../dm-2
lrwxrwxrwx. ... myvolg-myvol-real -> ../dm-3
lrwxrwxrwx. ... myvolg-myvol--snapshot -> ../dm-5
lrwxrwxrwx. ... myvolg-myvol--snapshot-cow -> ../dm-4
...
```

2. Mount the snapshot.

a. Mount the snapshot onto /mnt.

```
mount -t ext4 /dev/myvolg/myvol-snapshot /mnt
```

b. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-myvol
 477M 196M 252M 44% /myvol
/dev/mapper/myvolg-myvol--snapshot
 477M 196M 252M 44% /mnt
```

c. List the files on /mnt.

```
ls /mnt
initramfs-0-rescue-...img
initramfs-3.10.0-862.el7.x86_64.img
initramfs-4.1.12-112.16.4.el7uek.x86_64.img
...
```

- Note that these are the same files that were copied onto /myvol.

d. Remove the files on /myvol.

- Answer **y** when prompted to remove each file.

```
rm /myvol/init*
rm: remove regular file '/myvol/initramfs-0-rescue...'? y
```

```
rm: remove regular file '/myvol/initramfs-3.10.0-862...'? y
...

```

- e. List the files on /mnt.

```
ls /mnt
initramfs-0-rescue-...img
initramfs-3.10.0-693.el7.x86_64.img
initramfs-4.1.12-94.3.9.el7uek.x86_64.img
...
```

- Note that these files are still present.
- Removing files from the original volume does not change the snapshot's content.

3. Remove the snapshot.

- a. Unmount the snapshot from /mnt.

```
umount /mnt
```

- b. Use the lvremove command to remove the snapshot.

- Answer **y** when asked, "Do you really want to ..."

```
lvremove myvolg/myvol-snapshot
Do you really want to remove active logical volume myvolg/myvol-
snapshot? [y/n]: y
Logical volume "myvol-snapshot" successfully removed
```

- c. List the logical volumes.

```
lvs
 LV VG Attr LSize Pool Origin Data% ...
 myvol myvolg -wi-ao---- 500.00m
...
```

- Note that the myvol-snapshot logical volume has been deleted.

- d. List the contents of the /dev/myvolg and /dev/mapper directories.

```
ls -l /dev/myvolg
lrwxrwxrwx. ... myvol -> ../dm-2
ls -l /dev/mapper
crw-rw----. ... control
lrwxrwxrwx. ... myvolg-myvol -> ../dm-2
...
```

- Note that the "snapshot" entries have been deleted.

- e. Use the ls -l command to list the contents of the /etc/lvm/backup and /etc/lvm/archive directories.

```
ls -l /etc/lvm/backup
-rw-----. ... <date_time> myvolg
...
ls -l /etc/lvm/archive
-rw-----. ... <date_time> file_today
```

```
-rw----- . . . <date_time> myvolg_00000-...vg
-rw----- . . . <date_time> myvolg_00001-...vg
-rw----- . . . <date_time> myvolg_00002-...vg
-rw----- . . . <date_time> myvolg_00003-...vg
...
```

- Note that a new myvolg backup file was automatically created when the snapshot was removed (note the time stamp).
- Also note that a new archive file was automatically created (note the time stamp on the myvolg\_00003\* file).

## Practice 12-6: Increasing the Capacity of a Logical Volume

---

### Overview

In this practice, you increase the size of a logical volume and add a physical volume to a volume group.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Increase the size of a logical volume.

- a. List the volume group.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 1 1 0 wz--n- 1020.00m 520.00m
...
```

- Note that the `myvolg` volume group has 520 MB of free space.

- b. List the logical volume.

```
lvs
LV VG Attr LSize Pool Origin Data% ...
myvol myvolg -wi-ao--- 500.00m
...
```

- Note that the `myvol` logical volume is 500 MB in size.

- c. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-myvol
 477M 2.3M 445M 1% /myvol
```

- Note that the size of the file system is 477 MB.

- d. Use the `lvextend` command to increase the size of the `myvolg/myvol` logical volume and the file system by 500 MB.

```
lvextend -L +500m -r myvolg/myvol
Size of logical volume myvolg/myvol changed from 500.00 MiB
(125 extents) to 1000.00 MiB (250 extents).
Logical volume myvol successfully resized
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/myvolg-myvol is mounted on /myvol; on-
line resizing required
old_desc_blocks = 4, new_desc_blocks = 8
```

```
The filesystem on /dev/mapper/myvolg-myvol is now 1024000 blocks long.
```

- The `-r` option causes the file system to be resized.

e. List the volume group.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 1 1 0 wz--n- 1020.00m 20.00m
...
```

- Note that the `myvolg` volume group now has only 20 MB of free space.

f. List the logical volume.

```
lvs
LV VG Attr LSize Pool Origin Data% ...
myvol myvolg -wi-ao--- 1000.00m
...
```

- Note that the `myvol` logical volume is now 1,000 MB in size.

g. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-myvol
 961M 2.5M 910M 1% /myvol
```

- Note that the size of the file system is now 961 MB.

h. Use the `ls -l` command to list the contents of the `/etc/lvm/backup` and `/etc/lvm/archive` directories.

```
ls -l /etc/lvm/backup
-rw-----. . . <date_time> myvolg
...
ls -l /etc/lvm/archive
-rw-----. . . <date_time> file_today
-rw-----. . . <date_time> myvolg_00000-...vg
-rw-----. . . <date_time> myvolg_00001-...vg
-rw-----. . . <date_time> myvolg_00002-...vg
-rw-----. . . <date_time> myvolg_00003-...vg
-rw-----. . . <date_time> myvolg_00004-...vg
...
```

- Note that a new `myvolg` backup file was automatically created when the logical volume was extended (note the time stamp).
- Also note that a new archive file was automatically created (note the time stamp on the `myvolg_00004*` file).

2. Add a physical volume to a volume group.

a. List the physical volumes.

```
pvs
PV VG Fmt Attr PSize PFree
...
/dev/xvdc2 myvolg lvm2 a-- 1020.00m 20.00m
/dev/xvdd2 lvm2 --- 1.00g 1.00g
```

- Note that the `/dev/xvdd2` physical volume is not assigned to a volume group.

b. List the volume group.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 1 1 0 wz--n- 1020.00m 20.00m
...
```

- Note that the `myvolg` volume group is 1020 MB in size.

c. Use the `vgextend` command to add the `/dev/xvdd2` physical volume to the `myvolg` volume group.

```
vgextend -v myvolg /dev/xvdd2
Wiping internal VG cache
Wiping cache of LVM-capable devices
Wiping signatures on new PV /dev/xvdd2.
Archiving volume group "myvolg" metadata (seqno 7).
Adding physical volume '/dev/xvdd2' to volume group 'myvolg'.
Volume group "myvolg" will be extended by 1 new physical
volumes
Creating volume group backup "/etc/lvm/backup/myvolg" (seqno
8).
Volume group "myvolg" successfully extended
```

d. List the physical volumes.

```
pvs
PV VG Fmt Attr PSize PFree
...
/dev/xvdc2 myvolg lvm2 a-- 1020.00m 20.00m
/dev/xvdd2 myvolg lvm2 a-- 1020.00m 1020.00m
```

- Note that the `/dev/xvdd2` physical volume is now assigned to the `myvolg` volume group.

e. List the volume group.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 2 1 0 wz--n- 1.99g <1.02g
...
```

- Note that the `myvolg` volume group now has two physical volumes (PVs).

- Note that VSize and VFree have increased.

f. List the logical volume.

```
lvs
 LV VG Attr LSize Pool Origin Data% ...
 myvol myvolg -wi-ao---- 1000.00m
 ...
```

g. Use the `ls -l` command to list the contents of the `/etc/lvm/backup` and `/etc/lvm/archive` directories.

```
ls -l /etc/lvm/backup
-rw-----. . . <date_time> myvolg
...
ls -l /etc/lvm/archive
-rw-----. . . <date_time> file_today
-rw-----. . . <date_time> myvolg_00000-...vg
-rw-----. . . <date_time> myvolg_00001-...vg
-rw-----. . . <date_time> myvolg_00002-...vg
-rw-----. . . <date_time> myvolg_00003-...vg
-rw-----. . . <date_time> myvolg_00004-...vg
-rw-----. . . <date_time> myvolg_00005-...vg
...
```

- Note that a new `myvolg` backup file was automatically created when the physical volume was added to the volume group (note the time stamp).
- Also note that a new archive file was automatically created (note the time stamp on the `myvolg_00005*` file).

3. Increase the size of the `myvolg/myvol` logical volume and the file system.

a. Increase the size of the `myvolg/myvol` logical volume and the file system by 500 MB.

```
lvextend -L +500m -r myvolg/myvol
 Size of logical volume myvolg/myvol changed from 1000.00 MiB
 (250 extents) to 1.46 GiB (375 extents).
 Logical volume myvolg/myvol successfully resized.
 resize2fs 1.42.9 (28-Dec-2013)
 Filesystem at /dev/mapper/myvolg-myvol is mounted on /myvol; on-
 line resizing required
 old_desc_blocks = 8, new_desc_blocks = 12
 The filesystem on /dev/mapper/myvolg-myvol is now 1536000 blocks
 long.
```

b. List the logical volume.

```
lvs
 LV VG Attr LSize Pool Origin Data% ...
 myvol myvolg -wi-ao---- 1.46g
 ...
```

- c. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-myvol
 1.5G 2.7M 1.4G 1% /myvol
```

- Note that the size of the file system is now 1.5 GB.

- d. Use the `ls -l` command to list the contents of the `/etc/lvm/backup` and `/etc/lvm/archive` directories.

```
ls -l /etc/lvm/backup
-rw-----. ... <date_time> myvolg
...
ls -l /etc/lvm/archive
-rw-----. ... <date_time> file_today
-rw-----. ... <date_time> myvolg_00000-...vg
-rw-----. ... <date_time> myvolg_00001-...vg
-rw-----. ... <date_time> myvolg_00002-...vg
-rw-----. ... <date_time> myvolg_00003-...vg
-rw-----. ... <date_time> myvolg_00004-...vg
-rw-----. ... <date_time> myvolg_00005-...vg
-rw-----. ... <date_time> myvolg_00006-...vg
...
```

- Note that a new `myvolg` backup file was automatically created when the logical volume was extended (note the time stamp).
- Also note that a new archive file was automatically created (note the time stamp on the `myvolg_00006*` file).

## Practice 12-7: Restoring Volume Group Metadata

---

### Overview

In this practice, you restore volume group metadata from a backup.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

- View the current physical volumes, volume group, and logical volumes configuration.
  - Use the `pvs` command to display information about the physical volumes.

```
pvs
PV VG Fmt Attr PSize PFree
...
/dev/xvdc2 myvolg lvm2 a-- 1020.00m 0
/dev/xvdd2 myvolg lvm2 a-- 1020.00m 540.00m
```

- Use the `vgs` command to display information about the volume groups.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 2 1 0 w--n- 1.99g 540.00m
...
```

- Use the `lvs` command to display information about the logical volumes.

```
lvs
LV VG Attr LSize Pool Origin Data% Move Log ...
myvol myvolg -wi-ao--- 1.46g
...
```

- Restore the volume group metadata from a backup.

- Use the `vgcfgrestore` command to restore the `myvolg` volume group metadata.

```
vgcfgrestore myvolg
Restored volume group myvolg
```

- Repeat the previous tasks to execute the `pvs`, `vgs`, and `lvs` commands.

```
pvs
PV VG Fmt Attr PSize PFree
...
/dev/xvdc2 myvolg lvm2 a-- 1020.00m 0
/dev/xvdd2 myvolg lvm2 a-- 1020.00m 540.00m

vgs
VG #PV #LV #SN Attr VSize VFree
```

```
myvolg 2 1 0 wz--n- 1.99g 540.00m
...
lvs
LV VG Attr LSize Pool Origin Data% Move Log ...
myvol myvolg -wi-ao---- 1.46g
...
```

- Note that the LVM information is the same as the output in Task 1.

## Practice 12-8: Creating a Thinly Provisioned Logical Volume

---

### Overview

In this practice, you create a thin pool and two thinly provisioned logical volumes. You then monitor the allocated pool data and extend the size of the thin pool.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Remove the logical volume.

- a. Unmount the `myvol` logical volume.

```
umount /myvol
```

- b. Use the `lvremove` command to remove the `myvol` logical volume.

- Answer `y` when asked, “Do you really want to ...”

```
lvremove myvolg/myvol
```

```
Do you really want to remove active logical volume myvolg/myvol?
[y/n]: y
```

```
Logical volume "myvol" successfully removed
```

- c. Remove the `/myvol` mount point.

```
rmdir /myvol
```

- d. By using the `vi` editor, remove the following line from `/etc/fstab`:

```
vi /etc/fstab
```

```
/dev/mapper/myvolg-myvol /myvol ext4 defaults 0 0
```

2. Remove a physical volume from a volume group.

- a. Use the `vgs` command to display information about the volume groups.

```
vgs
```

VG	#PV	#LV	#SN	Attr	VSize	VFree
myvolg	2	0	0	wz--n-	1.99g	1.99g
...						

- Note that the `myvolg` volume group has two physical volumes (PV).

- b. Use the `pvs` command to display information about the physical volumes.

```
pvs
```

PV	VG	Fmt	Attr	PSize	PFree
...					
/dev/xvdc2	myvolg	lvm2	a--	1020.00m	1020.00m
/dev/xvdd2	myvolg	lvm2	a--	1020.00m	1020.00m

- Notice that both physical volumes are allocated to the `myvolg` volume group.

- c. Use the `vgreduce` command to remove a physical volume from a volume group.

- Remove `/dev/xvdd2` from `myvolg`.

```
vgreduce myvolg /dev/xvdd2
Removed "/dev/xvdd2" from volume group "myvolg"
```

- d. Use the `vgs` command to display information about the volume groups.

```
vgs
VG #PV #LV #SN Attr VSize VFree
myvolg 1 0 0 wz--n- 1020.00m 1020.00m
...
```

- Note that the `myvolg` volume group now has one physical volume.

- e. Use the `pvs` command to display information about the physical volumes.

```
pvs
PV VG Fmt Attr PSize PFree
...
/dev/xvdc2 myvolg lvm2 a-- 1020.00m 1020.00m
/dev/xvdd2 lvm2 --- 1.00g 1.00g
```

- Note that only the `/dev/xvdc2` physical volume is allocated to the `myvolg` volume group.

### 3. Create a thin pool.

- a. Use the `lvcreate` command to create a 100 MB thin pool named `mythinpool` in the `myvolg` volume group.
- Note that you must specify the size of the pool because you are creating a pool of physical space.

```
lvcreate -v -L 100m -T myvolg/mythinpool
Executing: /usr/sbin/modprobe dm-thin-pool
Setting chunk size to 64.00 KiB.
...
Logical volume "mythinpool" created
```

- b. Use the `lvs` command to display information about the logical volumes.

```
lvs
LV VG Attr LSize Pool Origin Data% ...
mythinpool myvolg twi-a-tz-- 100.00m 0.00
...
```

- c. List the contents of the `/dev/myvolg` directory.

```
ls -l /dev/myvolg
total 0
lrwxrwxrwx. 1 root root 7 <date_time> mythinpool -> ../../dm-4
```

### 4. Create two thinly provisioned volumes.

- a. Use the `lvcreate` command to create a 200 MB thin volume named `thinvol1` in the `myvolg/mythinpool` thin pool.

- Note that you are specifying a virtual size for the thin volume that is greater than the pool that contains it.

```
lvcreate -v -V 200m -T myvolg/mythinpool -n thinvol1
```

Making thin LV thinvol1 in pool mythinpool in VG myvolg using segtype thin.

...

Logical volume "thinvol1" created

- Use the `lvcreate` command to create another 200 MB thin volume named `thinvol2` in the `myvolg/mythinpool` thin pool.

- Do not include the `-v` (verbose) option.

```
lvcreate -V 200m -T myvolg/mythinpool -n thinvol2
```

WARNING: Sum of all thin volume sizes (400.00 MiB) exceeds the size of thin pool myvolg/mythinpool (100.00 MiB).

WARNING: You have not turned on protection against thin pools running out of space.

WARNING: Set activation/thin\_pool\_autoextend\_threshold below 100 to trigger automatic extension of thin pools before they get full.

Logical volume "thinvol2" created.

- Use the `lvs` command to display information about the logical volumes.

```
lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data...
mythinpool	myvolg	twi-aotz--	100.00m			0.00
thinvol1	myvolg	Vwi-a-tz--	200.00m	mythinpool		0.00
thinvol2	myvolg	Vwi-a-tz--	200.00m	mythinpool		0.00
...						

- Note that the "Data%" column values are 0.00.

- List the contents of the `/dev/myvolg` and `/dev/mapper` directories.

```
ls -l /dev/myvolg
```

...

lrwxrwxrwx. ... thinvol1 -> ../dm-6

lrwxrwxrwx. ... thinvol2 -> ../dm-7

```
ls -l /dev/mapper
```

...

lrwxrwxrwx. ... myvolg-thinvol1 -> ../dm-6

lrwxrwxrwx. ... myvolg-thinvol2 -> ../dm-7

...

- In this example, the files that represent the thin volumes are symbolic links to `dm-6` and `dm-7`.

## 5. Create a file system on the thin volume.

- Create an `ext4` file system on the `thinvol1` thin volume.

```
mkfs.ext4 /dev/myvolg/thinvol1
```

```
...
Writing superblocks and filesystem accounting information: done
```

- b. Create an ext4 file system on the thinvol2 thin volume.

```
mkfs.ext4 /dev/myvolg/thinvol2
```

```
...
```

```
Writing superblocks and filesystem accounting information: done
```

- c. Create the /myvol1 and /myvol2 directories.

```
mkdir /myvol1 /myvol2
```

- d. Mount the file systems.

- Mount /dev/myvolg/thinvol1 on /myvol1.
- Mount /dev/myvolg/thinvol2 on /myvol2.

```
mount /dev/myvolg/thinvol1 /myvol1
```

```
mount /dev/myvolg/thinvol2 /myvol2
```

- e. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/mapper/myvolg-thinvol1
 190M 1.6M 175M 1% /myvol1
/dev/mapper/myvolg-thinvol2
 190M 1.6M 175M 1% /myvol2
```

- Note that the df command shows the size of the file system as 190M. This is an over-allocation of the actual available storage in the thin pool.

- f. Use the lvs command to display information about the logical volumes.

```
lvs
 LV VG Attr LSize Pool Origin Data...
 mythinpool myvolg twi-aotz-- 100.00m 22.12
 thinvol1 myvolg Vwi-aotz-- 200.00m mythinpool 5.53
 thinvol2 myvolg Vwi-aotz-- 200.00m mythinpool 5.53
 ...
```

- This shows that you have used 22.12% of the allocated pool data (100 MB).
- This also shows that each thin volume has used 5.53% of 200 MB.

6. Copy files to a thin volume and monitor usage.

- a. Copy /boot/vmlinuz\* to /myvol1.

```
cp /boot/vmlinuz* /myvol1
```

- b. Run the sync command and then run the lvs command to display information about the logical volumes.

```
sync
```

```
lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data...
mythinpool	myvolg	twi-aotz--	100.00m			40.12
thinvol1	myvolg	Vwi-aotz--	200.00m	mythinpool		14.53
thinvol2	myvolg	Vwi-aotz--	200.00m	mythinpool		5.53
...						

- This shows that you have used 40.12% of the allocated pool data (100 MB).
- This also shows that the thin volume mounted on /myvol1, thinvol1, has used 14.53% of 200 MB.

- c. Use the `lvextend` command to increase the size of the myvolg/mthinpool thin pool to 500 MB.

```
lvextend -L 500m myvolg/mythinpool
Size of logical volume myvolg/mythinpool_tdata changed from
100.00 MiB (25 extents) to 500.00 MiB (125 extents).
Logical volume myvolg/mythinpool_tdata successfully resized.
```

- d. Use the `lvs` command to display information about the logical volumes.

```
lvs
LV VG Attr LSize Pool Origin Data...
mythinpool myvolg twi-aotz-- 500.00m 8.02
thinvol1 myvolg Vwi-aotz-- 200.00m mythinpool 14.53
thinvol2 myvolg Vwi-aotz-- 200.00m mythinpool 5.53
...
```

- Note that the size of the thin pool is 500 MB and the percentage used is 8.02%.

## Practice 12-9: Using Snapper with LVM Thin Provisioned Logical Volumes

### Overview

In this practice, you install the snapper software package, create a snapper configuration file, and use snapper commands to create and manage snapshots.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Install the snapper software package and view files provided by the package.
  - a. Use the `yum` command to install the snapper software package.
    - Answer `y` to install the packages.

```
yum install snapper
...
Is this ok [y/d/N]: y
...
Complete!
```

- b. Use the `rpm -ql` command to view the files provided by the snapper package.

```
rpm -ql snapper
/etc/cron.daily/snapper
/etc/cron.hourly/snapper
/etc/dbus-1/system.d/org.opensus.Snapper.conf
/etc/logrotate.d/snapper
/usr/bin/snapper
...
/usr/sbin/snapperd
/usr/share/dbus-1/system-services/org.opensus.Snapper.service
...
/usr/share/man/man5/snapper-configs.5.gz
/usr/share/man/man8/snapper.8.gz
/usr/share/man/man8/snapperd.8.gz
```

- Note the two `cron` snapper files.
  - By default, snapper sets up a `cron.hourly` job to create snapshots in the `.snapshots` subdirectory of the volume and a `cron.daily` job to clean up old snapshots.
  - You can edit the configuration file to disable or change this behavior.

2. Create a snapper configuration file for the LVM thin volume mounted on /myvol1.
  - a. Use the `snapper create-config` command to create a configuration file named `myvol1_snap` for the LVM ext4 file system mounted on /myvol1.

```
snapper -c myvol1_snap create-config -f "lvm(ext4)" /myvol1
```

- This command adds an entry to /etc/sysconfig/snapper.
- This command creates the /etc/snapper/configs/myvol1\_snap configuration file.
- This command creates a .snapshots directory in the /myvol1 directory.

- b. View the contents of the /etc/sysconfig/snapper file.

```
cat /etc/sysconfig/snapper
...
SNAPPER_CONFIGS="myvol1_snap"
```

- c. View the snapper configuration file for the LVM volume.

```
cat /etc/snapper/configs/myvol1_snap
...
subvolume to snapshot
SUBVOLUME="/myvol1"

filesystem type
FSTYPE="lvm(ext4)"
...

start comparing pre- and post-snapshot in background after...
BACKGROUND_COMPARISON="yes"

run daily number cleanup
NUMBER_CLEANUP="yes"
...

create hourly snapshots
TIMELINE_CREATE="yes"

cleanup hourly snapshots after some time
TIMELINE_CLEANUP="yes"
...

cleanup empty pre-post-pairs
EMPTY_PRE_POST_CLEANUP="yes"
...
```

- Note that a description of the parameters in the snapper configuration file is found in the `snapper-configs(5)` man page.

- d. Use the `ls -la` command to view a long list of all files in the /myvol1 directory.

```
ls -la /myvol1
...
```

```
drwxr-x---snapshots
...
```

- Snapshots of the /myvol1 file system are stored in the .snapshots subdirectory.

3. Create pre and post snapshots of the /myvol1 file system.

- a. Use the `snapper create -t pre` to create a pre snapshot of the volume defined by the `myvol1_snap` configuration file.

- Include the `-p` option to display the number of the snapshot being created.

```
snapper -c myvol1_snap create -t pre -p
1
```

- In this example, the pre snapshot number is 1. This might be different in your case.

- b. View the contents of the /myvol1/.snapshots directory.

```
ls -l /myvol1/.snapshots
...
drwxr-xr-x ... 1
```

- c. View the contents of the /myvol1/.snapshots/1 directory.

- This example uses 1 as the pre snapshot number.
- Use the number returned in task 3a.

```
ls -l /myvol1/.snapshots/1
...
-rw----- ... info.xml
drwxr-x--- ... snapshot
```

- d. View the `info.xml` file in the /myvol1/.snapshots/1 directory.

```
cat /myvol1/.snapshots/1/info.xml
<?xml version="1.0"?>
<snapshot>
 <type>pre</type>
 <num>1</num>
 <date>...</date>
</snapshot>
```

- e. Modify the contents of the volume by deleting the `vmlinuz-3.10*` file.

```
rm /myvol1/vmlinuz-3.10*
rm: remove regular file '/myvol1/vmlinuz-3.10...' ? y
```

- f. Use `snapper create -t post` to create a post snapshot of the volume defined by the `myvol1_snap` configuration file.

- Include the `--pre-num 1` option to associate this post snapshot with the pre snapshot 1.
- Include the `-p` option to display the number of the snapshot being created.

```
snapper -c myvol1_snap create -t post --pre-num 1 -p
2
```

- In this example, the snapshot number is 2. This might be different in your case.
- g. View the contents of the `/myvol1/.snapshots/2` directory.
- This example uses 2 as the post snapshot number.
  - Use the number returned in Task 3f.

```
ls -l /myvol1/.snapshots/2
...
-rw----- ... info.xml
drwxr-x--- ... snapshot
```

- h. View the `info.xml` file in the `/myvol1/.snapshots/2` directory.

```
cat /myvol1/.snapshots/2/info.xml
<?xml version="1.0"?>
<snapshot>
 <type>post</type>
 <num>2</num>
 <date>...</date>
 <pre_num>1</pre_num>
</snapshot>
```

4. List the snapshots that exist for the `/myvol1` volume.

- Use the `snapper list` command to list the snapshots that exist for volume defined by the `myvol1_snap` configuration file.
- The sample output is shown. Your output may differ.

Type	#	Pre #	Date	User	Cleanup	Description	Userdata
single	0		...	root		current	
pre	1		...	root			
post	2	1	...	root			

5. Undo the changes from post snapshot 2 to pre snapshot 1.

- List the contents of the `/myvol1` volume.

```
ls /myvol1
lost+found
vmlinuz-0-rescue-...
vmlinuz-4.1.12-94.3.9.el7uek.x86_64
...
```

- Recall that `vmlinuz-3.10*` was deleted from `/myvol1` after the pre snapshot number 1 and before the post snapshot number 2.
- Undoing the change restores the `vmlinuz-3.10*` file to `/myvol1`.

6. Remove the snapper package.

- Use the `yum` command to remove snapper. This will remove the `cron.hourly` and `cron.daily` snapper cron jobs.

```
yum remove snapper
Loaded plugins: langpacks, ulninfo
Resolving Dependencies
...
Is this ok [y/N]: y
...
Removed:
snapper.x86_64 0:0.2.8-4.el7

Complete!
```

## Practice 12-10: Creating a RAID Device

---

### Overview

In this practice, you remove the logical volume and LVM entities, create a RAID array device, and create a file system, and mount the RAID device. You remove the RAID device at the end of the practice.

### Assumptions

You are the root user on the **host03** VM.

### Tasks

1. Remove the logical volume, volume group, and physical volume.

- a. Unmount the `myvol1` and `myvol2` logical volumes.

```
umount /myvol1 /myvol2
```

- b. Use the `lvremove` command to remove the `mythinpool` logical volume.

- Answer **y** to all queries.
- Removing a thin pool removes dependent thin volumes.

```
lvremove myvolg/mythinpool
```

Removing pool "mythinpool" will remove ... dependent volume(s).  
Proceed? [y/n]: **y**

Do you really want to remove active logical volume  
`myvolg/thinvol1`? [y/n]: **y**

Logical volume "thinvol1" successfully removed

Do you really want to remove active logical volume  
`myvolg/thinvol2`? [y/n]: **y**

Logical volume "thinvol2" successfully removed

Logical volume "thinvol1-snapshot1" successfully removed

Logical volume "thinvol1-snapshot2" successfully removed

...

Do you really want to remove active logical volume  
`myvolg/mythinpool`? [y/n]: **y**

Logical volume "mythinpool" successfully removed

- c. Use the `vgremove` command to remove the `myvolg` volume group.

```
vgremove myvolg
```

Volume group "myvolg" successfully removed

- d. Use the `pvremove` command to remove the physical volumes.

```
pvremove /dev/xvdc2 /dev/xvdd2
```

Labels on physical volume "/dev/xvdc2" successfully wiped

Labels on physical volume "/dev/xvdd2" successfully wiped

- e. Remove the /myvol1 and /myvol2 mount points.

```
rmdir /myvol1 /myvol2
```

2. Create a new RAID array.

- a. Use the mdadm command to create a RAID-1 device, /dev/md0, from the /dev/xvdc2 and /dev/xvdd2 partitions.

- Answer **y** when asked, “Continue creating array?”

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/xvdc2
/dev/xvdd2
```

```
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
```

```
Continue creating array? y
```

```
mdadm: Defaulting to version 1.2 metadata
```

```
mdadm: array /dev/md0 started.
```

- b. View the /proc/mdstat file to check the status of the MD RAID devices. You may or may not see a resync message, depending on how soon you run this command.

```
cat /proc/mdstat
```

```
Personalities : [raid1]
md0 : active raid1 xvdd2[1] xvdc2[0]
 1046528 blocks super 1.2 [2/2] [UU]
 [======>.....] resync = 62.1% (652124/1047552)
finish=0.3min speed=16634K/sec

unused devices: <none>
```

- c. Run the previous command again to confirm the resync has completed.

```
cat /proc/mdstat
```

```
Personalities : [raid1]
md0 : active raid1 xvdd2[1] xvdc2[0]
 1046528 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

- Ensure that the resync is complete before continuing.

- d. Use the mdadm command to view information about the RAID device.

```
mdadm --query /dev/md0
```

```
/dev/md0: 1022.00MiB raid1 2 devices, 0 spares. Use mdadm --
detail for more detail.
```

```
mdadm --detail /dev/md0
```

```
/dev/md0:
 Version : 1.2
 Creation Time : ...
 Raid Level : raid1
 Array Size : 1046528 (1021.00 MiB 1071.64 MB)
 Used Dev Size : 1046528 (1021.00 MiB 1071.64 MB)
 Raid Devices : 2
 Total Devices : 2
 Persistence : Superblock is persistent

 Update Time : ...
 State : clean
 Active Devices : 2
 Working Devices : 2
 Failed Devices : 0
 Spare Devices : 0

 Consistency Policy : unknown

 Name : host03.example.com:0 (local to host
host03.example.com)
 UUID : ...
 Events : 17

 Number Major Minor RaidDevice State
 0 202 34 0 active sync /dev/xvdc2
 1 202 50 1 active sync /dev/xvdd2
```

3. Create a file system on the RAID device.

- a. Create an ext4 file system on /dev/md0.

```
mkfs.ext4 /dev/md0
...
Writing superblocks and filesystem accounting information: done
```

- b. Create a mount point named /raid.

```
mkdir /raid
```

- c. Mount the file system.

```
mount /dev/md0 /raid
```

- d. Display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/md0 990M 2.6M 921M 1% /raid
```

- e. Copy `/boot/init*` to the mounted RAID device, `/raid`.

```
cp /boot/init* /raid
ls /raid
initramfs-0-rescue-...
initramfs-3.10.0-862.el7.x86_64.img
initramfs-4.1.12-112.16.4.el7uek.x86_64.img
...
```

- This step confirms you can write to the file system.

4. Create the `mdadm` configuration file, `/etc/mdadm.conf`.

- Use the `vi` editor to create `/etc/mdadm.conf` and add the following entry.
  - This step does not need to be performed in this practice. You are directed to remove this file in the next task.

```
vi /etc/mdadm.conf
ARRAY /dev/md0 devices=/dev/xvdc2,/dev/xvdd2
```

- With this file, the RAID array is properly detected and initialized after a reboot.

5. Remove the RAID array.

- a. Remove the `/etc/mdadm.conf` file.

```
rm /etc/mdadm.conf
rm: remove regular file '/etc/mdadm.conf'? y
```

- b. Unmount the raid volume.

```
umount /raid
```

- c. Remove the raid directory.

```
rmdir /raid
```

- d. Use the `mdadm` command to deactivate the array and release all resources.

```
mdadm --stop /dev/md0
mdadm: stopped /dev/md0
```

- e. Display the attributes of the `/dev/xvdc3` block device.

```
blkid | grep raid
/dev/xvdc2: UUID="..." UUID_SUB="..."
LABEL="host03.example.com:0" TYPE="linux_raid_member"
/dev/xvdd2: UUID="..." UUID_SUB="..."
LABEL="host03.example.com:0" TYPE="linux_raid_member"
```

- Note that `/dev/xvdc2` and `/dev/xvdd2` have a type of "linux\_raid\_member".

- f. Use the `mdadm` command to overwrite the `md` superblock on `/dev/xvdc2` and `/dev/xvdd2`.

```
mdadm --zero-superblock /dev/xvdc2 /dev/xvdd2
```

- g. Display the attributes of `/dev/xvdc2` and `/dev/xvdd2` to ensure that the `md` superblock was overwritten.

- This example also repeats the prior step to show that `/dev/xvdc2` and `/dev/xvdd2` no longer have a type of `linux_raid_member`.

```
blkid /dev/xvdc2
blkid /dev/xvdd2
blkid | grep raid
```

- No output indicates the superblock was overwritten.

## Practice 12-11: Removing Partitions

---

### Overview

In this practice, you delete the partitions on the `/dev/xvdc` and `/dev/xvdd` devices in preparation for the next lesson. You also log off from **host03**.

### Assumptions

You are the `root` user on the **host03** VM.

### Tasks

1. Use the `fdisk` command to delete the partitions on `/dev/xvdc`.
  - Use the `p` command to print the partition table and show that there are two partitions.
  - Use the `d` command and select 1 to delete partition 1.
  - Use the `d` command to delete partition 2. You do not need to specify the partition when only one partition exists.
  - Use the `p` command to print the partition table and confirm there are no partitions.
  - Use the `w` command to save the partition table and exit the `fdisk` utility.

```
fdisk /dev/xvdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p

Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x8e7a9de5

 Device Boot Start End Blocks Id System
/dev/xvdc1 2048 2099199 1048576 83 Linux
/dev/xvdc2 2099200 4196351 1048576 8e Linux
LVM

Command (m for help): d
Partition number (1,2, default 2): 1
```

```

Partition 1 is deleted

Command (m for help): d
Selected partition 2
Partition 2 is deleted

Command (m for help): p

Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x8e7a9de5

 Device Boot Start End Blocks Id System

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

```

2. Use the `fdisk` command to delete the partitions on `/dev/xvdd`.
  - Use the `p` command to print the partition table and show that there are two partitions.
  - Use the `d` command and select 1 to delete partition 1.
  - Use the `d` command to delete partition 2. You do not need to specify the partition when only one partition exists.
  - Use the `p` command to print the partition table and confirm there are no partitions.
  - Use the `w` command to save the partition table and exit the `fdisk` utility.

```

fdisk /dev/xvdd
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p

Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes

```

```

Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: ...

 Device Boot Start End Blocks Id System
/dev/xvdd1 2048 2099199 1048576 83 Linux
/dev/xvdd2 2099200 4196351 1048576 8e Linux
LVM

Command (m for help): d
Partition number (1,2, default 2): 1
Partition 1 is deleted

Command (m for help): d
Selected partition 2
Partition 2 is deleted

Command (m for help): p

Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: ...

 Device Boot Start End Blocks Id System

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

```

3. View the xvdd devices in the /proc/partitions file to ensure that there are no partitions on xvdc and xvdd.

```

grep xvdd /proc/partitions
 202 0 20971520 xvda
 202 1 1048576 xvda1
 202 2 19921920 xvda2
 202 16 4194304 xvdb
 202 32 20971520 xvdc

```

202	48	20971520	xvdd
-----	----	----------	------

4. Log off from **host03**.

Use the `logout` command to close the ssh connection to **host03**.

```
logout
```

```
Connection to host03 closed.
```

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.

## **Practices for Lesson 13: Advanced Storage Administration**

## Practices for Lesson 13: Overview

---

### Practices Overview

In these practices, you:

- Create and mount a file system on `/dev/xvdc`
- Set quotas on a directory
- Explore and configure Udev
- Use the `kpartx` utility

## Practice 13-1: Creating and Mounting a File System

### Overview

In this practice, you:

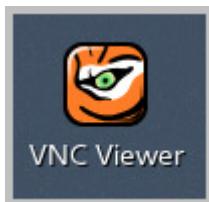
- Create a partition on a storage device
- Create an ext4 file system on the partition
- Mount the file system on /Dev

### Assumptions

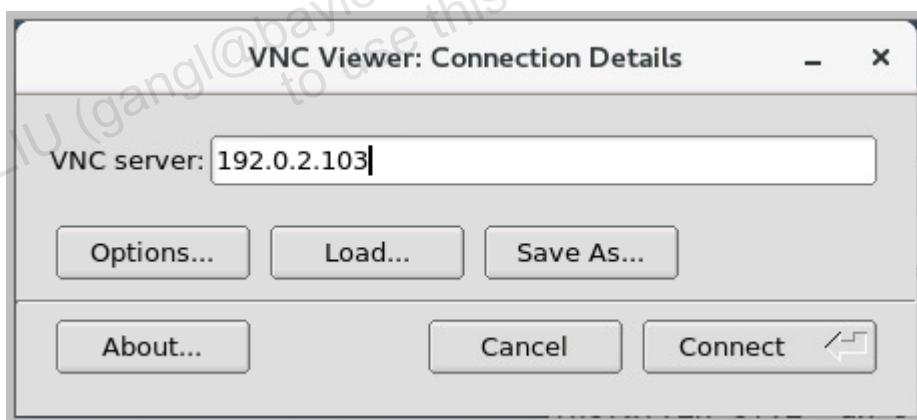
- You are the `root` user on **gateway**.

### Tasks

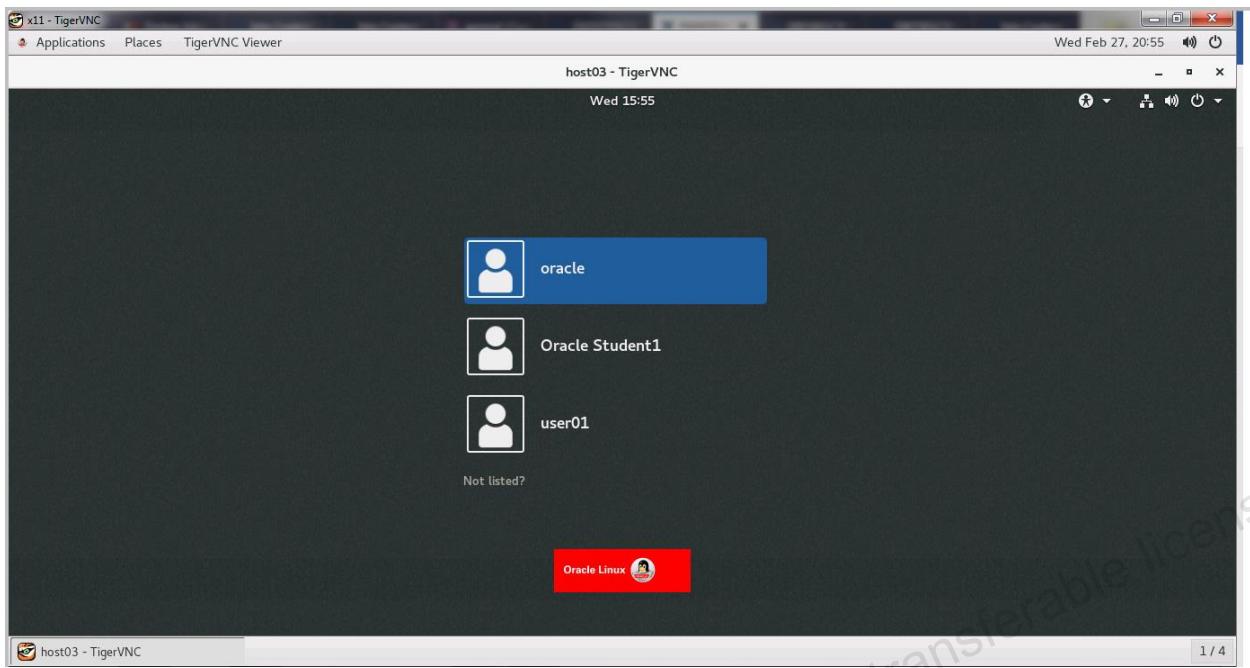
1. Log in to **host03** by using VNC Viewer.
  - a. Double-click the “VNC Viewer” icon on the **gateway** desktop.



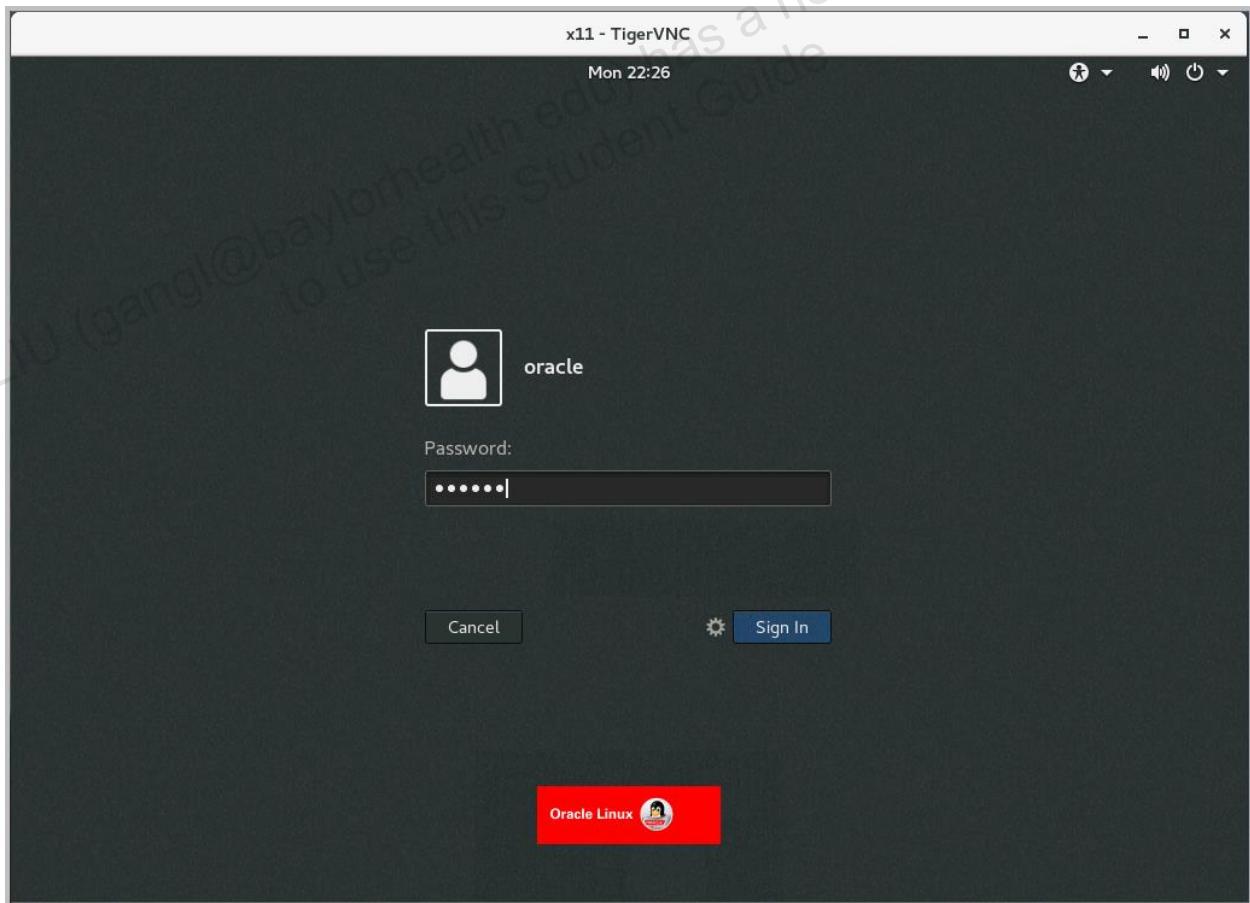
- b. Enter 192.0.2.103 as shown in the following window.
  - Click Connect.



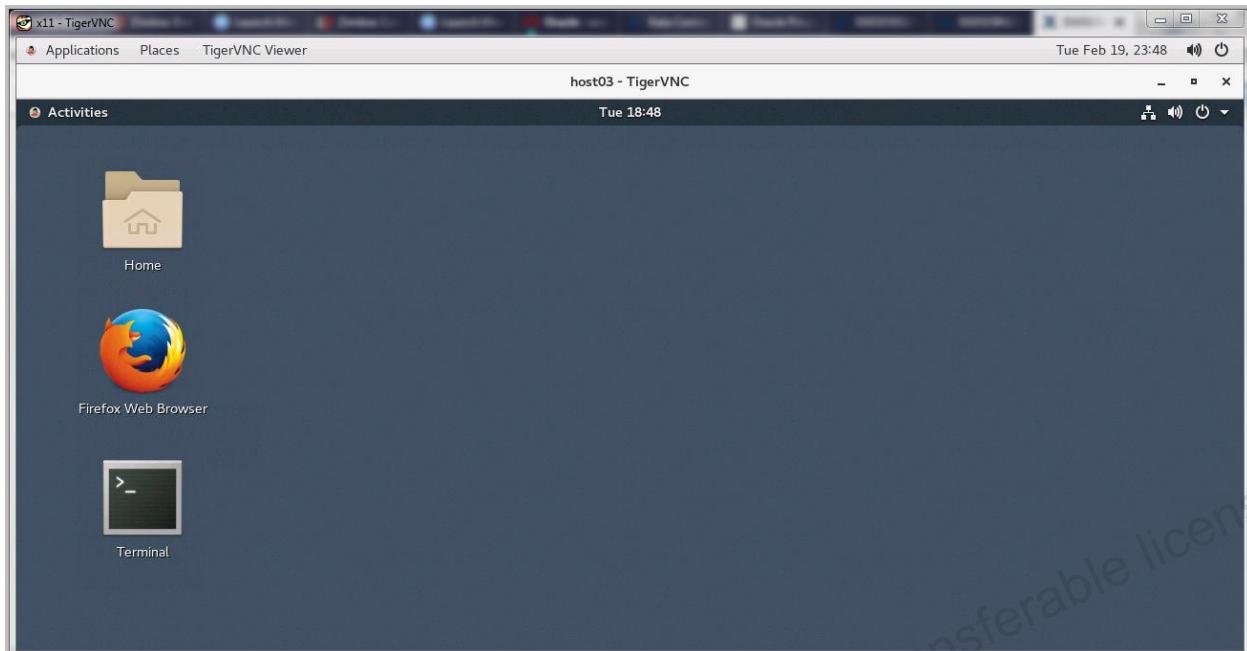
- The login window appears.
- Click in the window to see the `oracle` user.



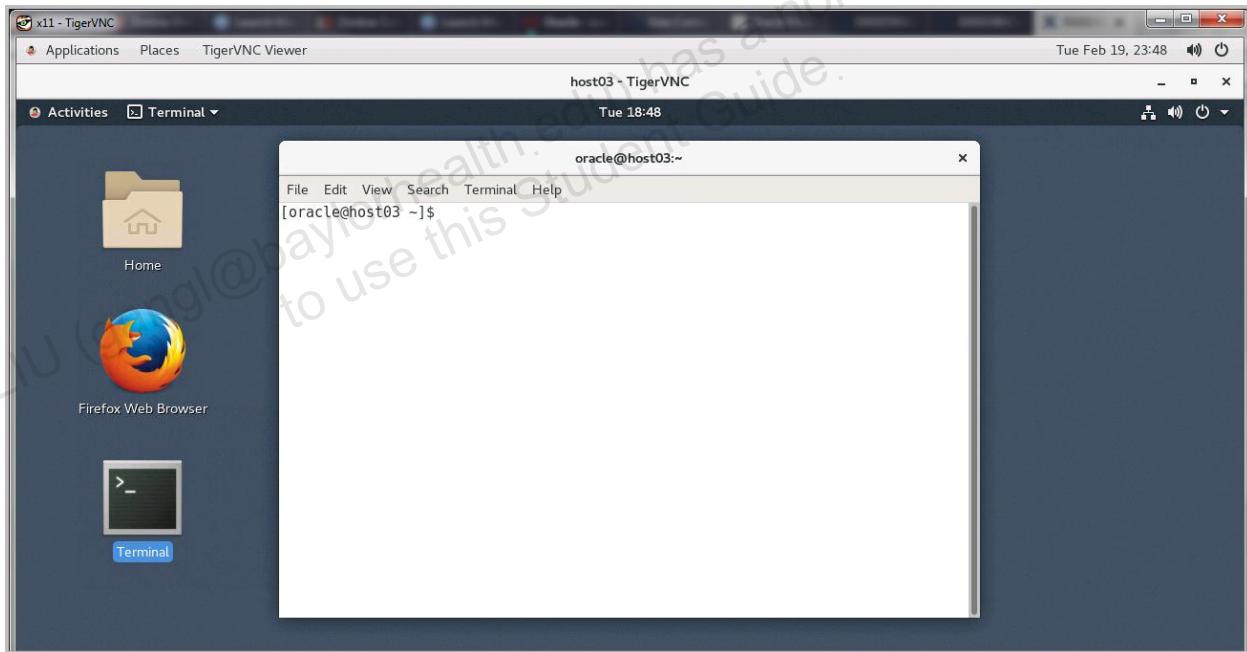
- c. Click the `oracle` user and login.



- The desktop appears as shown.



- d. Double-click the Terminal icon to open a terminal window as shown.



2. Use the `su -` command to become the `root` user.

```
[oracle@host03 ~]$ su -
Password:
Last login: ...
[root@host03 ~]#
```

3. Partition a storage device using fdisk.

- a. As the root user on host03, use the fdisk command to display the partition table.

- The /dev/xvdc device is unused.

```
[root@host03 ~]# fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvda1 * 2048 2099199 1048576 83 Linux
/dev/xvda2 2099200 41943039 19921920 8e Linux
LVM
Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

- b. Use the fdisk command to create a 1 GB partition /dev/xvdc.

```
[root@host03 ~]# fdisk /dev/xvdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type:
 p primary (0 primary, 0 extended, 4 free)
 e extended
Select (default p):
Using default response p
Partition number (1-4, default 1): ENTER
First sector (2048-41943039, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default
41943039): +1G
Partition 1 of type Linux and of size 1 GiB is set

Command (m for help): p

Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x8e7a9de5

 Device Boot Start End Blocks Id System
/dev/xvdc1 2048 2099199 1048576 83 Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

4. Create a file system on /dev/xvdc1.

- Use the `mkfs` command to make an ext4 file system on /dev/xvdc1.

```
[root@host03 ~]# mkfs -t ext4 /dev/xvdc1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376
...
Writing superblocks and filesystem accounting information: done
```

5. Mount the file system.

- a. Use the `mkdir` command to create a mount point.

```
[root@host03 ~]# mkdir /Dev
```

- b. Use the `mount` command to mount /dev/xvdc1 on /Dev with ACL support.

- Include the `-o acl` mount option for ACL support.

```
[root@host03 ~]# mount -t ext4 -o acl /dev/xvdc1 /Dev
```

- c. Use the `df` command to display the mounted file systems.

```
df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/xvdc1 976M 2.6M 907M 1% /Dev
```

## Practice 13-2: Setting Disk Quotas

### Overview

In this practice, you:

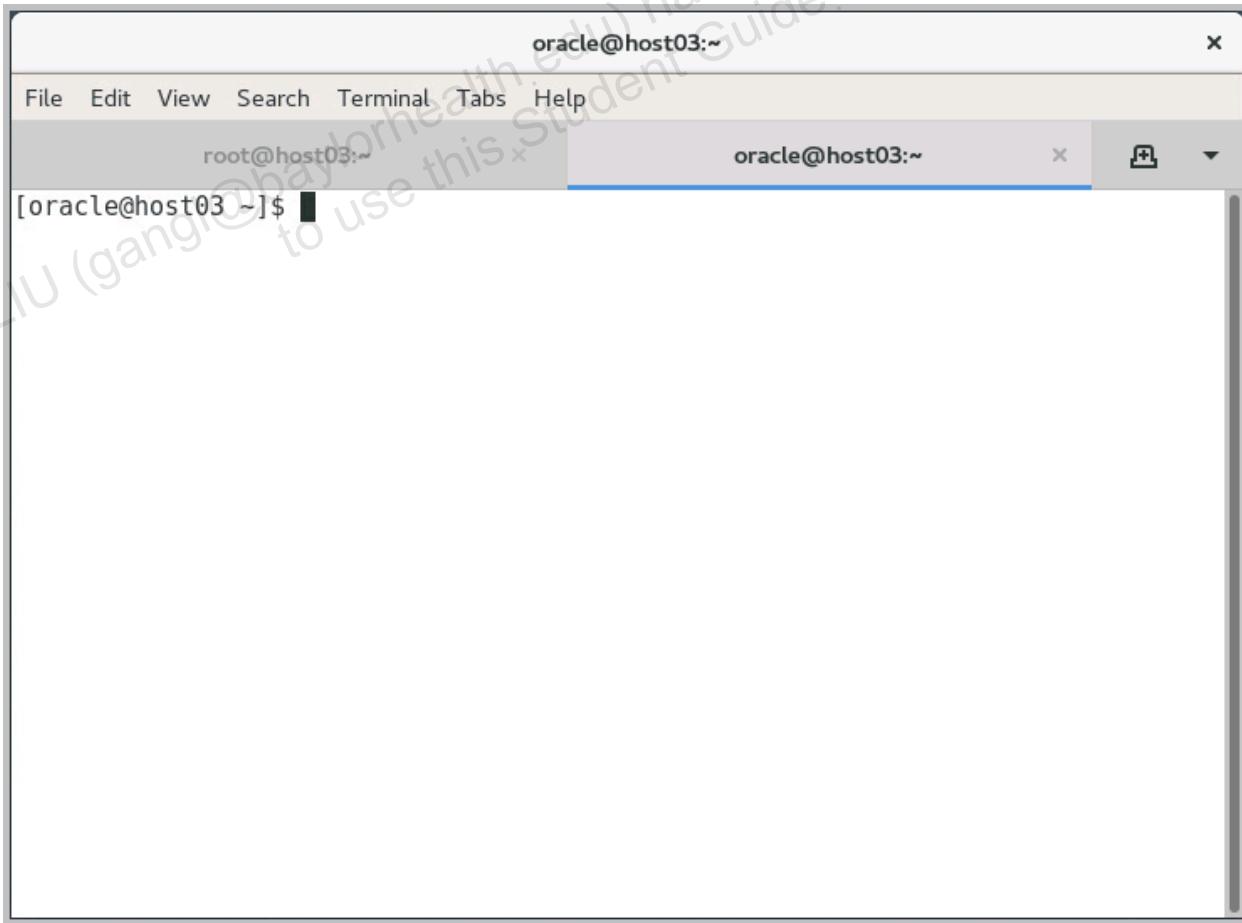
- Set quotas on a directory for the `oracle` user
- Remove the quotas on the directory

### Assumptions

- Ensure that you are using VNC Viewer 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.
  - a. 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. Grant the `oracle` user permissions on the `/Dev` directory.

- a. 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
```

- b. 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.

- Click the “`root@host03`” tab to enter commands as the `root` user.

```
[root@host03]# setfacl -m u:oracle:rwx /Dev
```

- c. 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
```

- d. 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
```

3. 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/xvdc1` 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/xvdc1
/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/xvdc1
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 0
oracle -- 0 0 0 1 0 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 1000):
Filesystem blocks soft hard inodes soft hard
/dev/xvdc1 0 0 0 1 0 0 (old entry)
/dev/xvdc1 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/xvdc1
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 0
oracle -- 0 0 2048 1 0 0 0
```

#### 4. As the oracle user, verify the disk quota setting.

- Click the “oracle@host03” tab to enter commands as the oracle user.
- Use the `dd if=/dev/zero of=bigfile bs=1M count=4096` command to attempt to create a 4 MB file on /Dev.
- Note the “Disk quota exceeded” error message.

```
[oracle@host03]# cd /Dev
[oracle@host03]# dd if=/dev/zero of=bigfile bs=1M count=4096
xvdc1: 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 1000):
Filesystem blocks quota limit grace files quota limit grace
/dev/xvdc1 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 previous execution of `quota` command.

```
[oracle@host03]$ quota
Disk quotas for user oracle (uid 1000):
Filesystem blocks quota limit grace files quota limit grace
/dev/xvdc1 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
```

5. As the `root` user, reset the `/dev/xvdc1` partition.

- Click the “`root@host03`” tab to enter commands as the `root` user.
- 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
```

- Use the `setfac1` command to remove the ACL from the `/Dev` directory.

```
[root@host03]# setfac1 -b /Dev
```

- Use the `umount` command to unmount `/Dev`.

```
[root@host03 ~]# umount /Dev
```

- e. Click the “X” on the “oracle@host03” tab to close the tab.
6. Restore system to original configuration.
  - a. Use the `fdisk` command to remove the partition on `/dev/xvdc`.

```
fdisk /dev/xvdc
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p

Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x8e7a9de5

 Device Boot Start End Blocks Id System
/dev/xvdc1 2048 2099199 1048576 83 Linux

Command (m for help): d
Selected partition 1
Partition 1 is deleted

Command (m for help): p

Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x8e7a9de5

 Device Boot Start End Blocks Id System

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

- b. Remove the /Dev mountpoint.

```
rmdir /Dev
```

- c. Show that the /dev/xvdc1 partition no longer exists.

```
fdisk -l|grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvda1 * 2048 2099199 1048576 83 Linux
/dev/xvda2 2099200 41943039 19921920 8e Linux
LVM
Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

## Practice 13-3: Exploring and Configuring Udev Rules

### 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 -q1` command to view the “udev” files included with the `systemd` RPM package.

```
rpm -q1 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.

- Sample output is shown. Files on your system might be different.

```
ls /lib/udev/rules.d
01-md-raid-creating.rules 73-idrac.rules
100-balloon.rules 73-seat-late.rules
10-dm.rules 75-net-description.rules
...
ls /etc/udev/rules.d
70-persistent-ipoib.rules 99-vmware-scsi-timeout.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-51760/block/xvdd
N: xvdd
S: disk/by-label/Btrfs
...
E: DEVNAME=/dev/xvdd
E: DEVPATH=/devices/vbd-51760/block/xvdd
E: DEVTYPE=disk
E: ID_BTRFS_READY=1
E: ID_FS_LABEL=Btrfs
E: ID_FS_LABEL_ENC=Btrfs
E: ID_FS_TYPE=btrfs
E: ID_FS_USAGE=filesystem
...
E: ID_PART_TABLE_TYPE=dos
E: MAJOR=202
E: MINOR=48
E: MPATH_SBIN_PATH=/sbin
E: SUBSYSTEM=block
E: TAGS=:systemd:
E: USEC_INITIALIZED=75531
```

- 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-51760/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-51760/block/xvdd':
KERNEL=="xvdd"
SUBSYSTEM=="block"
DRIVER==""
ATTR{ro}=="0"
ATTR{size}=="41943040"
...
looking at parent device '/devices/vbd-51760':
KERNELS=="vbd-51760"
SUBSYSTEMS=="xen"
DRIVERS=="vbd"
ATTRS{devtype}=="vbd"
ATTRS{max_queues}=="1"
ATTRS{max_indirect_segs}=="32"
ATTRS{max_ring_page_order}=="0"
ATTRS{nodename}=="device/vbd/51760"
```

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
disk/by-label/Btrfs disk/by-uuid/... 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
```

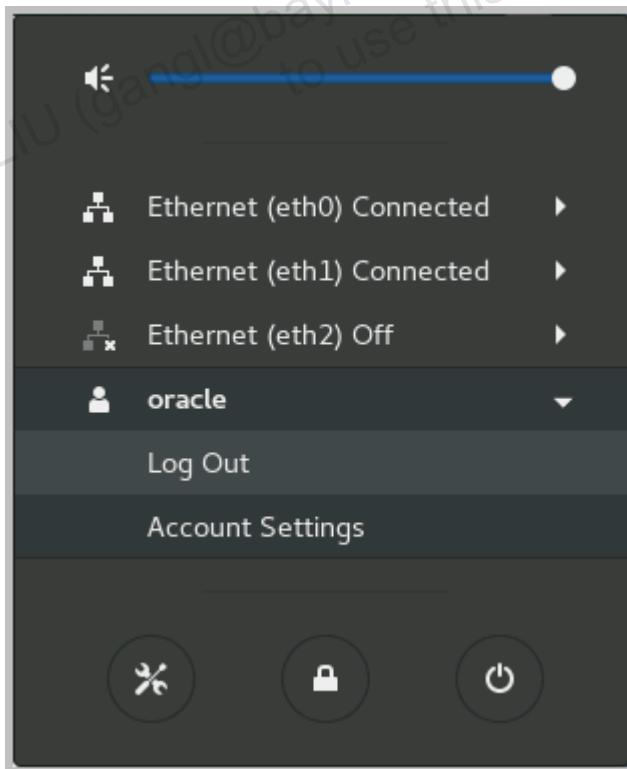
- c. Use the ls command to list the /dev/my\* devices.

- Note that /dev/my\_disk no longer exists.

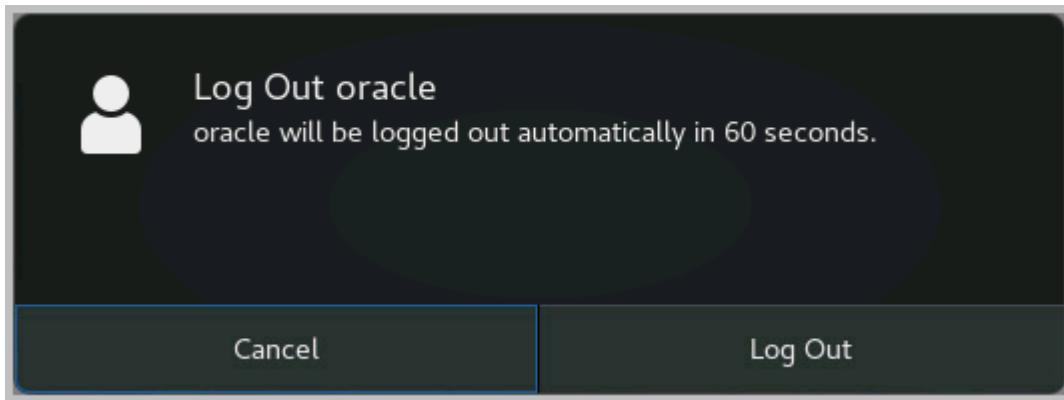
```
ls /dev/my*
ls: cannot access /dev/my*: No such file or directory
```

5. Log out and close the VNC connection to host03.

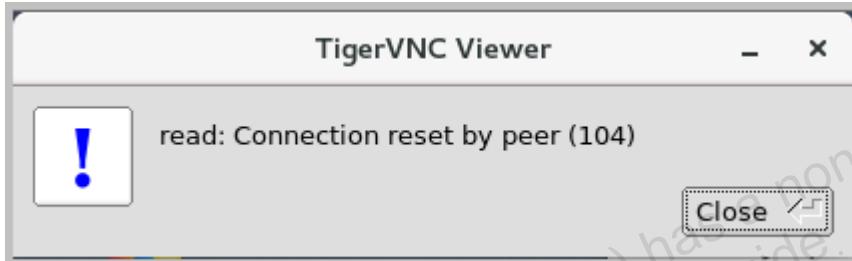
- a. Click the down arrow in the top-right corner of the window to display the drop-down menu.
- b. Click the oracle user, then click Log Out as shown.



- c. Click Log Out as shown.



- The **host03** desktop closes.
- The following window might appear. Close the window by clicking the X in the upper-right corner.



- If you are logged on to **host03** by using ssh, log off.

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

## Practice 13-4: Using kpartx

---

### Overview

In this practice, you use the `kpartx` utility to create device maps from a partition table.

### Assumptions

- You are the root user on **gateway**.
- A `system.img` file, which represents a partitioned disk device for an Oracle VM Server for x86 virtual machine, exists in the `/stage` directory on **gateway**.
- This practice is performed on **gateway**.

### Tasks

1. View the `system.img` file in the `/stage` directory on **gateway**.

- a. Change to the `/stage` directory.

```
[root@gateway]# cd /stage
```

- b. Use the `ls -l` command to list the contents of the directory.

- The `/stage` directory contains files used by various practices in this course.
- Note the `system.img` file, which is used in this practice.

```
[root@gateway stage]# ls -l
...
-rw-r--r-- 1 root root 17179869184 ... system.img
```

- This `system.img` file represents a physical disk for an Oracle VM Server for x86 virtual machine.
- The `system.img` file might appear in a `vm.cfg` file as shown (do not run this command).

```
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,xvdc,w',
 'file:/OVS/running_pool/host03/u02.img,xvdd,w',
 ...]
```

- The bold line in the `vm.cfg` file shows how this `system.img` file is used to represent the `xvda` device.

2. Review the partition information on the `system.img` file.

- The `system.img` file (or the `/dev/xvda` device in the virtual machine) is partitioned and has Oracle Linux installed on the device.
- a. Use the `file` command to view information about the `system.img` file.

- This command shows information related to the partition table on the image file.

```
[root@gateway stage]# file system.img
system.img: x86 boot sector; partition 1: ID=0x83, active,
starthead 32, startsector 2048, 2097152 sectors; partition 2:
ID=0x83, starthead 170, startsector 2099200, 20971520 sectors;
partition 3: ID=0x83, starthead 254, startsector 23070720,
4194304 sectors; partition 4: ID=0x5, starthead 254, startsector
27265024, 6289408 sectors, code offset 0x63
```

- The following command shows exactly how the `xvda` device is partitioned (do not run this command):

```
lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
...
xvda 202:0 0 16G 0 disk
└─xvda1 202:1 0 1G 0 part /boot
└─xvda2 202:2 0 10G 0 part /
└─xvda3 202:3 0 2G 0 part /home
└─xvda4 202:4 0 1K 0 part
└─xvda5 202:5 0 3G 0 part [SWAP]
...
```

- This shows 5 partitions with:

- `xvda1` mounted on `/boot`
- `xvda2` mounted on `/`
- `xvda3` mounted on `/home`
- `xvda4` is not mounted
- `xvda5` is a swap partition

### 3. Review the partition information on the `system.img` file.

- Use the `kpartx -l` command to list the partitions on the `system.img` file.
- The output shows that the `system.img` file contains five partitions.

```
[root@gateway stage]# kpartx -l system.img
loop0p1 : 0 2097152 /dev/loop0 2048
loop0p2 : 0 20971520 /dev/loop0 2099200
loop0p3 : 0 4194304 /dev/loop0 23070720
loop0p4 : 0 2 /dev/loop0 27265024
loop0p5 : 0 6287360 /dev/loop0 27267072
...
```

### 4. Create device maps from the partition table on `system.img`.

- 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.

```
[root@gateway stage]# ls /dev/mapper
control
```

- Use the kpartx -a command to add the device mappings for the detected partitions.

```
[root@gateway stage]# kpartx -a system.img
```

- Use the ls command to list the /dev/mapper directory.

- Note that a file was created for each of the 5 partitions.

```
[root@gateway stage]# ls /dev/mapper
control loop0p1 loop0p2 loop0p3 loop0p4 loop0p5
```

- Recall the 5 partitions are used as follows:

- Partition 1 is mounted on /boot
- Partition 2 is mounted on /
- Partition 3 is mounted on /home
- Partition 4 is not mounted
- Partition 5 is a swap partition

- Mount the loop0p1 device created by the kpartx command.

- Use the mkdir command to create a mount point, /mnt/map1.

```
[root@gateway stage]# mkdir /mnt/map1
```

- Use the mount command to mount /dev/mapper/loop0p1 on /mnt/map1.

```
[root@gateway stage]# mount /dev/mapper/loop0p1 /mnt/map1
```

- Use the ls command to view the files on /mnt/map1.

- Note that these are the files that would be in the /boot directory on an Oracle Linux system.

```
[root@gateway stage]# ls /mnt/map1
config-3.10.0-862.el7.x86_64
config-4.1.12-112.16.4.el7uek.x86_64
efi
grub2
initramfs-0-rescue-3496f337cf70480b9b5f808802b94e6f.img
initramfs-3.10.0-862.el7.x86_64.img
initramfs-4.1.12-112.16.4.el7uek.x86_64.img
lost+found
symvers-3.10.0-862.el7.x86_64.gz
symvers-4.1.12-112.16.4.el7uek.x86_64.gz
System.map-3.10.0-862.el7.x86_64
System.map-4.1.12-112.16.4.el7uek.x86_64
vmlinuz-0-rescue-3496f337cf70480b9b5f808802b94e6f
vmlinuz-3.10.0-862.el7.x86_64
vmlinuz-4.1.12-112.16.4.el7uek.x86_64
```

- d. Use the `umount` command to unmount `/mnt/map1`.

```
[root@gateway stage]# umount /mnt/map1
```

6. Mount the `loop0p2` device created by the `kpartx` command.

- a. Use the `mount` command to mount `/dev/mapper/loop0p2` on `/mnt/map1`.

```
[root@gateway stage]# mount /dev/mapper/loop0p2 /mnt/map1
```

- b. Use the `ls` command to view the files on `/mnt/map1`.

- Note that these are the files that would be in the `/` (root) directory on an Oracle Linux system.

```
[root@gateway stage]# ls /mnt/map1
bin dev home lib64 media opt root sbin sys usr
boot etc lib lost+found mnt proc run srv tmp var
```

- c. Use the `umount` command to unmount `/mnt/map1`.

```
[root@gateway stage]# umount /mnt/map1/
```

7. Mount the `loop0p3` device created by the `kpartx` command.

- a. Use the `mount` command to mount `/dev/mapper/loop0p3` on `/mnt/map1`.

```
[root@gateway stage]# mount /dev/mapper/loop0p3 /mnt/map1
```

- b. Use the `ls` command to view the files on `/mnt/map1`.

- Note that these are the files that would be in the `/home` directory on an Oracle Linux system.

```
[root@gateway stage]# ls /mnt/map1
lost+found oracle
```

- c. Use the `umount` command to unmount `/mnt/map1`.

```
[root@gateway stage]# umount /mnt/map1/
```

8. Mount the `loop0p4` device created by the `kpartx` command.

- Recall that the `loop0p4` device represents partition 4, which is not mounted.

- a. Use the `mount` command to mount `/dev/mapper/loop0p4` on `/mnt/map1`.

- This command fails as expected.

```
[root@gateway stage]# mount /dev/mapper/loop0p4 /mnt/map1
mount: wrong fs type, bad option, bad superblock on
/dev/mapper/loop0p4,
 missing codepage or helper program, or other error
```

In some cases useful info is found in syslog - try  
`dmesg | tail` or so.

9. Mount the `loop0p5` device created by the `kpartx` command.

- Recall that the `loop0p5` device represents a swap partition.

- a. Use the `mount` command to mount `/dev/mapper/loop0p5` on `/mnt/map1`.
  - This command fails as expected.

```
[root@gateway stage]# mount /dev/mapper/loop0p5 /mnt/map1
mount: unknown filesystem type 'swap'
```

10. Remove the `kpartx` device mapping on **gateway**.

- a. Use the `rmdir` command to delete `/mnt/map1`.

```
[root@gateway stage]# rmdir /mnt/map1
```

- b. Use the `kpartx -d` command to disconnect the device.

```
[root@gateway stage]# kpartx -d system.img
loop deleted : /dev/loop0
```

- c. Use the `ls` command to list the contents of `/dev/mapper`.

- Note that the device mapping no longer exists in `/dev/mapper`.

```
[root@gateway stage]# ls /dev/mapper
control
```

## **Practices for Lesson 14: File Sharing**

## Practices for Lesson 14: Overview

---

### Overview

In these practices, you:

- Configure an NFS server and client and mount an exported file system
- Configure and use an FTP server

## Practice 14-1: Configuring an NFS Server and an NFS Client

---

### Overview

In this practice, you:

- Export a file system from the **host03** VM and mount it on the **host02** VM
- Ensure that required packages are installed and that services are running
- Use various NFS-related commands and files to share file systems using NFS

### Assumptions

- You are logged on to **gateway** as the **root** user.
- This practice is performed on the **host02** and **host03** VMs.
- The prompts include either **host02** or **host03** to indicate which system to enter the command from.

### Tasks

#### 1. Log in to **host03**.

- Use the `ssh` command to log in to **host03** as the **root** user.

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

#### 2. Create a new partition on `/dev/xvdc`.

##### a. Use the `fdisk` command to display the available devices on your system.

- Note that there are no partitions on `/dev/xvdc` and `/dev/xvdd`.

```
fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
 /dev/xvda1 * 2048 2099199 1048576 83 Linux
 /dev/xvda2 2099200 41943039 19921920 8e Linux
 LVM

Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

##### b. Use the `parted` `mkpart` command to create a new primary partition on `/dev/xvdc`.

- The new partition uses all the space on the device.
- If you get the error shown, “Error: `/dev/xvdc`: unrecognised disk label”, run the `mktabe msdos` command as shown, then continue with the `mkpart` command,

```
parted /dev/xvdc
```

```
GNU Parted 3.1
```

```
Using /dev/xvdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Error: /dev/xvdc: unrecognised disk label
(parted) mktable msdos
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? ENTER
Start? 0
End? 100%
Warning: The resulting partition is not properly aligned for
best performance.
Ignore/Cancel? i
(parted) quit
Information: You may need to update /etc/fstab.
```

- c. Use the `fdisk` command to display the partition table on all available devices.

```
fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvda1 * 2048 2099199 1048576 83 Linux
/dev/xvda2 2099200 41943039 19921920 8e Linux
LVM
Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvdc1 1 41943039 20971519+ 83 Linux
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

- Note that the `/dev/xvdc1` partition is shown.

3. Create a new partition on `/dev/xvdd`.

- a. Use the `parted` `mkpart` command to create a new primary partition on `/dev/xvdd`.
- The new partition uses all the space on the device.
  - If you get the error shown, “Error: `/dev/xvdd`: unrecognised disk label”, run the `mktable msdos` command as shown, then continue with the `mkpart` command,

```
parted /dev/xvdd
GNU Parted 3.1
Using /dev/xvdd
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Error: /dev/xvdd: unrecognised disk label
(parted) mktable msdos
```

```
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? ENTER
Start? 0
End? 100%
Warning: The resulting partition is not properly aligned for
best performance.
Ignore/Cancel? i
(parted) quit
Information: You may need to update /etc/fstab.
```

- b. Use the `fdisk` command to display the partition table on all available devices.

```
fdisk -l | grep /dev
Disk /dev/xvda: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvda1 * 2048 2099199 1048576 83 Linux
/dev/xvda2 2099200 41943039 19921920 8e Linux
LVM
Disk /dev/xvdb: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/xvdc: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvdc1 1 41943039 20971519+ 83 Linux
Disk /dev/xvdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
/dev/xvdd1 1 41943039 20971519+ 83 Linux
Disk /dev/mapper/ol-root: 18.2 GB, 18249416704 bytes, 35643392
sectors
Disk /dev/mapper/ol-swap: 2147 MB, 2147483648 bytes, 4194304
sectors
```

- Note that the `/dev/xvdd1` partition is shown.

4. Create and mount file systems on `/dev/xvdc1` and `/dev/xvdd1`.

- a. Use the `mkfs` command to make an ext4 file system on `/dev/xvdc1`.

```
[root@host03 ~]# mkfs -t ext4 /dev/xvdc1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 inodes, 5242879 blocks
262143 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2153775104
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
```

```
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736,
 1605632, 2654208,
 4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- b. Use the `mkfs` command to make an ext4 file system on `/dev/xvdd1`.

```
[root@host03 ~]# mkfs -t ext4 /dev/xvdd1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 inodes, 5242879 blocks
262143 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2153775104
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736,
 1605632, 2654208,
 4096000
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- c. Use the `mkdir` command to create `/Test` and `/Dev` mount points

```
[host03]# mkdir /Test /Dev
```

- d. Use the `vi` editor to add the following entries to the bottom of `/etc/fstab`.

```
[host03]# vi /etc/fstab
/dev/xvdc1 /Test ext4 defaults 0 0
/dev/xvdd1 /Dev ext4 defaults 0 0
```

- e. Use the `mount -a` command to mount all the file systems defined in `/etc/fstab`.

```
[host03]# mount -a
```

- f. Use the `df` command to display the mounted file system on `/dev/xvdc1`.

```
[host03]# df -h
Filesystem Size Used Avail Use% Mounted on
...
/dev/xvdc1 20G 45M 19G 1% /Test
/dev/xvdd1 20G 45M 19G 1% /Dev
```

- This shows `/dev/xvdc1` is mounted on `/Test`.
- This shows `/dev/xvdd1` is mounted on `/Dev`.

5. Export an NFS file system on **host03**.

- a. Use the `vi` editor and edit `/etc/exports` to export `/Test` to all client systems.

```
[host03]# vi /etc/exports
/Test *
```

- b. Use the `yum` command to install the `rpcbind` package.

```
[host03]# yum install rpcbind
...
Is this ok [y/d/N]: y
...
Complete!
```

- c. Use the `systemctl` command to start and enable the `rpcbind` service.

```
[host03]# systemctl start rpcbind
[host03]# systemctl enable rpcbind
```

- d. Use the `yum` command to install the `nfs-utils` package.

```
[host03]# yum install nfs-utils
...
Is this ok [y/d/N]: y
...
Complete!
```

- e. Use the `systemctl` command to start and enable the `nfs` service.

```
[host03]# systemctl start nfs
[host03]# systemctl enable nfs
```

- f. Use the `systemctl` command to start and enable the `nfslock` service.

```
[host03]# systemctl start nfslock
[host03]# systemctl enable nfslock
```

- g. Use the `showmount -e` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Test *
```

- If the exported file system is not listed, restart the `nfs` service.

- Whenever a new entry is made to /etc/exports, restart the nfs service.
6. Configure an NFS server from the command line.
- Use the exportfs command to export /Dev to all clients and allow read/write permission.
    - Include the -i option to ignore /etc/exports entries.

```
[host03]# exportfs -i -o rw *:/Dev
```

- Use the showmount command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Test *
/Dev *
```

- Note that both exported file systems are listed.
  - You do not need to restart the nfs service when using exportfs.
- Use the cat command to view the contents of /var/lib/nfs/etab.

```
[host03]# cat /var/lib/nfs/etab
/Test
 *(ro,sync,wdelay,hide,nocrossmnt,secure,root_squash,no_all_
squash,no_subtree_check,secure_locks,acl,no_pnfs,anonuid=65534,a
nongid=65534,sec=sys,ro,secure,root_squash,no_all_squash)
/Dev
 *(rw,sync,wdelay,hide,nocrossmnt,secure,root_squash,no_all_
squash,no_subtree_check,secure_locks,acl,no_pnfs,anonuid=65534,a
nongid=65534,sec=sys,rw,secure,root_squash,no_all_squash)
```

- Note that both exported file systems are listed in this master export table.
  - The rpc.mountd process reads this file when a client attempts to mount an NFS file system.
- Use the vi editor to edit /etc/exports.

- Change the entry exporting /Test to the following:

```
[host03]# vi /etc/exports
/Test *(rw,no_root_squash)
```

- The rw option allows client systems to make changes to the file system.
  - The no\_root\_squash option allows root users on client systems to retain root privileges on the file system.
- Run the exportfs -r command on host03.

```
[host03]# exportfs -r
```

- This command re-exports the entries in /etc/exports and synchronizes /var/lib/nfs/etab with /etc/exports.
- Use the cat command to view the contents of /var/lib/nfs/etab.

```
[host03]# cat /var/lib/nfs/etab
/Test
 *(rw,sync,wdelay,hide,nocrossmnt,secure,no_root_squash,no_a
```

```
11_squash,no_subtree_check,secure_locks,acl,no_pnfs,anonuid=6553
4,anongid=65534,sec=sys,rw,secure,no_root_squash,no_all_squash)
```

- Note the new options on the /Test NFS share.
- Also note that /Dev is no longer listed.

g. Use the `showmount -e` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Test *
```

- This confirms that the /Dev file system is no longer exported.

7. Log in to **host02**. Do not log off from **host03**.

- Open a second terminal window on **gateway**.
- From the second terminal window on **gateway**, use the `ssh` command to log in to **host02** as the `root` user.

```
[gateway]$ ssh root@host02
root@host02's password:
Last login: ...
```

8. Mount the exported /Test file system on **host02**.

- Use the `mkdir` command to create a mountpoint named `/remote_dev` on **host02**.

```
[host02]# mkdir /remote_dev
```

- Use the `mount` command to mount the exported file system from **host03**, /Test, with `rw` and `nosuid` options on the local mountpoint, `/remote_dev`.
  - The `rw` option mounts the file system with read/write permissions.
  - The `nosuid` option does not allow `setuid` or `setgid` bits to take effect.

```
[host02]# mount -t nfs -o rw,nosuid host03:/Test /remote_dev
mount.nfs: no route to host
```

- In this example, the `mount` command fails.
- It takes some time for the `mount` command to fail. Rather than wait for the command to fail, you can press `Ctrl + C` to abort the `mount` command.
- If the `mount` command fails, use the `systemctl` command to stop `firewalld` on **host03**.
  - For the purposes of this practice, stop `firewalld` on **host03** to allow **host02** to mount the exported file system from **host03**.

```
[host03]# systemctl stop firewalld
```

- `firewalld` and other system security-related issues are covered in the another course.

d. Re-issue the `mount` command from **host02**, if it is necessary to mount the NFS share.

```
[host02]# mount -t nfs -o rw,nosuid host03:/Test /remote_dev
```

- In this example, the `mount` command is successful.

e. Run the `mount` command to view the mount information for the NFS share.

```
[host02]# mount | grep nfs
...
```

```
host03:/Test on /remote_dev type nfs4
(rw,nosuid,relatime,vers=4.1,rsize=524288,wszie=524288,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.0.2.102,local_lock=none,addr=192.0.2.103)
```

- Note the mount options (rw,nosuid), the NFS version (4.1), and the server and client IP addresses.
9. Verify that the NFS file system is mounted with `rw` (read/write) permissions.
- a. On **host02**, use the `df` command to display the mounted file systems.

```
[host02]# df -h
Filesystem Size Used Avail Use% Mounted on
...
host03:/Test 20G 44M 19G 1% /remote_dev
```

- Note that the `host03:/Test` file system is mounted on the local file system `/remote_dev`.
- b. Use the `ls` command to list the contents of `/remote_dev` on **host02**.

```
[host02]# ls /remote_dev
lost+found
```

- c. Use the `ls` command to list the contents of `/Test` on **host03**.

```
[host03]# ls /Test
lost+found
```

- Note that the contents of `/Test` on **host03** are the same as `/remote_dev` on **host02** because they are the same directories.
- d. From **host03**, use the `vi` command to create a file on `/Test`. Enter some content in the file.

```
[host03]# vi /Test/test
<enter some content>
```

- e. From **host02**, use the `vi` command to edit the file created from **host03**. Make some changes to the content.

```
[host02]# vi /remote_dev/test
<change the content>
```

- f. From **host03**, use the `cat` command to view the contents of the file in `/Test`.

```
[host03]# cat /Test/test
<content has changed>
```

- This again confirms `/Test` on **host03** is the same as `/remote_dev` on **host02** and that the file system has read/write permissions.

## Practice 14-2: Removing the NFS Configuration

---

### Overview

In this practice, you remove the NFS configuration you set up previously, in preparation for the next practice.

### Assumptions

- You are the `root` user on **host02** and **host03**.
- The prompts include either **host02** or **host03** to indicate which system to enter the command from.

### Tasks

1. From **host02**, unmount the exported file systems.

- From **host03**, `/Test` is mounted on `/remote_dev`.
  - Use the `df` command to display the mounted file systems.
  - Note that the `host03:/Test` filesystem is mounted on `/remote_dev`.

```
[host02]# df -h
Filesystem Size Used Avail Use% Mounted on
...
host03:/Test 20G 44M 19G 1% /remote_dev
```

- Use the `umount` command to unmount `/remote_dev`.

- Use the `cd` command to ensure you are not in `/remote_dev` before unmounting.

```
[host02]# cd
[host02]# umount /remote_dev
```

- Use the `rmdir` command to remove `/remote_dev`.

```
[host02]# rmdir /remote_dev
```

- Use the `df` command to verify this file system is unmounted.

```
[host02]# df -h
Filesystem Size Used Avail Use% Mounted on
...
```

- You can see that `/remote_dev` has been unmounted.

2. From **host03**, stop the `nfs` service and remove changes to `/etc/exports`.

- Use the `systemctl` command to stop the `nfs` service.

```
[host03]# systemctl stop nfs
```

- Use the `systemctl` command to stop the `nfs` service from automatically starting at boot time.

```
[host03]# systemctl disable nfs
Removed symlink /etc/systemd/system/multi-user.target.wants/nfs-
server.service.
```

- Use the `vi` editor and edit `/etc/exports` on **host03** to delete the entry.

```
[host03]# vi /etc/exports
/test *(rw,no_root_squash)
```

3. From **host03**, unmount the file systems mounted on **/Dev** and **/Test**.

- a. Use the `umount` command to unmount the file systems on **/Dev** and **/Test**.

```
[host03]# umount /Dev /Test
```

- b. Use the `vi` editor to remove the following entries from `/etc/fstab`.

```
[host03]# vi /etc/fstab
/dev/xvdc1 /Test ext4 defaults 0 0
/dev/xvdd1 /Dev ext4 defaults 0 0
```

- c. Use the `rmdir` command to remove **/Dev** and **/Test**.

```
[host03]# rmdir /Dev /Test
```

4. Remove the `/dev/xvdc1` and `/dev/xvdd1` partitions.

- a. Use the `parted` command to remove the `/dev/xvdc1` partition.

```
[host03]# parted /dev/xvdc
GNU Parted 3.1
Using /dev/xvdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdc: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
1 512B 21.5GB 21.5GB primary ext4

(parted) rm 1
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdc: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags

(parted) quit
Information: You may need to update /etc/fstab.
```

- b. Use the `parted` command to remove the `/dev/xvdd1` partition.

```
[host03]# parted /dev/xvdd
GNU Parted 3.1
Using /dev/xvdd
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Partition Flags:

Number Start End Size Type File system Flags
 1 512B 21.5GB 21.5GB primary ext4

(parted) rm 1
(parted) quit
Information: You may need to update /etc/fstab.
```

- c. View the /proc/partitions file to show the partitions no longer exist.

```
[host03]# cat /proc/partitions
major minor #blocks name

 202 0 20971520 xvda
 202 1 1048576 xvda1
 202 2 19921920 xvda2
 202 16 4194304 xvdb
 202 32 20971520 xvdc
 202 48 20971520 xvdd
 249 0 17821696 dm-0
 249 1 2097152 dm-1
```

## Practice 14-3: Configuring an FTP Server

---

### Overview

In this practice, you:

- Install the `vsftpd` server package on **host03** and start the service
- Install the `ftp` (client) package and test the setup

### Assumptions

You are the `root` user on **host03** VM.

### Tasks

1. Install and start `vsftpd` on **host03**.

- a. Use the `yum` command to install the `vsftpd` package on **host03**.

```
yum install vsftpd
...
Is this ok [y/N]: y
...
Complete!
```

- b. Use the `systemctl` command to start the `vsftpd` service.

```
systemctl start vsftpd
```

- c. Use the `systemctl` command to configure `vsftpd` to start at boot time.

```
systemctl enable vsftpd
Created symlink from /etc/systemd/system/multi-
user.target.wants/vsftpd.service to
/usr/lib/systemd/system/vsftpd.service.
```

2. Use the `yum` command to install the `ftp` (client) package on **host03**.

- Answer ‘y’ when prompted.

```
yum install ftp
...
Is this ok [y/N]: y
...
Complete!
```

3. Test the setup.

- a. Use the `ftp` command to connect to `localhost` and log in as anonymous.
- Use any password.
- b. After connecting, run the `ls` command to display the contents of `/var/ftp`.
- c. Conclude the test by running the `quit` command to exit.

```
ftp localhost
Connected to localhost (127.0.0.1).
```

```

220 (vsFTPd 3.0.2)
Name (localhost:root): anonymous
331 Please specify the password.
Password: your_email_address
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (127,0,0,1,179,34).
150 Here comes the directory listing.
drwxr-xr-x 2 0 0 4096 May 27 06:26 pub
226 Directory send OK.
ftp> quit
221 Goodbye.

```

- Note that the contents of /var/ftp is a pub directory for anonymous users.

- d. Use the grep command to display the ftp user information in /etc/passwd.

```

grep ftp /etc/passwd
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

```

- In this example, the home directory of the ftp user is /var/ftp.

- e. Ensure that the permissions on the home directory are set to 755. Change the settings if necessary.

```

ls -ld /var/ftp
drwxr-xr-x 3 root root 4096 <date_time> /var/ftp

```

- In this example, the permissions are correct.

- f. Copy the /root/anaconda-ks.cfg file to /var/ftp/pub and rename it as test\_file.

```

cp /root/anaconda-ks.cfg /var/ftp/pub/test_file

```

- This file is used in the next practice.

- g. Use the chmod command to change the permissions on /var/ftp/pub/test\_file to 666.

- Use the ls -l command to view the permissions after making the change.

```

chmod 666 /var/ftp/pub/test_file
ls -l /var/ftp/pub/test_file
-rw-rw-rw- ... /var/ftp/pub/test_file

```

- In this example, the permissions are correct.

## Practice 14-4: Downloading a File from an FTP Server

### Overview

In this practice, you install the `ftp` package on the **host02** VM and download a file from the FTP server on **host03**.

### Assumptions

You are the `root` user on the **host02** VM.

### Tasks

1. Use the `yum` command to install the `ftp` (client) package on **host02**.
2. Answer ‘y’ when prompted.

```
yum install ftp
...
Is this ok [y/N]: y
...
Complete!
```

3. Download a file using `ftp` to **host02** from the FTP server, **host03**.
  - a. From **host02**, use the `cd` command to change back to your home directory.
  - b. Use the `ftp` utility to connect to the FTP server, **host03**, as the `anonymous` user.
    - Use any password.

```
cd
pwd
/root
b. # ftp host03
Connected to host03 (192.0.2.103).
220 (vsFTPd 3.0.2)
Name (host03:root): anonymous
331 Please specify the password.
Password: your_email_address
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

- c. Enter `help` or `?` to display a list of available commands.

```
ftp> help
...
```

- d. Get help on some of the available commands, for example:

```
ftp> help get
get receive file
```

```

ftp> help put
put send one file
ftp> help mget
mget get multiple files
ftp> help cd
cd change remote working directory
ftp> help ls
ls list contents of remote directory
ftp> help !
!
escape to the shell

```

- e. Use the `cd` command to change to the `pub` directory.

```

ftp> cd pub
Directory successfully changed.

```

- f. Use the `ls` command to list the contents of the `/var/ftp/pub` directory on the FTP server.

```

ftp> ls
227 Entering Passive Mode (192,0,2,103,174,14).
150 Here comes the directory list.
-rw-rw-rw- 1 0 0 1765 <date_time> test_file
226 Directory send OK.

```

- Note that `test_file` exists in the `pub` directory on the FTP server.

- g. Use the `!ls` command to list the contents of the local directory on `host02`.

```

ftp> !ls
anaconda-ks.cfg Documents initial-setup.cfg Pictures
Desktop Downloads Music Public
...

```

- h. Use the `get` command to download the `test_file` file from the FTP server.

```

ftp> get test_file
local: test_file remote: test_file
227 Entering Passive Mode (192,0,2,103,38,129).
150 Opening BINARY mode data connection for test_file ...
226 Transfer complete.
2310 bytes received in ...

```

- i. Use the `!ls` command to list the contents of the local directory on `host02`.

```

ftp> !ls
anaconda-ks.cfg Documents initial-setup.cfg Pictures
Desktop Downloads Music Public
...
test_file

```

- Note that the `test_file` exists in the local directory on `host02`.

- j. Use the `quit` command to exit `ftp`.

```
ftp> quit
221 Goodbye.
```

## Practice 14-5: Restoring VMs to Original Configuration

### Overview

In this practice, you remove `ftp` and the `vsftpd` service you configured previously, in preparation for the next practice.

### Assumptions

You are the `root` user on **host02** and **host03**.

### Tasks

1. Use the `yum` command to remove the `ftp` package on **host02**.

- Answer `y` to “Is this ok.”

```
[host02]# yum remove ftp
...
Is this ok [y/N]: y
...
Complete!
```

2. Use the `rm` command to remove `test_file` from the root user’s home directory.

```
[host02]# cd
[host02]# pwd
/root
[host02]# rm test_file
rm: remove regular file 'test_file'? y
```

3. Log off from **host02** in preparation for the next practice.

- a. Use the `logout` command to close the `ssh` connection to **host02**.

```
[host02]# logout
Connection to host02 closed.
```

All remaining commands in this practice are run from the **host03** VM.

4. Remove the FTP configuration on **host03**.

- a. Use the `service` command to view the status of the `vsftpd` service.

```
[host03]# systemctl status vsftpd
vsftpd.service - Vsftpd ftp daemon
 Loaded: loaded (/usr/lib/systemd/system/vsftpd.service;
 enabled; vendor preset: disabled)
 Active: active (running) since ...
 ...

```

- Note that the `vsftpd` service is active (running) and enabled.
- b. Use the `systemctl` command to stop the `vsftpd` service and disable it so that it does not start automatically at boot time.

```
[host03]# systemctl stop vsftpd
[host03]# systemctl disable vsftpd
Removed symlink /etc/systemd/system/multi-
user.target.wants/vsftpd.service.
```

- c. Use the `yum` command to remove the `ftp` package on **host03**.

- Answer `y` to “Is this ok.”

```
[host03]# yum remove ftp
...
Is this ok [y/N]: y
...
Complete!
```

5. Use the `rm` command to remove the `/var/ft/pub/test_file`.

```
[host03]# rm /var/ftp/pub/test_file
rm: remove regular file '/var/ftp/pub/test_file'? y
```

6. Start the `firewalld` service on **host03** by running the following command:

```
[host03]# systemctl start firewalld
```

7. Log off from **host03**.

Use the `logout` command to close the `ssh` connection to **host03**.

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

## **Practices for Lesson 15: Kerberos and IPA Services**

## Practices for Lesson 15: Overview

---

### Overview

In these practices, you configure:

- Kerberos server and create administrative users
- IPA server and IPA Client
- SSSD Services

## Practice 15-1: Kerberos Authentication

---

### Overview

In this practice you:

- Verify installation of the Kerberos server packages
- Configure the Kerberos server

### Assumptions

- This practice is performed on **host06** VM.
- You are the `root` user on **gateway**.

### Tasks

1. Log in to the **host06** VM guest from **gateway**.
  - Use the `ssh` command to log in to **host06**.
  - Enter the password for `root`.

```
[gateway]# ssh host06
root@host06's password:
Last login: ...
[root@host06 ~]#
```

2. Verify the kerberos server, workstation, and library packages are installed.
  - a. Use the `rpm` command, pipe the output to the `grep` command, and search for the string “kerberos”.
    - The `python-kerberos` package is already installed.

```
rpm -qa | grep kerberos
python-kerberos-1.1-15.el7.x86_64
```

- b. Use the `rpm` command, pipe the output to the `grep` command, and search for the string “krb5”.
  - The `krb5-workstation` package is already installed.
  - The `krb5-libs` package is already installed.
  - The `krb5-server` package is already installed.

```
rpm -qa | grep krb5
sssd-krb5-common-1.16.2-13.el7.x86_64
krb5-workstation-1.15.1-34.el7.x86_64
sssd-krb5-1.16.2-13.el7.x86_64
krb5-libs-1.15.1-34.el7.x86_64
krb5-pkinit-1.15.1-34.el7.x86_64
krb5-server-1.15.1-34.el7.x86_64
```

3. Use the `vi` editor to edit `/etc/krb5.conf` and configure settings for the Kerberos realm:
  - You can edit the `krb5.conf` file as follows by using the `vi` command.

- Uncomment to define the example.local domain realm.
- You also define the kerberos key distribution center and kerberos server hostnames.
- Add pam debugging and validation settings.
- Rather than edit the /etc krb5.conf file as shown, an edited file exists on **gateway** in the /stage directory. You can use the scp command to copy the file from /stage on **gateway** to /etc on **host06**.

```
vi /etc/krb5.conf
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = EXAMPLE.LOCAL
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
EXAMPLE.LOCAL = {
 kdc = krbsvr.example.local
 admin_server = krbsvr.example.local
}

[domain_realm]
.example.local = EXAMPLE.LOCAL
example.local = EXAMPLE.LOCAL

[appdefaults]
pam = {
 debug = true
 validate = false
}
```

4. Use the vi editor to edit /var/kerberos/krb5kdc/kdc.conf for configuring the key distribution center:
  - You can edit the kdc.conf file as follows by using the vi command.
  - You change the encryption type for the master key.

- You add a flag for preauthorization.
- Rather than edit the `/var/kerberos/krb5kdc/kdc.conf` file as shown, an edited file exists on **gateway** in the `/stage` directory. You can use the `scp` command to copy the file from `/stage` on **gateway** to `/var/kerberos/krb5kdc` on **host06**.

```
vi /var/kerberos/krb5kdc/kdc.conf
kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
EXAMPLE.LOCAL = {
 #master_key_type = aes256-cts
master_key_type = des-hmac-sha1
default_principal_flags = +preauth
 acl_file = /var/kerberos/krb5kdc/kadm5.acl
 dict_file = /usr/share/dict/words
 admin_keytab = /etc/kadm5.keytab
 supported_enctypes = aes256-cts:normal aes128-cts:normal des3-
hmac-sha1:normal \
 arcfour-hmac:normal des-hmac-sha1:normal des-cbc-md5:normal
des-cbc-crc:normal
}
```

5. Create the Kerberos database and store the database password in a stash file:

- The `create` command creates the database for storing keys for the `example.local` kerberos realm.
- The `-s` creates the stash file of the master key.
- You use a password or “`oracle`” for the master key.

```
/usr/sbin/kdb5_util create -s
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for
realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: oracle
Re-enter KDC database master key to verify: oracle
```

6. Use the `vi` editor to edit `/var/kerberos/krb5kdc/kadm5.acl` and define those principals with administrative access to the Kerberos database.

- This configuration allows users with the `admin` principal to gain full administrative rights of the Kerberos database.
- Change `.COM` to `.local` as shown:

```
vi /var/kerberos/krb5kdc/kadm5.acl
*/admin@EXAMPLE.local *
```

7. Use the kadmin.local command to create a principal for each user who should have rights over the admin instance:

- Enter a password of “oracle” for the principal.

```
kadmin.local -q "addprinc root/admin"
Authenticating as principal root/admin@EXAMPLE.COM with
password.

WARNING: no policy specified for root/admin@EXAMPLE.COM;
defaulting to no policy
Enter password for principal "root/admin@EXAMPLE.COM": oracle
Re-enter password for principal "root/admin@EXAMPLE.COM": oracle
Principal "root/admin@EXAMPLE.COM" created.
```

8. Cache the keys that kadmin uses to decrypt administration Kerberos tickets in /etc/kadm5.keytab.

- Run the following command using the kadmin/admin argument as shown:

```
kadmin.local -q "ktadd -k /etc/kadm5.keytab kadmin/admin"
Authenticating as principal root/admin@EXAMPLE.LOCAL with
password.

Entry for principal kadmin/admin with kvno 2, encryption type
aes256-cts-hmac-sha1-96 added to keytab
WRFILE:/etc/kadm5.keytab.

Entry for principal kadmin/admin with kvno 2, encryption type
aes128-cts-hmac-sha1-96 added to keytab
WRFILE:/etc/kadm5.keytab.
```

- Run the command again but use the kadmin/changepw argument.

```
kadmin.local -q "ktadd -k /etc/kadm5.keytab kadmin/changepw"
Authenticating as principal root/admin@EXAMPLE.LOCAL with
password.

Entry for principal kadmin/changepw with kvno 2, encryption type
aes256-cts-hmac-sha1-96 added to keytab
WRFILE:/etc/kadm5.keytab.

Entry for principal kadmin/changepw with kvno 2, encryption type
aes128-cts-hmac-sha1-96 added to keytab
WRFILE:/etc/kadm5.keytab.
```

9. Start the key distribution center and administration services:

```
systemctl start krb5kdc
systemctl start kadmin
```

10. Enable the key distribution center and administration service to start at boot up:

```
systemctl enable krb5kdc
Created symlink from /etc/systemd/system/multi-
user.target.wants/krb5kdc.service to
/usr/lib/systemd/system/krb5kdc.service.
```

```
systemctl enable kadmin
Created symlink from /etc/systemd/system/multi-
user.target.wants/kadmin.service to
/usr/lib/systemd/system/kadmin.service.
```

11. Add other principals for users using the kadmin.local command.

- c. Run the following command.
- You add a password of "oracle" for the oracle principle.

```
kadmin.local -q "addprinc oracle"
Authenticating as principal root/admin@EXAMPLE.COM with
password.

WARNING: no policy specified for oracle@EXAMPLE.COM; defaulting
to no policy

Enter password for principal "oracle@EXAMPLE.COM": oracle
Re-enter password for principal "oracle@EXAMPLE.COM": oracle
Principal "oracle@EXAMPLE.COM" created.
```

- d. Run the following command.

```
kadmin.local -q "addprinc -randkey host/krbsvr.example.local"
Authenticating as principal root/admin@EXAMPLE.COM with
password.

WARNING: no policy specified for
host/krbsvr.example.local@EXAMPLE.COM; defaulting to no policy

Principal "host/krbsvr.example.local@EXAMPLE.COM" created.
```

- e. Run the following command.

```
kadmin.local -q "ktadd -k /etc/kadm5.keytab
host/krbsvr.example.local"
Authenticating as principal root/admin@EXAMPLE.LOCAL with
password.

Entry for principal host/krbsvr.example.local with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab
WRFILE:/etc/kadm5.keytab.

Entry for principal host/krbsvr.example.local with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab
WRFILE:/etc/kadm5.keytab.
```

## Practice 15-2: Installing Identity Management

---

### Overview

In this practice, you:

- Configure the IPA server
- Log in to the Identity Management Console
- Take a quick tour of the Identity Management server

### Assumptions

- You are the `root` user on **host06**.

### Tasks

1. Use `vi` and edit `/etc/hosts`.

- Add host names for the kerberos and IPA servers.

```
vi /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6
192.0.2.106 host06.example.local kdc.example.local
krbsvr.example.local ipa.example.local ipa
```

2. Verify the `ipa-server`, `bind` and `ldap` related packages are installed.

- Use the `rpm` command, pipe the output to the `grep` command, and search for the string “ipa”.
  - The `ipa-server` package is already installed.
  - The `ipa-server-dns` package is already installed.
  - The `ipa-server-common` package is already installed.

```
rpm -qa | grep ipa-server
ipa-server-4.6.4-10.0.1.el7.x86_64
ipa-server-dns-4.6.4-10.0.1.el7.noarch
ipa-server-common-4.6.4-10.0.1.el7.noarch
```

- Use the `rpm` command, pipe the output to the `grep` command, and search for the string “bind”.
  - The `bind` package is already installed.

```
rpm -qa | grep bind
...
bind-9.9.4-72.el7.x86_64
...
```

- Use the `rpm` command, pipe the output to the `grep` command, and search for the string “bind-dyndb-ldap”.

- The bind-dyndb-ldap package is already installed.

```
rpm -qa | grep bind-dyndb-ldap
bind-dyndb-ldap-11.1-4.el7.x86_64
```

3. Run the setup script ipa-server-install for configuring the IPA server with DNS.

a. Run the command with the --setup-dns option as shown.

- Enter a password with at least 8 characters when prompted.
  - Use "oracleipa" for the password as shown.
- For this exercise, use the same password for the Directory Manager and IPA admin.

```
/usr/sbin/ipa-server-install --setup-dns
```

The log file for this installation can be found in  
 /var/log/ipaserver-install.log

=====

This program will set up the IPA Server.

This includes:

- Configure a stand-alone CA (dogtag) for certificate management
- Configure the Network Time Daemon (ntpd)
- Create and configure an instance of Directory Server
- Create and configure a Kerberos Key Distribution Center (KDC)
  - Configure Apache (httpd)
  - Configure DNS (bind)
  - Configure the KDC to enable PKINIT

To accept the default shown in brackets, press the Enter key.

WARNING: conflicting time&date synchronization service 'chronyd'  
 will be disabled in favor of ntpd

Enter the fully qualified domain name of the computer  
 on which you're setting up server software. Using the form  
 <hostname>.<domainname>

Example: master.example.com.

Server host name [host06.example.local]: **ENTER**

Warning: skipping DNS resolution of host host06.example.local  
 The domain name has been determined based on the host name.

```
Please confirm the domain name [example.local]: ENTER
```

The kerberos protocol requires a Realm name to be defined.  
This is typically the domain name converted to uppercase.

```
Please provide a realm name [EXAMPLE.LOCAL]: ENTER
```

Certain directory server operations require an administrative user.

This user is referred to as the Directory Manager and has full access to the Directory for system management tasks and will be added to the instance of directory server created for IPA.

The password must be at least 8 characters long.

```
Directory Manager password: oracleipa
```

```
Password (confirm): oracleipa
```

The IPA server requires an administrative user, named 'admin'.  
This user is a regular system account used for IPA server administration.

```
IPA admin password: oracleipa
```

```
Password (confirm): oracleipa
```

Checking DNS domain example.local., please wait ...

```
Do you want to configure DNS forwarders? [yes]: no
```

No DNS forwarders configured

```
Do you want to search for missing reverse zones? [yes]: ENTER
```

```
Do you want to create reverse zone for IP 192.0.2.106 [yes]:
ENTER
```

```
Please specify the reverse zone name [2.0.192.in-addr.arpa.]:
ENTER
```

Using reverse zone(s) 2.0.192.in-addr.arpa.

The IPA Master Server will be configured with:

Hostname: host06.example.local

IP address(es): 192.0.2.106

Domain name: example.local

Realm name: EXAMPLE.LOCAL

BIND DNS server will be configured to serve IPA domain with:

Forwarders: No forwarders

Forward policy: only

Reverse zone(s): 2.0.192.in-addr.arpa.

```
Continue to configure the system with these values? [no]: yes

The following operations may take some minutes to complete.
Please wait until the prompt is returned.

Configuring NTP daemon (ntpd)
[1/4]: stopping ntpd
[2/4]: writing configuration
[3/4]: configuring ntpd to start on boot
[4/4]: starting ntpd
Done configuring NTP daemon (ntpd).

Configuring directory server (dirsrv). Estimated time: 30
seconds
[1/44]: creating directory server instance
...
[44/44]: configuring directory to start on boot
Done configuring directory server (dirsrv).

Configuring Kerberos KDC (krb5kdc)
[1/10]: adding kerberos container to the directory
...
[10/10]: configuring KDC to start on boot
Done configuring Kerberos KDC (krb5kdc).

Configuring kadmin
[1/2]: starting kadmin
[2/2]: configuring kadmin to start on boot
Done configuring kadmin.

Configuring ipa-custodia
[1/5]: Making sure custodia container exists
[2/5]: Generating ipa-custodia config file
[3/5]: Generating ipa-custodia keys
[4/5]: starting ipa-custodia
[5/5]: configuring ipa-custodia to start on boot
Done configuring ipa-custodia.

Configuring certificate server (pki-tomcatd). Estimated time: 3
minutes
[1/28]: configuring certificate server instance
...
[28/28]: configuring certmonger renewal for lightweight CAs
Done configuring certificate server (pki-tomcatd).

Configuring directory server (dirsrv)
[1/3]: configuring TLS for DS instance
[2/3]: adding CA certificate entry
[3/3]: restarting directory server
```

```
Done configuring directory server (dirsrv) .
Configuring ipa-otpd
[1/2]: starting ipa-otpd
[2/2]: configuring ipa-otpd to start on boot
Done configuring ipa-otpd.
Configuring the web interface (httpd)
[1/22]: stopping httpd
...
[22/22]: enabling oddjobd
Done configuring the web interface (httpd) .
Configuring Kerberos KDC (krb5kdc)
[1/1]: installing X509 Certificate for PKINIT
Done configuring Kerberos KDC (krb5kdc) .
Applying LDAP updates
Upgrading IPA:.. Estimated time: 1 minute 30 seconds
[1/10]: stopping directory server
[2/10]: saving configuration
[3/10]: disabling listeners
[4/10]: enabling DS global lock
[5/10]: disabling Schema Compat
[6/10]: starting directory server
[6/10]: starting directory server
[7/10]: upgrading server
[8/10]: stopping directory server
[9/10]: restoring configuration
[10/10]: starting directory server
Done.
Restarting the KDC
Configuring DNS (named)
[1/12]: generating rndc key file
...
[12/12]: changing resolv.conf to point to ourselves
Done configuring DNS (named) .
Restarting the web server to pick up resolv.conf changes
Configuring DNS key synchronization service (ipa-dnskeysyncd)
[1/7]: checking status
[2/7]: setting up bind-dyndb-ldap working directory
[3/7]: setting up kerberos principal
[4/7]: setting up SoftHSM
[5/7]: adding DNSSEC containers
[6/7]: creating replica keys
[7/7]: configuring ipa-dnskeysyncd to start on boot
```

```
Done configuring DNS key synchronization service (ipa-dnskeysyncd).
Restarting ipa-dnskeysyncd
Restarting named
Updating DNS system records
Configuring client side components
Using existing certificate '/etc/ipa/ca.crt'.
Client hostname: host06.example.local
Realm: EXAMPLE.LOCAL
DNS Domain: example.local
IPA Server: host06.example.local
BaseDN: dc=example,dc=local

Skipping synchronizing time with NTP server.
New SSSD config will be created
Configured sudoers in /etc/nsswitch.conf
Configured /etc/sssd/sssd.conf
trying https://host06.example.local/ipa/json
[try 1]: Forwarding 'schema' to json server
'https://host06.example.local/ipa/json'
trying https://host06.example.local/ipa/session/json
[try 1]: Forwarding 'ping' to json server
'https://host06.example.local/ipa/session/json'
[try 1]: Forwarding 'ca_is_enabled' to json server
'https://host06.example.local/ipa/session/json'
Systemwide CA database updated.
Adding SSH public key from /etc/ssh/ssh_host_rsa_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_key.pub
Adding SSH public key from /etc/ssh/ssh_host_ed25519_key.pub
[try 1]: Forwarding 'host_mod' to json server
'https://host06.example.local/ipa/session/json'
SSSD enabled
Configured /etc/openldap/ldap.conf
Configured /etc/ssh/ssh_config
Configured /etc/ssh/sshd_config
Configuring example.local as NIS domain.
Client configuration complete.
The ipa-client-install command was successful

=====
Setup complete

Next steps:
```

1. You must make sure these network ports are open:

TCP Ports:

- \* 80, 443: HTTP/HTTPS
- \* 389, 636: LDAP/LDAPS
- \* 88, 464: kerberos
- \* 53: bind

UDP Ports:

- \* 88, 464: kerberos
- \* 53: bind
- \* 123: ntp

2. You can now obtain a kerberos ticket using the command:  
`'kinit admin'`

This ticket will allow you to use the IPA tools (e.g.,  
`ipa user-add`) and the web user interface.

Be sure to back up the CA certificates stored in  
`/root/cacert.p12`

These files are required to create replicas. The password for  
these files is the Directory Manager password

b. Authenticate with the IPA server with the admin username.

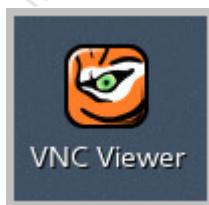
- Enter the password created for the IPA admin.

```
kinit admin
```

```
Password for admin@EXAMPLE.LOCAL: oracleipa
```

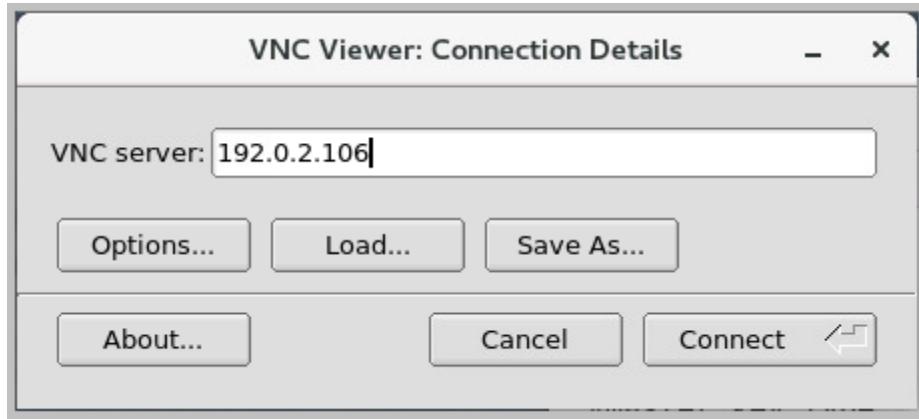
4. From **gateway**, connect to **host06** by using VNC Viewer.

a. Double-click the “VNC Viewer” icon on the **gateway** desktop.

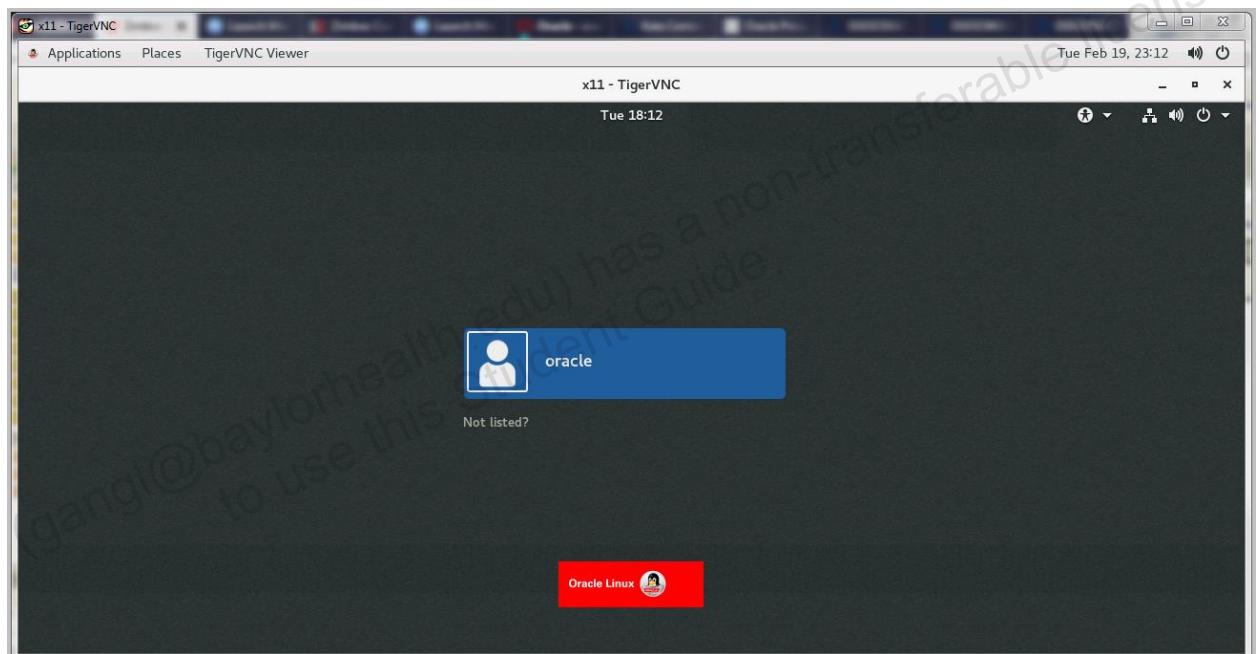


b. Enter 192.0.2.106 as shown in the following window.

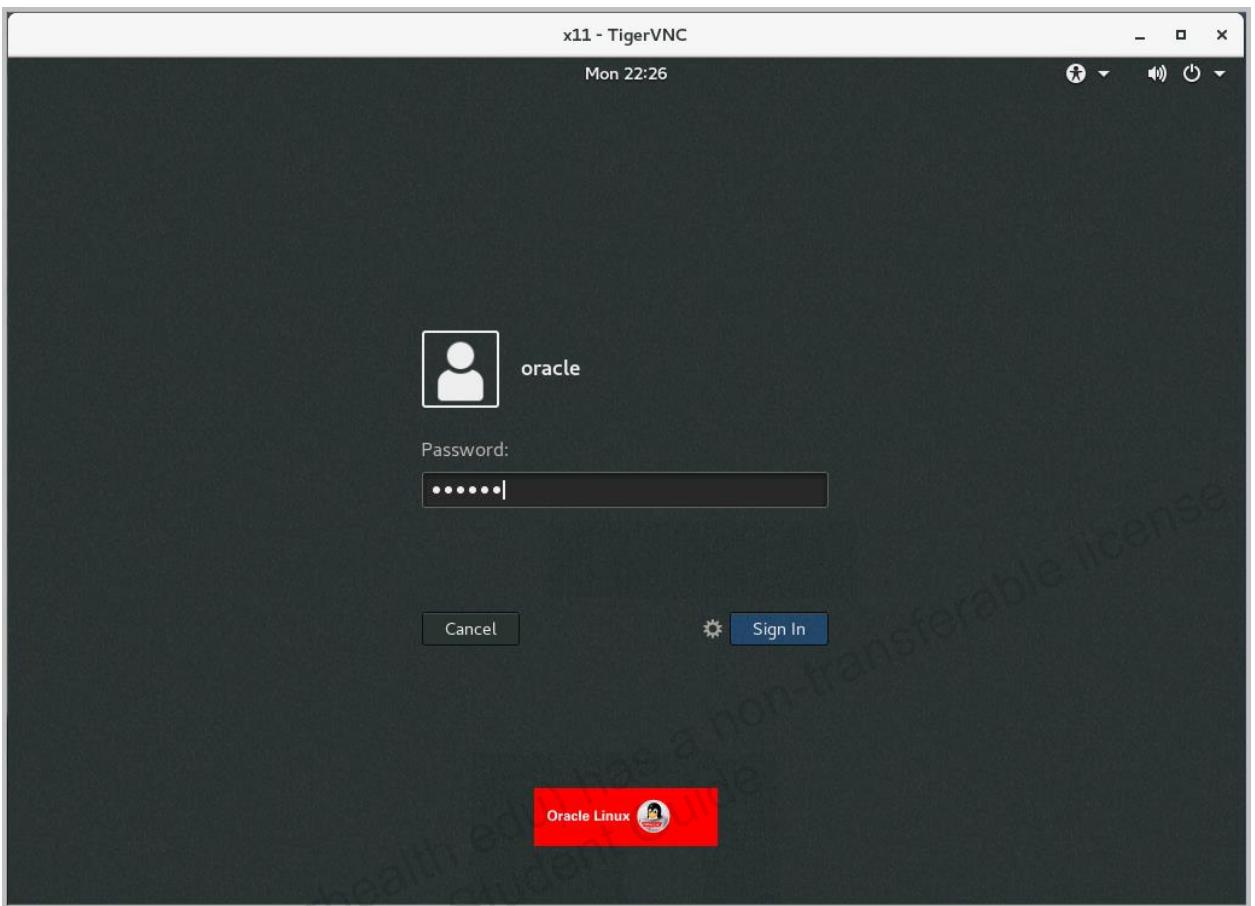
- Click Connect.



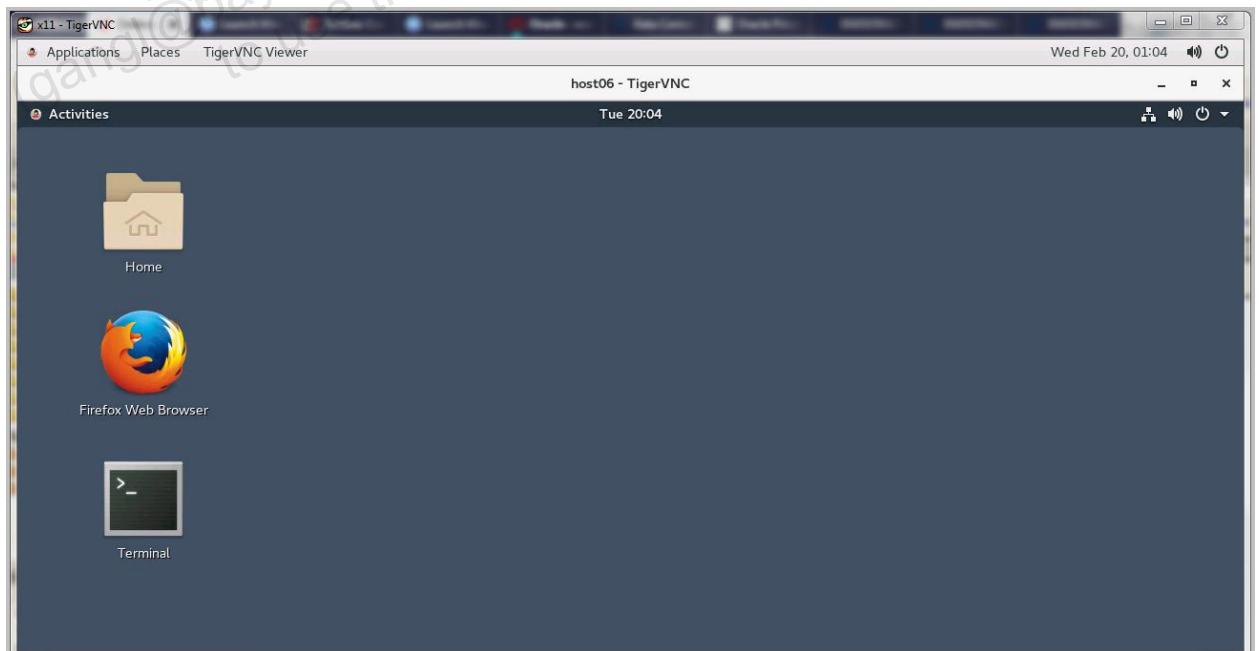
- The login window appears.
- Click in the window to see the `oracle` user.



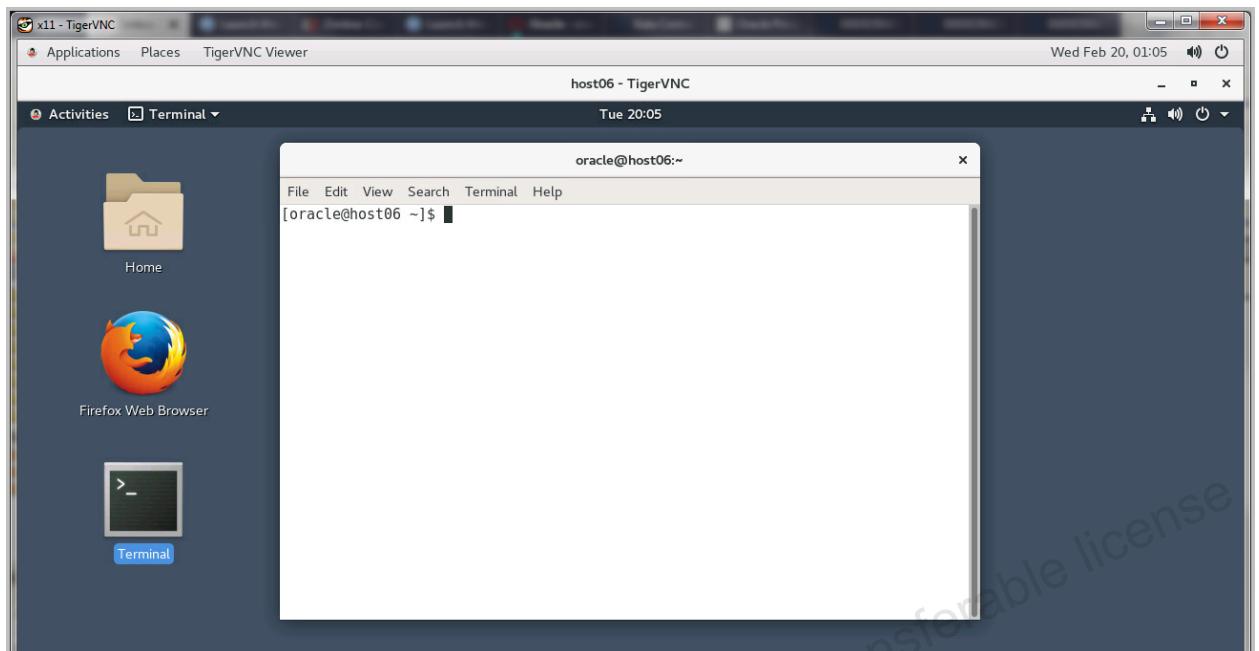
- c. Click the `oracle` user and login.



- The desktop appears as shown.



- Double-click the Terminal icon to open a terminal window as shown.



5. Bring up the Firefox browser for signing in to the Identity Management GUI.

- a. Use the `su -` command to become the root user.

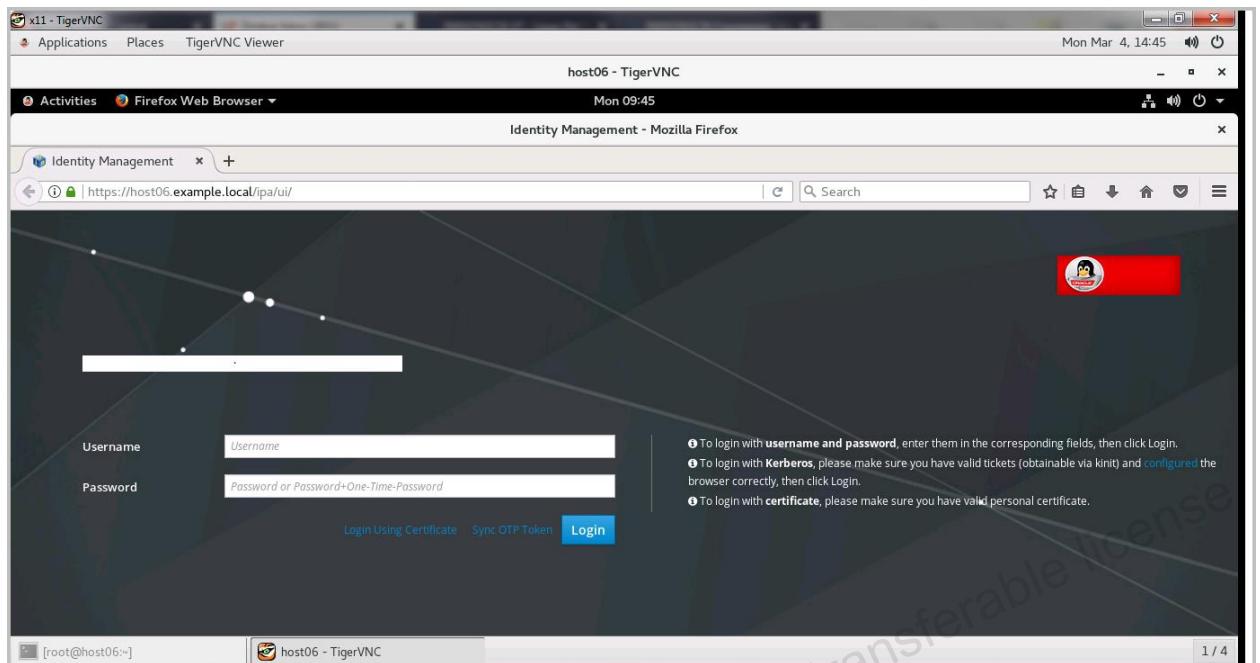
```
[oracle@host06 ~]$ su -
Password:
Last login: ...
[root@host06 ~]#
```

- b. Bring up web browser by using the following command.

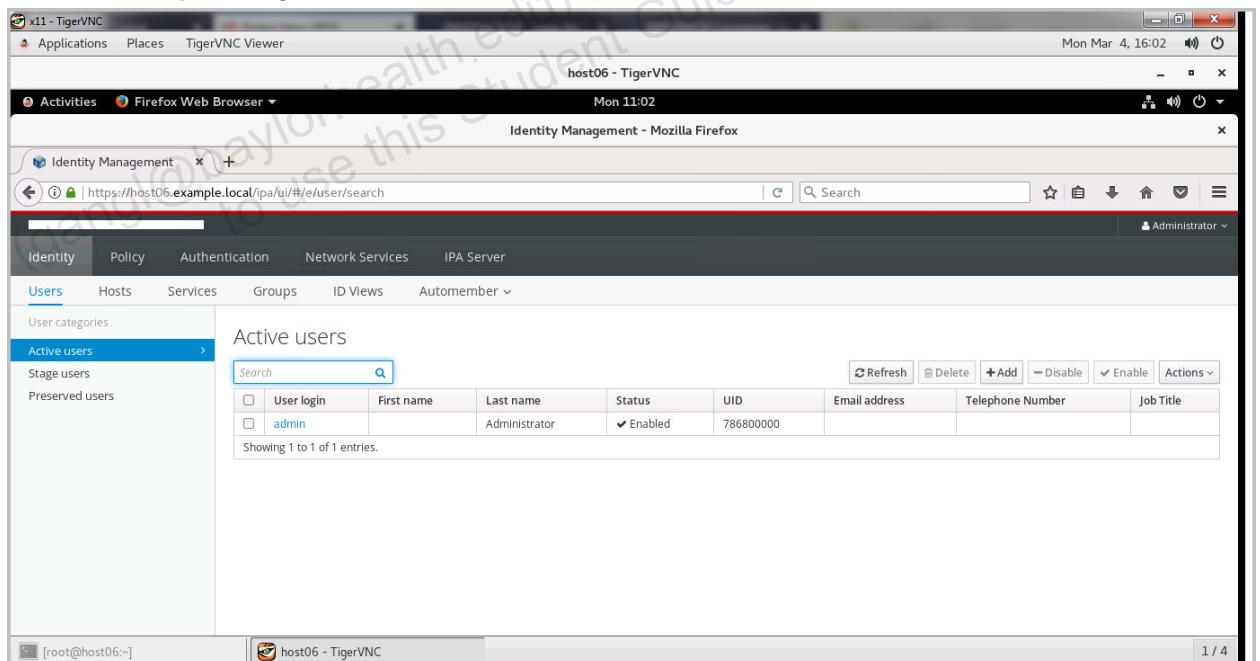
```
firefox &
```

6. Use the Identity Management user interface.

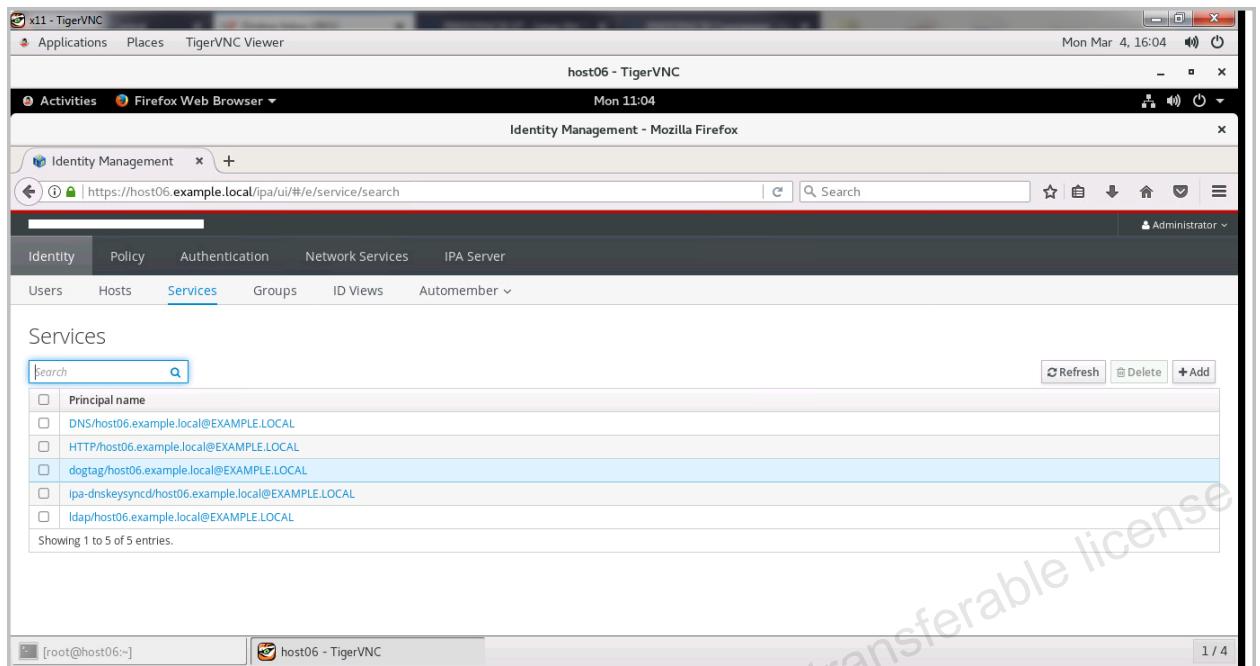
- a. Enter `127.0.0.1` in the browser address bar.
    - The Identity Management sign in screen appears in the browser.



- b. Enter the username `admin` and the `oracleipa` password for the IPA admin.
- Click the Login button to submit the credentials.
  - The Identity Management interface appears in the browser.

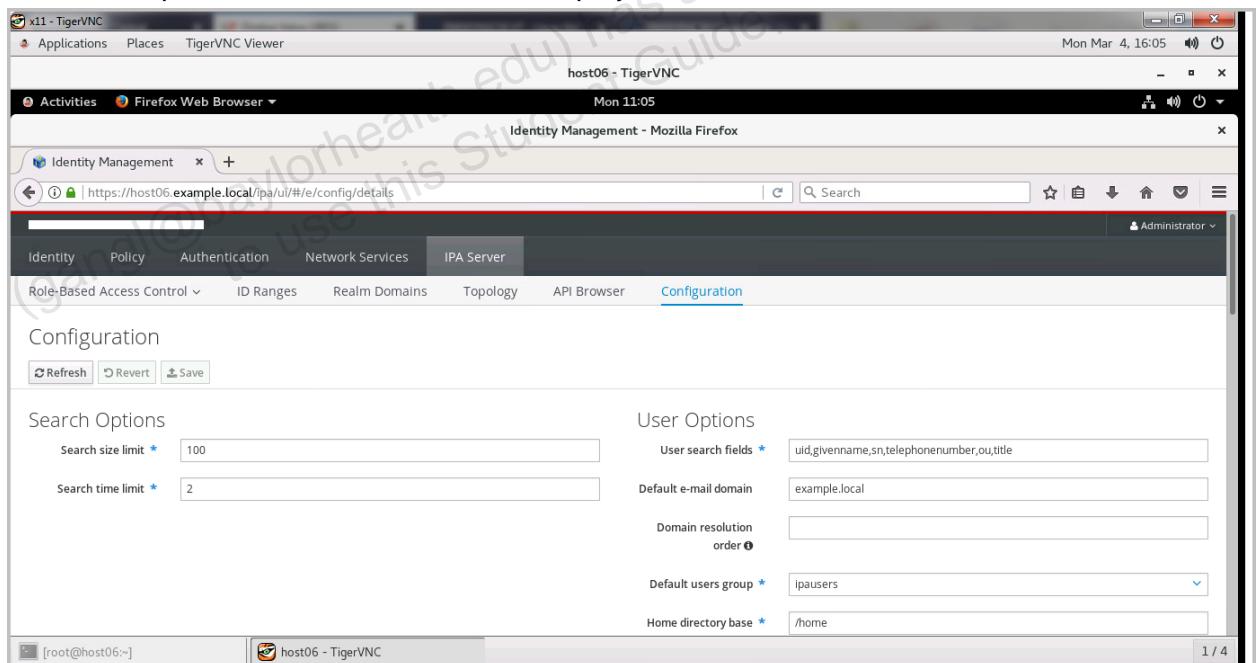


- c. Click on the Services tab.
- A list of installed Identity servers are displayed.



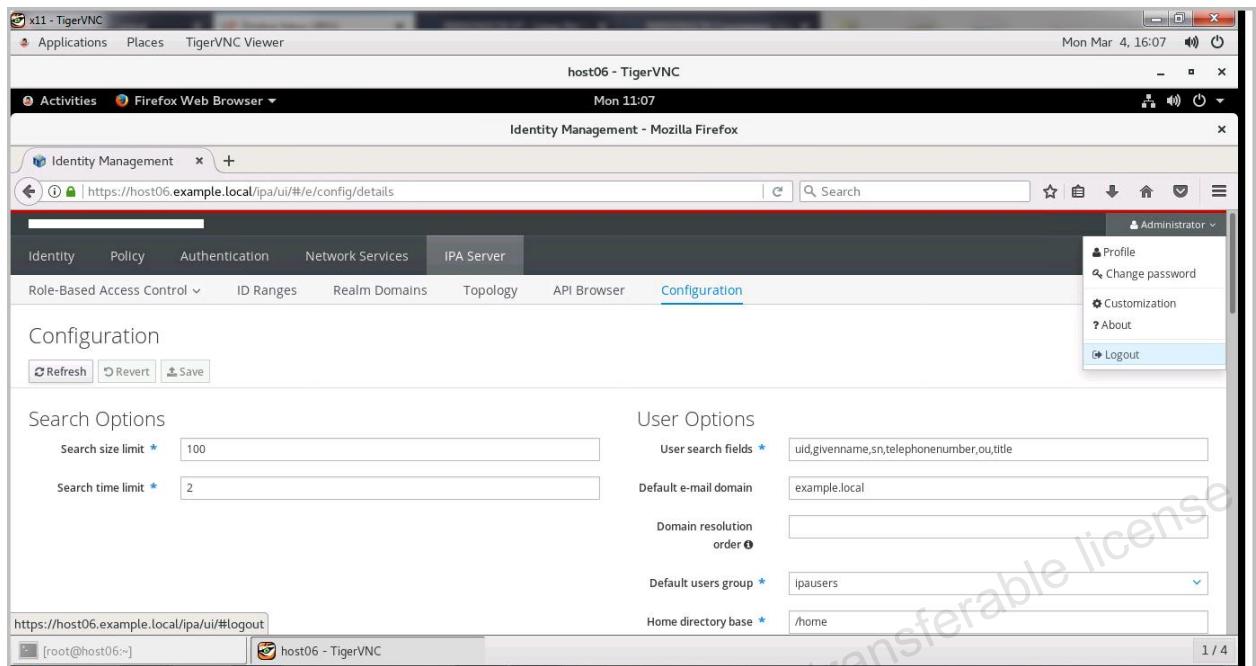
d. Click on the IPA Server tab and then Configuration.

- A list of options for the IPA Server are displayed.



e. To logout, click on the Administrator tab and then click Logout.

- You are logged out of the IPA Management console.



- f. Click the x in top right corner of the browser to close Firefox.
- The Firefox browser is closed.

## Practice 15-3: Configuring SSSD Services

---

### Overview

In this practice, you:

- Confirm installation of SSSD
- Confirm status of the SSSD running
- Configure SSSD for NSS
- Configure SSSD for PAM
- Configure SSSD for sudo

### Assumptions

- This practice is performed on **host06** VM.

### Tasks

1. Open a new terminal on **host06**. Right-click on the desktop and click **Open Terminal**.
2. Configure the Name Service Switch (NSS).
  - a. Use the `authconfig` command to enable SSSD.

```
authconfig --enablenesssd --update
```

- b. Use the `cp` command to create a backup copy of `/etc/nsswitch.conf` file.

```
cp /etc/nsswitch.conf /etc/nsswitch.BAK
```

- c. Use the `vi` command to edit the `/etc/nsswitch.conf` for mapping the NSS service.

- At around line number 57, verify the `services` entry includes `sss` as shown.

```
vi /etc/nsswitch.conf
...
services: files sss
...
```

- d. Use the `cp` command to create a backup copy of `/etc/sssd/sssd.conf` file.

```
cp /etc/sssd/sssd.conf /etc/sssd/sssd.original
```

- e. Use the `vi` command to edit the `sssd.conf`.

- At or around line 15, verify the `services` entry includes `nss` as shown.
- At or around line 18, edit to add the `[nss]` configuration as shown.

```
vi /etc/sssd/sssd.conf
...
[sssd]
services = sudo, nss, ifp, pam, ssh
...
[nss]
memcache_timeout = 600
homedir_substring = /home
```

```
filter_groups = root
filter_users = root
reconnection_retries = 3
entry_cache_timeout = 300
entry_cache_nowait_percentage = 75
```

- f. Save and close the file by entering :x.
- g. Use the `systemctl` command to restart the SSSD service.

```
systemctl restart sssd
```

- h. Verify SSSD service is actively running.

```
systemctl status sssd
 sssd.service - System Security Services Daemon
 Loaded: loaded (/usr/lib/systemd/system/sssd.service;
 enabled; vendor preset: disabled)
 Active: active (running) ...
...
...
```

### 3. Configuring PAM

- a. Use the `authconfig` command to enable SSSD for system authentication:

```
authconfig --update --enablesssd --enablesssdauth
```

- b. Use the `vi` command to edit the configuration file `sssd.conf`.
- c. Verify PAM exists as a SSSD service:
  - At or around line 15, verify `pam` is an included service.
  - Add the following configurations if they are not present.

```
vi /etc/sssd/sssd.conf
...
[sssd]
config_file_version = 2
reconnection_retries = 3
sbus_timeout = 30
services = sudo, nss, ifp, pam, ssh
...
```

- d. Next, verify a PAM section exists with the following parameters defined.
  - At or around line 30, verify `pam` is an included service.
  - Add the following configurations if they are not present.

```
[pam]
reconnection_retries = 3
offline_credentials_expiration = 2
offline_failed_login_attempts = 3
offline_failed_login_delay = 5
...
```

- e. Save and close the file by entering :x.
- f. Use the `systemctl` command to restart the SSSD service.

```
systemctl restart sssd
```

- g. Use the `systemctl` command to verify SSSD service is actively running:

```
systemctl status sssd
 sssd.service - System Security Services Daemon
 Loaded: loaded (/usr/lib/systemd/system/sssd.service; enabled; vendor preset: disabled)
 Active: active (running) since ...
 ...

```

#### 4. Configuring sudo.

- a. Use the `rpm` command to query and verify the installation of the `sssd-common` package.

```
rpm -qa sssd-common
sssd-common-1.16.2-13.el7.x86_64
```

- b. Use the `vi` command to edit the `sssd.conf` file.

- Verify and add `[sudo]` to the SSSD services if it is not already listed.
- At or around line number 18, verify `sudo` is included in the `services` values.
- At or around line 36, add the `[sudo]` section to the `sssd.conf` file if it does not exist.
- Add an LDAP domain to the `[sudo]` section for providing LDAP information for the `sudo` service.
- At or around line 50, add the section for ldap authentication.

```
vi /etc/sssd/sssd.conf
...
services = sudo, nss, ifp, pam, ssh
...
[sudo]
[domain/LDAP]
id_provider = ldap
sudo_provider = ldap
ldap_uri = ldap://example.local
ldap_sudo_search_base = ou=sudoers,dc=example,dc=local
```

- c. Save and close the file by entering :x.
- d. Use the `systemctl` command to restart the SSSD service.

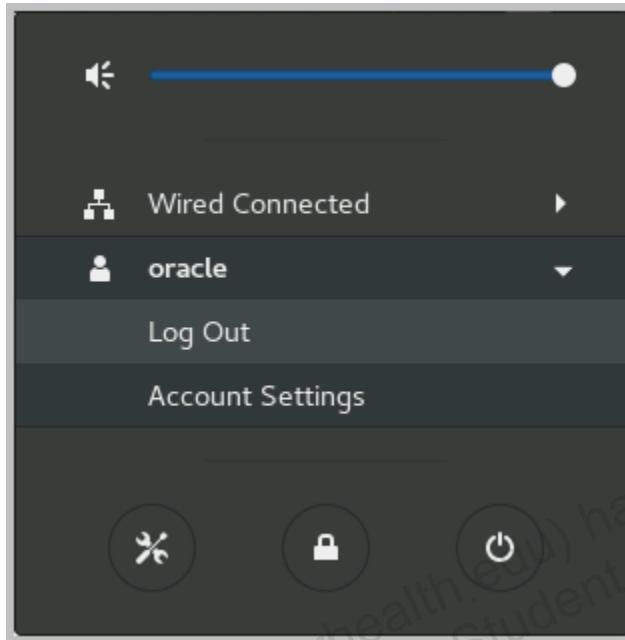
```
systemctl restart sssd
```

- e. Use the `systemctl` command to verify SSSD service is actively running:

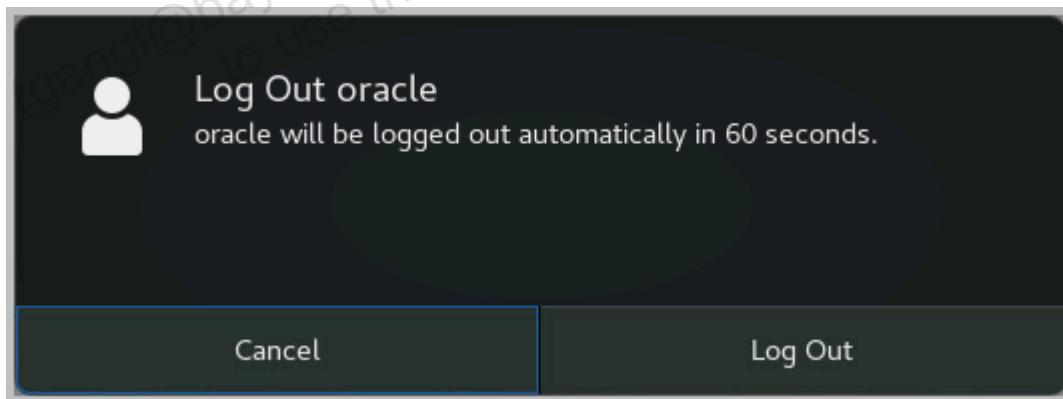
```
systemctl status sssd
 sssd.service - System Security Services Daemon
 Loaded: loaded (/usr/lib/systemd/system/sssd.service; enabled; vendor preset: disabled)
```

```
Active: active (running) ...
```

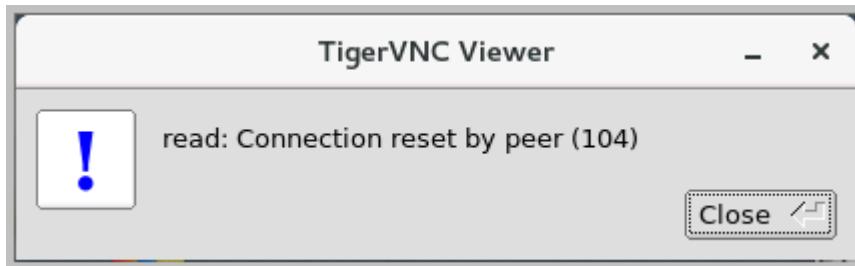
5. Log out and close the VNC connection to **host06**.
  - a. Click the down arrow in the top-right corner of the window to display the drop-down menu.
  - b. Click the `oracle` user, then click Log Out as shown.



- c. Click Log Out as shown.



- The **host06** desktop closes.
- The following window might appear. Close the window by clicking the X in the upper-right corner.



- d. If you are connected to **host06** from **gateway** by using the `ssh` command, log out from this window also.

```
[host06]# exit
logout
Connection to host06 closed.
[root@gateway ~]#
```

GANG LIU (gangli@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.