ORACLE
UNIVERSITY

Hardware and Software
Engineered to Work Together

Oracle Linux 7: System Administration

Activity Guide | Volume 2
D88168GC10
Edition 1.0 | January 2015 | D89911

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

ORACLE

Author

Craig McBride

Technical Contributors and Reviewers

Yasar Akthar, Gavin Bowe, Avi Miller, Chris Potter, Tim Hill, Manish Kapur,
Wim Coekaerts, Al Flournoy, Joel Goodman, Harald Van Breederode, Michele Dady,
Steve Miller, Antoinette O'Sullivan

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Course Introduction 1-1
Course Practice Environment: Security Credentials 1-2
Practices for Lesson 1: Overview 1-3
Practice 1-1: Exploring the dom0 Environment 1-4
Practice 1-2: Starting, Stopping, and Listing Guests 1-10
Practice 1-3: Exploring the host01 VM 1-12
Practice 1-4: Exploring the host02 VM 1-17
Practice 1-5: Logging Off from Your Student PC 1-20

Practices for Lesson 2: Introduction to Oracle Linux 2-1
Practices for Lesson 2: Overview 2-2
Practice 2-1: Quiz – Introduction to Oracle Linux 2-3
Solution 2-1: Quiz – Introduction to Oracle Linux 2-5
Practice 2-2: Viewing Kernel Information 2-6

Practices for Lesson 3: Installing Oracle Linux 7 3-1
Practices for Lesson 3: Overview 3-2
Practice 3-1: Installing Oracle Linux 7 3-3
Practice 3-2: Completing Initial Setup and FirstBoot 3-36
Practice 3-3: Logging in to Oracle Linux and Shutting Down 3-42
Practice 3-4: Re-Creating the host03 VM Guest 3-50

Practices for Lesson 4: Oracle Linux 7 Boot Process 4-1
Practices for Lesson 4: Overview 4-2
Practice 4-1: Exploring the GRUB 2 Boot Loader 4-3
Practice 4-2: Booting Different Kernels 4-10
Practice 4-3: Using the GRUB 2 Menu 4-18
Practice 4-4: Exploring `systemd` Units 4-24
Practice 4-5: Working with `systemd` Target and Service Units 4-29

Practices for Lesson 5: System Configuration 5-1
Practices for Lesson 5: Overview 5-2
Practice 5-1: Configuring Date and Time 5-3
Practice 5-2: Configuring NTP and Chrony 5-6
Practice 5-3: Exploring `/etc/sysconfig` 5-14
Practice 5-4: Exploring `/proc` 5-17
Practice 5-5: Exploring the sysfs File System 5-20
Practice 5-6: Using `sysctl` 5-22

Practices for Lesson 6: Package Management 6-1
Practices for Lesson 6: Overview 6-2
Practice 6-1: Using the `rpm` Utility 6-3
Practice 6-2: Accessing the Public Yum Server 6-7
Practice 6-3: Creating a Local Yum Repository 6-14
Practice 6-4: Using the `yum` Utility 6-17
Practice 6-5: Unbreakable Linux Network (ULN) 6-20

Practices for Lesson 7: Ksplice 7-1
Practices for Lesson 7: Overview 7-2

Practice 7-1: Using Ksplice Uptrack 7-3
Practice 7-2: Installing the Ksplice Offline Client and Kernel Updates 7-14

Practices for Lesson 8: Automating Tasks 8-1
Practices for Lesson 8: Overview 8-2
Practice 8-1: Automating Tasks 8-3

Practices for Lesson 9: Kernel Module Configuration 9-1
Practices for Lesson 9: Overview 9-2
Practice 9-1: Using Loadable Kernel Modules 9-3

Practices for Lesson 10: User and Group Administration 10-1
Practices for Lesson 10: Overview 10-2
Practice 10-1: Administering User Accounts 10-3
Practice 10-2: Administering Group Accounts 10-12
Practice 10-3: Implementing User Private Groups 10-13
Practice 10-4: Configuring Password Aging 10-15
Practice 10-5: Using the User Manager GUI 10-17
Practice 10-6: Restricting the Use of the su Command 10-25
Practice 10-7: Allowing the Use of the sudo Command 10-27

Practices for Lesson 11: Partitions, File Systems, and Swap 11-1
Practices for Lesson 11: Overview 11-2
Practice 11-1: Listing the Current Disk Partitions 11-3
Practice 11-2: Partitioning a Storage Device 11-7
Practice 11-3: Creating ext File Systems 11-12
Practice 11-4: Increasing Swap Space 11-15

Practices for Lesson 12: XFS File System 12-1
Practices for Lesson 12: Overview 12-2
Practice 12-1: Creating an XFS File System 12-3
Practice 12-2: Setting Quotas on an XFS File System 12-8
Practice 12-3: Backing Up and Restoring XFS File Systems 12-13

Practices for Lesson 13: Btrfs File System 13-1
Practices for Lesson 13: Overview 13-2
Practice 13-1: Creating a Btrfs File System 13-3
Practice 13-2: Working with Subvolumes and Snapshots 13-12
Practice 13-3: Recovering from Data Corruption 13-19

Practices for Lesson 14: Storage Administration 14-1
Practices for Lesson 14: Overview 14-2
Practice 14-1: Creating Linux LVM Partitions 14-3
Practice 14-2: Creating a Logical Volume 14-6
Practice 14-3: Creating a File System and Mounting a Logical Volume 14-9
Practice 14-4: Backing Up Volume Group Metadata 14-10
Practice 14-5: Creating a Logical Volume Snapshot 14-14
Practice 14-6: Increasing the Capacity of a Logical Volume 14-17
Practice 14-7: Restoring Volume Group Metadata 14-21
Practice 14-8: Creating a Thinly Provisioned Logical Volume 14-22
Practice 14-9: Using Snapper with LVM Thin Provisioned Logical Volumes 14-26
Practice 14-10: Creating a RAID Device 14-31

Practices for Lesson 15: Network Configuration 15-1

Practices for Lesson 15: Overview 15-2
Practice 15-1: Configuring the eth1 Network Interface 15-3
Practice 15-2: Using NetworkManager with the GNOME GUI 15-8
Practice 15-3: Using the Network Connection Editor 15-26
Practice 15-4: Using the nmcli Utility 15-29
Practice 15-5: Using the nmtui Utility 15-42
Practice 15-6: Using the ip Utility 15-46

Practices for Lesson 16: File Sharing 16-1
Practices for Lesson 16: Overview 16-2
Practice 16-1: Configuring an NFS Server and an NFS Client 16-3
Practice 16-2: Using Automounter 16-9
Practice 16-3: Configuring an FTP Server 16-12
Practice 16-4: Downloading a File from an FTP Server 16-14

Practices for Lesson 17: OpenSSH 17-1
Practices for Lesson 17: Overview 17-2
Practice 17-1: Connecting to a Remote Server by Using ssh 17-3
Practice 17-2: Configuring OpenSSH to Connect Without a Password 17-6

Practices for Lesson 18: Security Administration 18-1
Practices for Lesson 18: Overview 18-2
Practice 18-1: Configuring a `chroot` Jail 18-3
Practice 18-2: Configuring a `chroot` Jail for ftp Users 18-6
Practice 18-3: Exploring `firewalld` 18-11
Practice 18-4: Configuring `firewalld` 18-23
Practice 18-5: Configuring `iptables` 18-27
Practice 18-6: Configuring a TCP Wrapper 18-31

Practices for Lesson 19: Oracle on Oracle 19-1
Practices for Lesson 19: Overview 19-2
Practice 19-1: Using `sftp` to Upload `oracle` Packages 19-3
Practice 19-2: Installing and Running Oracle RDBMS Pre-Install 19-4
Practice 19-3: Preparing Disk for ASM Use 19-9
Practice 19-4: Installing and Configuring ASMLib 19-14

Practices for Lesson 20: System Monitoring 20-1
Practices for Lesson 20: Overview 20-2
Practice 20-1: Using `sosreport` to Collect System Information 20-3
Practice 20-2: Using Standard Linux Performance Monitoring Tools 20-6
Practice 20-3: Installing and Using OSWatcher 20-27
Practice 20-4: Using OSWatcher Analyzer 20-34

Practices for Lesson 21: System Logging 21-1
Practices for Lesson 21: Overview 21-2
Practice 21-1: Configuring System Logging 21-3
Practice 21-2: Using `rsyslog` Templates 21-8
Practice 21-3: Using `logwatch` 21-10
Practice 21-4: Using `journald` 21-12

Practices for Lesson 22: Troubleshooting 22-1
Practices for Lesson 22: Overview 22-2
Practice 22-1: Transferring Utilities from dom0 22-3

Practice 22-2: System Boots into Single-User Mode 22-5
Solution 22-2: System Boots into Single-User Mode 22-7
Practice 22-3: Status Commands Fail 22-9
Solution 22-3: Status Commands Fail 22-11
Practice 22-4: `cron` Job Fails to Run 22-12
Solution 22-4: `cron` Job Fails to Run 22-14
Practice 22-5: User Cannot Log In 22-16
Solution 22-5: User Cannot Log In 22-18
Practice 22-6: File System Troubleshooting 22-20
Solution 22-6: File System Troubleshooting 22-22
Practice 22-7: Logical Volume Space Is Exhausted 22-25
Solution 22-7: Logical Volume Space Is Exhausted 22-30
Practice 22-8: Network Connectivity Problem 22-32
Solution 22-8: Network Connectivity Problem 22-34
Practice 22-9: NFS Permission Problem 22-36
Solution 22-9: NFS Permission Problem 22-41
Practice 22-10: Remote Access Problem 22-45
Solution 22-10: Remote Access Problem 22-47
Practice 22-11: Log File Is Not Getting Updated 22-50
Solution 22-11: Log File Is Not Getting Updated 22-53
Appendix A: Source Code for Problem-Causing Executables 22-55

Appendixes: Remote Access Options 23-1
Appendixes: Remote Access Options – Overview 23-2
Appendix A: Using an NX Client to Connect to dom0 23-3
Appendix B: Using an NX Player to Connect to dom0 23-7
Appendix C: Using VNC (TightVNC) to Connect Directly to VM Guests 23-13
Appendix D: Using NoMachine Version 4 to Connect to dom0 23-16

Practices for Lesson 14: Storage Administration

Chapter 14

Practices for Lesson 14: Overview

Practices Overview

In these practices, you create and work with logical volumes and RAID devices.

Practice 14-1: 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 **host03** VM.
- There are no partitions on `/dev/xvdb` and `/dev/xvdd`.
- There are no file systems mounted on `/dev/xvdb` and `/dev/xvdd`.

Tasks

1. Create two 1G partitions on `/dev/xvdb`.
   a. Use the `fdisk` command to create two 1G primary partitions on `/dev/xvdb` as shown in the following:

```
# fdisk /dev/xvdb
Welcome to dfisk (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 (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): +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 (2099200-10485759, default 2099200): ENTER
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-10485759, default
10485759): +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/xvdb: 5368 MB, 5368709120 bytes, 10485760 sectors
...
    Device  Boot  Start      End    Blocks    Id   System
/dev/xvdb1          2048  2099199   1048576   83   Linux
/dev/xvdb2       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.
```

2. Create two 1G partitions on `/dev/xvdd`.

   a. Use the `fdisk` command to create two 1G primary partitions on `/dev/xvdd` as shown in the following:

```
# fdisk /dev/xvdd
Welcome to dfisk (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 (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): +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 (2099200-10485759, default 2099200): ENTER
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-10485759, default
10485759): +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: 5368 MB, 5368709120 bytes, 10485760 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 14-2: 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.

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/xvdb2       2099200  4196351   1048576    8e    Linux LVM
/dev/xvdd2       2099200  4196351   1048576    8e    Linux LVM
```

   b. Use the `pvcreate` command to create physical volumes on both partitions.

```
# pvcreate -v /dev/xvdb2 /dev/xvdd2
   Set up physical volume for "/dev/xvdb2" with 2097152 ...
   Zeroing start of device /dev/xvdb2
   Writing physical volume data to disk "/dev/xvdb2"
 Physical volume "/dev/xvdb2" successfully created
   Set up physical volume for "/dev/xvdd2" with 2097152 ...
   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/xvdb2" is a new physical volume of "1.00 GiB"
  --- NEW Physical volume ---
  PV Name            /dev/xvdb2
  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/xvdb2      lvm2  a--  1.00g 1.00g
```

```
  /dev/xvdd2       lvm2  a--  1.00g 1.00g
```

c. Use the `pvscan` command to scan all disks for physical volumes.

```
# pvscan
  PV /dev/xvdb2                         lvm2 [1.00 GiB]
  PV /dev/xvdd2                         lvm2 [1.00 GiB]
  Total: 2 [2.00 GiB] / in use: 0 [0 ] / in no VG: 2 [2.00 GiB]
```

3. Create a volume group.

   Use the `vgcreate` command to create a volume group named **myvolg** from the `/dev/xvdb2` physical volume.

```
# vgcreate -v myvolg /dev/xvdb2
    Adding physical volume 'dev/xvdb2' to volume group 'myvolg'
    Archiving volume group "myvolg" metadata (seqno 0).
    Creating volume group backup "/etc/lvm/backup/myvolg"...
  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
...
  VG Size             1020.00 MiB
...
```

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 all physical volumes.  This may take a while...
  Found volume group "myvolg" using metadata type lvm2
```

d. Display information about the physical volumes.

```
# pvs
  PV         VG      Fmt   Attr PSize    PFree
  /dev/xvdb2 myvolg  lvm2  a--  1020.00m 1020.00m
  /dev/xvdd2         lvm2  a--     1.00g    1.00g
```

- Note that the `/dev/xvdb2` 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
    Setting logging type to disk
    Finding volume group "myvolg"
    Archiving volume group "myvolg" metadata (seqno 1).
    Creating logical volume myvol
...
  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 Size                500.00 MiB
...
```

   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/xvdb2  myvolg  lvm2  a--  1020.00m 520.00m
  /dev/xvdd2          lvm2  a--     1.00g   1.00g
```

   - Note that the free space in the `/dev/xvdb2` 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 14-3: 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.

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-0
# ls -l /dev/mapper/myvolg-myvol
lrwxrwxrwx.  /dev/mapper/myvolg-myvol -> ../dm-0
```

   - Note that two entries were automatically created.
   - Note that both entries are symbolic links to /dev/dm-0.

   b. List the /dev/dm-0 entry.

```
# ls -l /dev/dm-0
brw-rw----.  /dev/dm-0
```

   - Note that /dev/dm-0 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
...
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.
   Use the vi editor to add the following line to /etc/fstab.

```
/dev/mapper/myvolg-myvol /myvol  ext4  defaults  0  0
```

Practice 14-4: 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.

Tasks

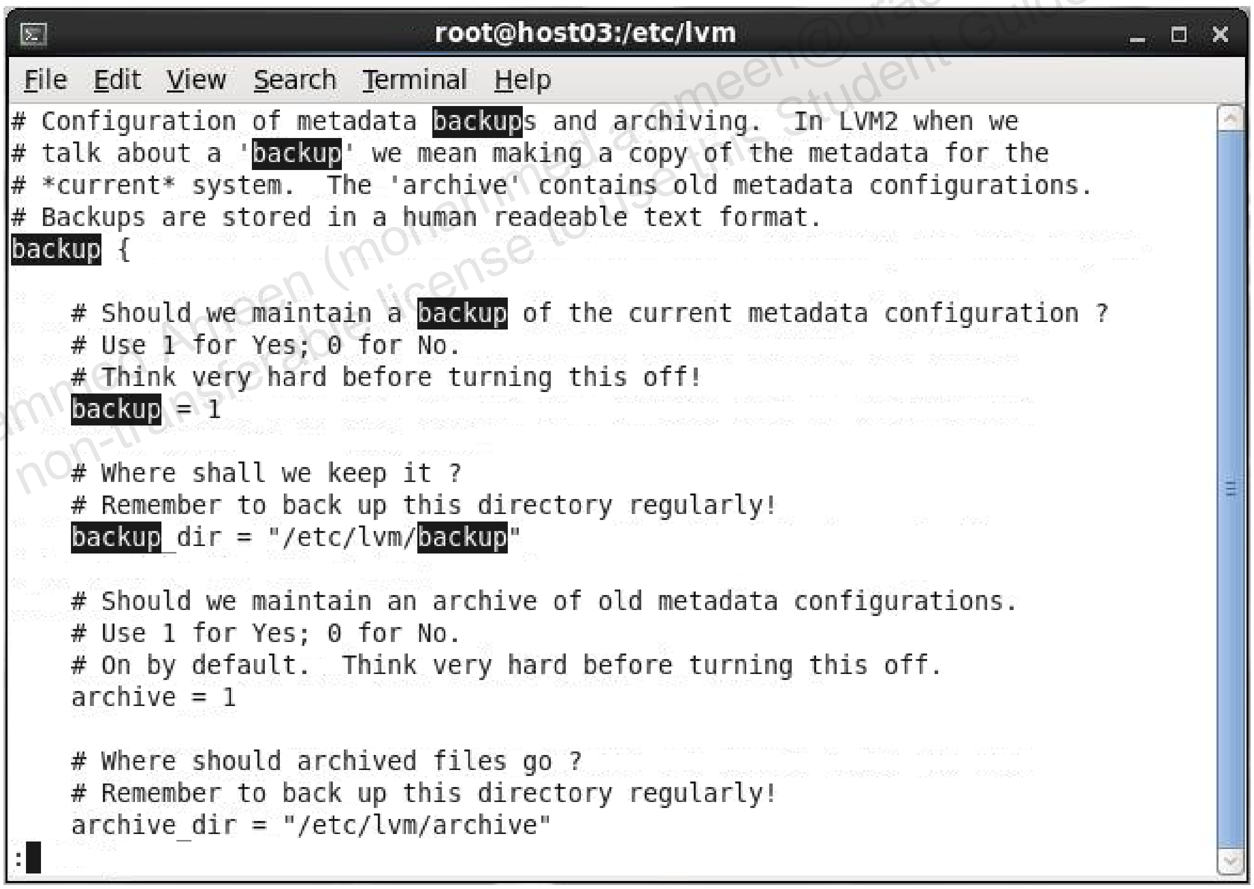
1. View the configuration of metadata backups and archiving.
   a. Use the `less` command to view the logical volume configuration, `/etc/lvm/lvm.conf`.

```
# 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 screenshot:



   - 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.

c. Press `q` to quit the `less` command.

```
:q
#
```

d. 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"
...
```

2. View the metadata backup and archive files.

a. 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.

b. 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/xvdb2"
...
       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. 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-...       myvolg_00001-...
```

- Note that the archive files for the `myvolg` volume group were created automatically.

d. 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/xvdb2"
...
```

- Note that the description states the archive file was created "before" executing the `vgcreate` command string.

e. 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/xvdb2"
...
```

- Note that the description states the archive file was created "before" executing the `lvcreate` command string.

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.

```
# vgcfgbackup -f file_today myvolg
Volume group "myvolg" successfully backed up.
```

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.

```
# 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 14-5: Creating a Logical Volume Snapshot

Overview

In this practice, you create a snapshot volume, mount the snapshot, and remove the snapshot volume.

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-3.10.0-123.el7.x86_64.img
initramfs-3.8.13-35.3.1.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 myvolg/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...
-rw-------. <date_time> myvolg_00001...
-rw-------. <date_time> myvolg_00002...
```

   - 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    0.00
```

   e. List the contents of the `/dev/myvolg` and `/dev/mapper` directories.

```
# ls –l /dev/myvolg
lrwxrwxrwx.  myvol -> ../dm-0
lrwxrwxrwx.  myvol-snapshot -> ../dm-1
# ls –l /dev/mapper
crw-rw----.  control
```

```
lrwxrwxrwx.  myvolg-myvol -> ../dm-0
lrwxrwxrwx.  myvolg-myvol-real -> ../dm-2
lrwxrwxrwx.  myvolg-myvol--snapshot -> ../dm-1
lrwxrwxrwx.  myvolg-myvol--snapshot-cow -> ../dm-3
```

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  100M   348M    23%  /myvol
/dev/mapper/myvolg-myvol--snapshot
               477M  100M   348M    23%  /mnt
```

   c. List the files on `/mnt`.

```
# ls /mnt
initramfs-0-rescue-...img
initramfs-3.10.0-123.el7.x86_64.img
initramfs-3.8.13-35.3.1.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-3.10.0-123...'? y
...
```

   e. List the files on `/mnt`.

```
# ls /mnt
initramfs-0-rescue-...img
initramfs-3.10.0-123.el7.x86_64.img
initramfs-3.8.13-35.3.1.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 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-0
# ls -l /dev/mapper
crw-rw----.  control
lrwxrwxrwx.  myvolg-myvol -> ../dm-0
...
```

- Note that the `dm-1` (**myvol-snapshot**) entries have been deleted.
- The `dm-2` and `dm-3` entries still exist in `/dev/mapper`.

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...
-rw-------. <date_time> myvolg_00001...
-rw-------. <date_time> myvolg_00002...
-rw-------. <date_time> myvolg_00003...
```

- 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 14-6: 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.

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
  Extending logical volume myvol to 1000.00 MiB
  Logical volume myvol successfully resized
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/myvolg-myvol is mounted on /myvol; on-
line resize required
...
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...
-rw-------. <date_time> myvolg_00001...
-rw-------. <date_time> myvolg_00002...
-rw-------. <date_time> myvolg_00003...
-rw-------. <date_time> myvolg_00004...
```

- 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/xvdb2  myvolg  lvm2  a--  1020.00m 20.00m
 /dev/xvdd2          lvm2  a--     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
```

```
    Checking for volume group “myvolg”
...
    Adding physical volume ‘/dev/xvdd2’ to volume group ‘myvolg’
...
  Volume group “myvolg” successfully extended
```

d. List the physical volumes.

```
# pvs
 PV           VG      Fmt   Attr PSize     PFree
 /dev/xvdb2   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...
-rw-------. <date_time> myvolg_00001...
-rw-------. <date_time> myvolg_00002...
-rw-------. <date_time> myvolg_00003...
-rw-------. <date_time> myvolg_00004...
-rw-------. <date_time> myvolg_00005...
```

- 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
  Extending logical volume myvol to 1.46 GiB
```

```
  Logical volume myvol successfully resized
...
The file system 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...
-rw-------. <date_time> myvolg_00001...
-rw-------. <date_time> myvolg_00002...
-rw-------. <date_time> myvolg_00003...
-rw-------. <date_time> myvolg_00004...
-rw-------. <date_time> myvolg_00005...
-rw-------. <date_time> myvolg_00006...
```

- 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 14-7: Restoring Volume Group Metadata

Overview

In this practice, you restore volume group metadata from a backup.

Tasks

1. View the current physical volumes, volume group, and logical volumes configuration.
   a. Use the `pvs` command to display information about the physical volumes.

```
# pvs
 PV          VG      Fmt   Attr PSize    PFree
 /dev/xvdb2  myvolg  lvm2  a--  1020.00m       0
 /dev/xvdd2  myvolg  lvm2  a--  1020.00m 540.00m
```

   b. Use the `vgs` command to display information about the volume groups.

```
# vgs
 VG       #PV #LV #SN Attr   VSize  VFree
 myvolg     2   1   0 wz--n- 1.99g  540.00m
```

   c. 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
```

2. Restore the volume group metadata from a backup.
   a. Use the `vgcfgrestore` command to restore the `myvolg` volume group metadata.

```
# vgcfgrestore myvolg
Restored volume group myvolg
```

   b. Repeat tasks 1a, 1b, and 1c (execute the `pvs`, `vgs`, `lvs` commands).

```
# pvs
 PV          VG      Fmt   Attr PSize    PFree
 /dev/xvdb2  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 14-8: 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.

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 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/xvdb2  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/xvdb2  myvolg   lvm2  a--  1020.00m 1020.00m
 /dev/xvdd2           lvm2  a--     1.00g    1.00g
```

- Note that only the `/dev/xvdb2` 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
...
  Logical volume "mythinpool" created
```

b. Use the `lvs` command to display information about the logical volumes.

```
# lvs
 LV         VG     Attr       LSize   Pool Origin Data% Move...
 mythinpool myvolg twi-a-tz-- 100.00m                0.00
```

c. List the contents of the `/dev/myvolg` directory.

```
# ls -l /dev/myvolg
ls: cannot access /dev/myvolg: No such file or directory
```

- Note that there is no entry in the `/dev` directory for the **myvolg** volume group.

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
...
 Logical volume "thinvol1" created
```

b. 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
 Logical volume "thinvol2" created
```

c. 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
 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.

d. List the contents of the `/dev/myvolg` and `/dev/mapper` directories.

```
# ls -l /dev/myvolg
lrwxrwxrwx.  thinvol1 -> ../dm-4
lrwxrwxrwx.  thinvol2 -> ../dm-5

# ls -l /dev/mapper
...
lrwxrwxrwx.  myvolg-thinvol1 -> ../dm-4
lrwxrwxrwx.  myvolg-thinvol2 -> ../dm-5
```

- In this example, the files that represent the thin volumes are symbolic links to `dm-4` and `dm-5`.

5. Create a file system on the thin volume.

a. 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-a-tz-- 100.00m                   22.12
 thinvol1    myvolg Vwi-a-tz-- 200.00m mythinpool          5.53
 thinvol2    myvolg Vwi-a-tz-- 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-a-tz-- 100.00m                   40.38
 thinvol1    myvolg Vwi-a-tz-- 200.00m mythinpool         14.66
 thinvol2    myvolg Vwi-a-tz-- 200.00m mythinpool          5.53
```

- This shows that you have used 40.38% of the allocated pool data (100 MB).
- This also shows that the thin volume mounted on /myvol1, **thinvol1**, has used 14.66% 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
  Extending logical volume mythinpool_tdata to 500.00 MiB
  Logical volume mythinpool 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-a-tz-- 500.00m                    8.07
 thinvol1    myvolg Vwi-a-tz-- 200.00m mythinpool         14.66
 thinvol2    myvolg Vwi-a-tz-- 200.00m mythinpool          5.53
```

- Note that the size of the thin pool is 500 MB and the percentage used is 8.07%.

Practice 14-9: 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.

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
...
Transaction Summary
=============================================================
Install  1 Package (+3 Dependent packages)

Total download size: 499 k
Installed size: 1.5 M
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
...
```

   - 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 listing 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>
  <uid>...</uid>
</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 the 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------- ... filelist-1.txt
-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>
  <uid>...</uid>
  <pre_num>1</pre_num>
</snapshot>
```

i. View the `filelist-1.txt` file in the `/myvol1/.snapshots/2` directory.

```
# cat /myvol1/.snapshots/2/filelist-1.txt
-..... /vmlinuz-3.10...
```

- Note that the `vmlinuz-3.10...` file was deleted after the pre snapshot and before the post snapshot.

4. Compare the pre and post snapshots.

a. Use the `snapper status` command to display the files and directories that have been added, removed, or modified between pre snapshot `1` and post snapshot `2`.

```
# snapper –c myvol1_snap status 1..2
-..... /vmlinuz-3.10...
```

b. Use the `snapper diff` command to display the differences between the contents of the files in pre snapshot `1` and post snapshot `2`.

```
# snapper –c myvol1_snap diff 1..2
Binary files /myvol1/.snapshots/1/snapshot/vmlinux-3.10... and
/myvol1/.snapshots/2/snapshot/vmlinuz-3.10... differ
```

5. 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.

   - Sample output is shown.

```
# snapper -c myvol1_snap list
Type   | # | Pre # | Date  | User  | Cleanup  |Description| Userdata
------+---+-------+------+------+---------+-----------+--------
single| 0 |       | ...  | root |          | current    |
pre   | 1 |       | ...  | root |          |            |
post  | 2 |  1    | ...  | root |          |            |
```

6. Undo the changes from post snapshot 2 to pre snapshot 1.

   a. List the contents of the /myvol1 volume.

```
# ls /myvol1
lost+found
vmlinuz-0-rescue-...
vmlinuz-3.8.13-35.3.1.el7uek.x86_64
vmlinuz-3.8.13-44.1.1.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.

   b. Use the snapper undochange command to revert the contents of the volume defined by the myvol1_snap configuration file to the pre snapshot 1.

```
# snapper -c myvol1_snap undochange 1..2
create:1 modify:0 delete:0
```

   c. List the contents of the /myvol1 volume.

```
# ls /myvol1
lost+found
vmlinuz-0-rescue-...
vmlinuz-3.10.0-123.el7.x86_64
vmlinuz-3.8.13-35.3.1.el7uek.x86_64
vmlinuz-3.8.13-44.1.1.el7uek.x86_64
```

   - Note that vmlinuz-3.10* is restored.

Practice 14-10: 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.

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 thinvol1?
[y/n]: y
  Logical volume "thinvol1" successfully removed
Do you really want to remove active logical volume 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 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/xvdb2 /dev/xvdd2
  Labels on physical volume "/dev/xvdb2" 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/xvdb2` and `/dev/xvdd2` partitions.
      - Answer **y** when asked, “Continue creating array?”

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/xvdb2
/dev/xvdd2
...
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.

```
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 xvdd2[1] xvdb2[0]
      1048000 blocks supper 1.2 [2/2] [UU]
      [==========>.......]  resync = ...
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] xvdb2[0]
      1048000 blocks supper 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: 1023.44MiB 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 : 1048000 (023.61 MiB 1073.15 MB)
  Used Dev Size : 1048000 (023.61 MiB 1073.15 MB)
   Raid Devices : 2
  Total Devices : 2
...
    Number   Major   Minor   RaidDevice   State
       0      202      18        0       active sync /dev/xvdb2
       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      992M  2.6M   922M    1%  /raid
```

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/xvdb2,/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. Use the `mdadm` command to deactivate the array and release all resources.

```
# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
```

   d. Display the attributes of the `/dev/xvdb3` block device.

```
# blkid | grep raid
/dev/xvdb2: 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/xvdb2` and `/dev/xvdd2` have a type of "`linux_raid_member`."

e. Use the `mdadm` command to overwrite the **md** superblock on `/dev/xvdb2` and `/dev/xvdd2`.

```
# mdadm --zero-superblock /dev/xvdb2
# mdadm --zero-superblock /dev/xvdd2
```

f. Display the attributes of `/dev/xvdb2` and `/dev/xvdd2` to ensure that the **md** superblock was overwritten.

```
# blkid /dev/xvdb2
# blkid /dev/xvdd2
```

- No output indicates the superblock was overwritten.

Practices for Lesson 15: Network Configuration

Chapter 15

Practices for Lesson 15: Overview

Practices Overview

In these practices, you:

- Configure the `eth1` network interface by editing network interface configuration files
- Use NetworkManager with the GNOME GUI to configure network interfaces
- Use the Network Connection editor to configure networking properties
- Use the `nmcli` utility to configure networking properties
- Use the `nmtui` text-based utility to configure network interfaces
- Use the `ip` utility to manage network links, addresses, and the ARP cache

Practice 15-1: Configuring the `eth1` Network Interface

Overview

In this practice, you:

- Display the available network interfaces on your system
- View the network interface configuration files
- Configure a static IP address for the `eth1` network interface
- Update your `/etc/hosts` file
- Display your route table
- Ensure connectivity to **dom0** and the other VM guests

Assumptions

- You are the `root` user on **host03** VM.
- You are connected to **host03** from **dom0** using VNC, not `ssh`.

If you were unable to complete the Oracle Linux 7 installation on **host03** in Practice 3-1, substitute **host02** for **host03** in this practice and all future practices in this course. Use the IP address of `192.168.1.102` for the `eth1` interface if configuring **host02**.

Tasks

1. Use the `ip addr` command to display your available network interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.103/24 brd 192.0.2.255 scope global eth0
    inet6 ...
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

   - Note that you have two Ethernet interfaces (`eth0` and `eth1`) and the loopback interface (`lo`).
   - The `eth0` Ethernet interface has an IP address, but `eth1` does not.

2. View the network interface configuration files.
   a. Use the `cd` command to change to the `/etc/sysconfig/network-scripts` directory.

```
# cd /etc/sysconfig/network-scripts
```

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

```
# ls
ifcfg-eth0  ifdown-post    ifup-bnep  ifup-routes
```

```
ifcfg-eth1   ifdown-ppp      ifup-eth    ifup-sit
ifcfg-lo     ifdown-routes   ipup-ippp   ifup-Team
...
```

- Note that you have a configuration file for `eth0`, `ifcfg-eth0`.
- Note that you have a configuration file for `eth1`, `ifcfg-eth1`.
- Note that you have a configuration file for the loopback interface, `ifcfg-lo`.
- Several interface control scripts exist in this directory to activate and deactivate network interfaces.

c. Use the `cat` command to view the contents of the `ifcfg-eth0` file.

```
# cat ifcfg-eth0
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=eth0
UUID=...
ONBOOT=yes
HWADDR=00:16:3E:00:01:03
IPADDR0=192.0.2.103
PREFIX0=24
GATEWAY0=192.0.2.1
DNS1=152.68.154.3
DNS2=10.216.106.3
DOMAIN=example.com
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

- Note that this Ethernet interface is configured with a static IPv4 address:
  - `BOOTPROTO=none`
  - `IPADDR0=192.0.2.103`
- The network interface configuration file parameters are described in http://docs.oracle.com/cd/E37670_01/E41138/html/ol_about_netconf.html.

3. Configure `eth1` with a static IP address of `192.168.1.103`.

   Use the `vi` editor to edit the `ifcfg-eth1` file as follows (the required changes are listed in the following bullets and are in **bold** font):

   - Change `BOOTPROTO=dhcp` to **`BOOTPROTO=none`**
   - Change `DEFROUTE=yes` to **`DEFROUTE=no`**
   - Change `PEERDNS=yes` to **`PEERDNS=no`**

- Change PEERROUTES=yes to **PEERROUTES=no**
- Change ONBOOT=no to **ONBOOT=yes**
- Add **IPADDR0=192.168.1.103**
- Add **PREFIX0=24**

```
# vi ifcfg-eth1
HWADDR=00:16:3E:00:02:03
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=no
PEERDNS=no
PEERROUTES=no
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eth1
UUID=...
ONBOOT=yes
IPADDR0=192.168.1.103
PREFIX0=24
```

4. Use the vi editor to edit the /etc/hosts file as follows:

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

5. Use the systemctl command to restart the network service.

```
# systemctl restart network
```

6. Use the ip addr command to display the status of the interfaces.

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

```
    inet6 ...
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

- Note that both `eth0` and `eth1` now have IP addresses.

7. Use network interface control scripts to stop and start a specific network interface.
   a. Use the `ifdown` script to stop the `eth1` interface.

```
# ifdown eth1
```

   b. Use the `ip addr` command to display the status of the interfaces.

```
# ip addr
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

   - Note that the `eth1` interface does not have IP addresses.

   c. Use the `ifup` script to start the `eth1` interface.

```
# ifup eth1
Connection successfully activated (D-Bus active path: ...)
```

   d. Use the `ip addr` command to display the status of the interfaces.

```
# ip addr
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

   - Note that the `eth1` interface now has IP addresses.

8. Display the route table.
   a. Use the `netstat -r` command (or `route`) to display the route table.

```
# netstat -r
Destination   Gateway       Genmask         Flags ...  Iface
default       example.com   0.0.0.0         UG         eth0
192.0.2.0     *             255.255.255.0   U          eth0
192.168.1.0   *             255.255.255.0   U          eth1
```

   - Note that all packets destined for the `192.168.1` subnet use the `eth1` interface.
   - Note that all packets destined for the `192.0.2` subnet use the `eth0` interface.
   - Note that all other packets are routed through the `192.0.2.1` default gateway (`example.com`), via `eth0`.

b. You can also use the `ip route` command to display the route table in a different format.

```
# ip route
default via 192.0.2.1 dev eth0 proto static metric 1024
192.0.2.0/24 dev eth0 proto kernel scope link src 192.0.2.103
192.168.1.0/24 dev eth1 proto kernel scope ... 192.168.1.103
```

9. Use the `ping` command to verify that you can communicate to **dom0** and the other VM guests.
   - Press **Ctrl + C** to kill the `ping` command.

```
# ping dom0
64 bytes from example.com (192.0.2.1)...
CTRL-C
# ping host01
64 bytes from host01.example.com (192.0.2.101)...
CTRL-C
# ping host02
64 bytes from host02.example.com (192.0.2.102)...
CTRL-C
```

Practice 15-2: Using NetworkManager with the GNOME GUI

Overview

In this practice, you:

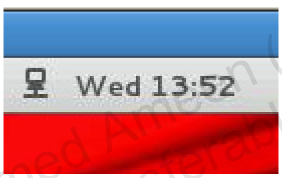
- Ensure that the NetworkManager software package is installed
- Use the NetworkManager GUI to view network status and to disable and enable a network interface
- Use NetworkManager's Network Settings window to disable and enable a network connection
- Use the Network Settings window to configure a network connection
- Use the Network Settings window to add a connection profile
- Use the NetworkManager GUI to select a different connection profile
- View the network interface configuration file for the new connection profile

Assumptions

- You are connected to **host03** VM by using vncviewer.
- You are the `root` user on **host03** VM.

Tasks

1. Install and start NetworkManager if necessary.
   - NetworkManager is installed and running if you see this computer screen icon on the GNOME panel as follows:



   a. Use the `rpm` command to verify that the NetworkManager package is installed.

```
# rpm -qa | grep -i networkmanager
NetworkManager-libreswan-...
NetworkManager-config-server--...
NetworkManager-glib-...
NetworkManager-...
```

   - In this example, NetworkManager is installed.

   b. If NetworkManager is not installed, use the `yum` command to install the package.

```
# yum install NetworkManager
...
```

   c. Use the `systemctl` command to verify that NetworkManager is running.

```
# systemctl status NetworkManager
NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager...)
   Active: active (running) since ...
...
```

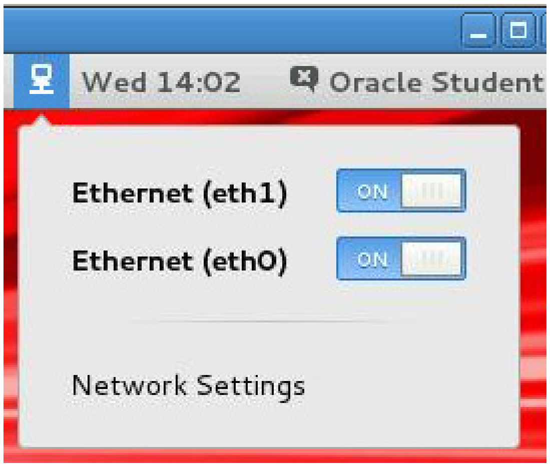
- In this example, NetworkManager is running.

d. If NetworkManager is not running, use the `systemctl` command to start it.

```
# systemctl start NetworkManager
```

2. Use NetworkManager to view network status and disable and enable a network interface.

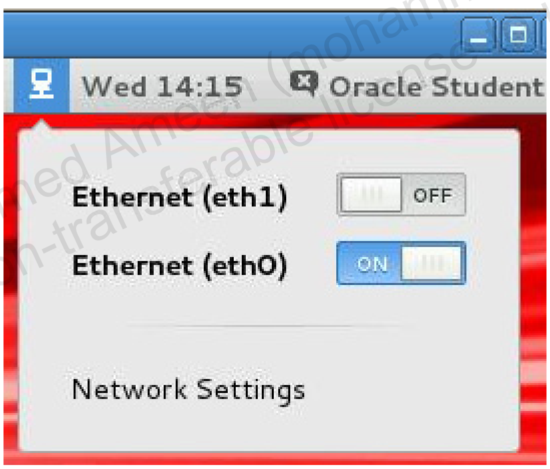
a. Select the computer screen icon to display the following:



- In this example, there are two Ethernet interfaces.
- Both network interfaces are **ON**.

b. Select the **Ethernet (eth1)** entry to toggle the **ON/OFF** switch to **OFF**.

- Select the computer screen icon again to show that **Ethernet (eth1)** is **OFF**.



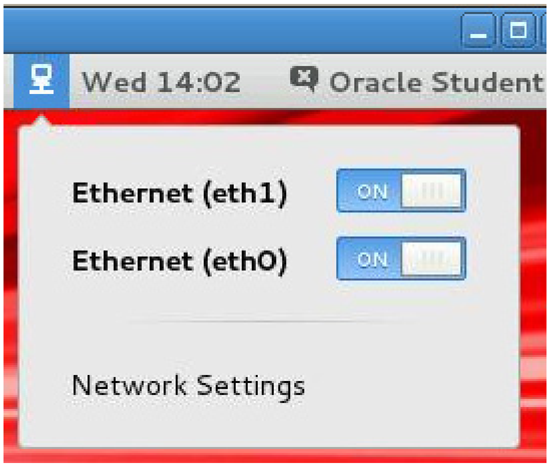
c. From a command window, use the `ip addr` command to display the status of the interfaces.

```
# ip addr
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

- Note that the `eth1` interface does not have IP addresses.

d. Select the computer screen icon and then select the **Ethernet (eth1)** entry to toggle the **ON/OFF** switch to **ON**.

- Select the computer screen icon again to show that **Ethernet (eth1)** is **ON**.



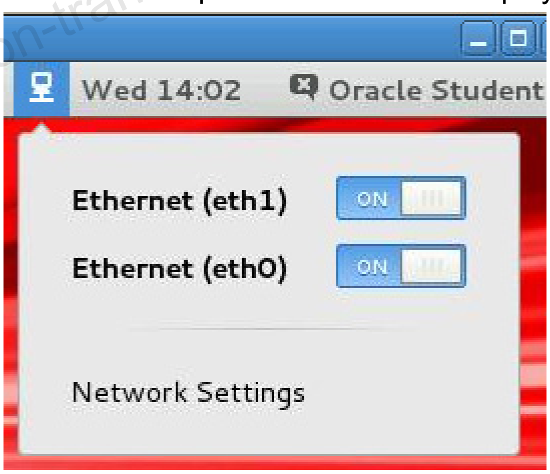
e. From a command window, use the `ip addr` command to display the status of the interfaces.

```
# ip addr
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

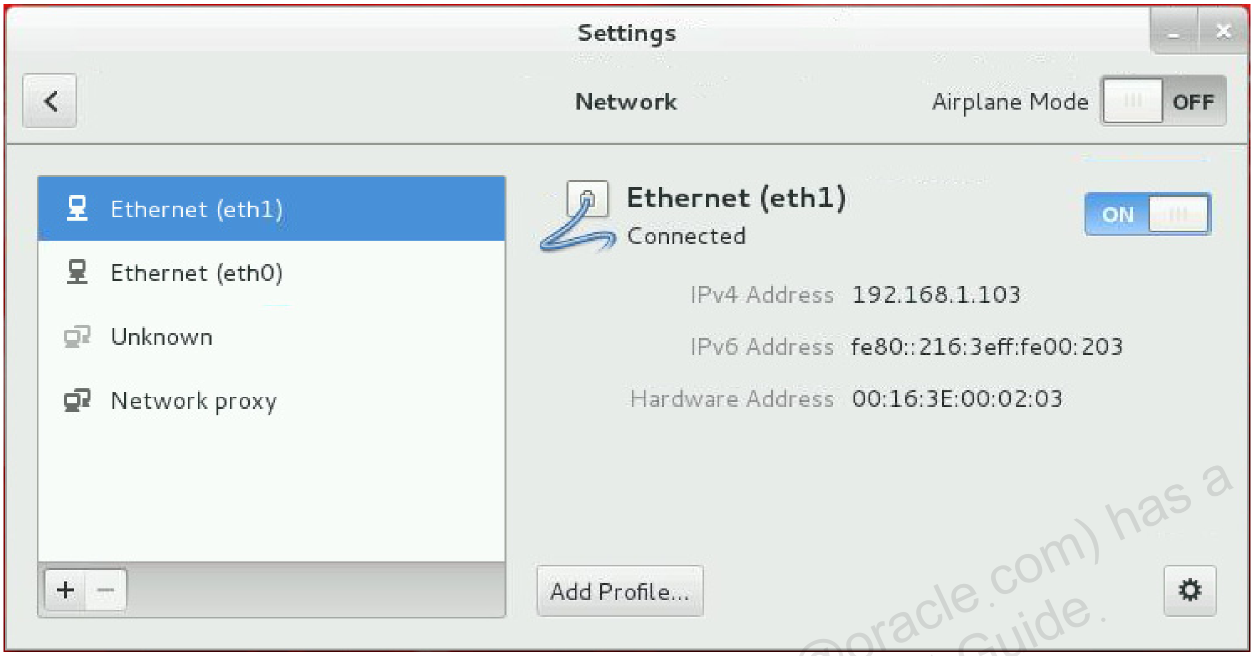
- Note that the `eth1` interface now has IP addresses.

3. Use the **Network Settings** window to disable and enable a network connection.

a. Select the computer screen icon to display the following drop-down menu:

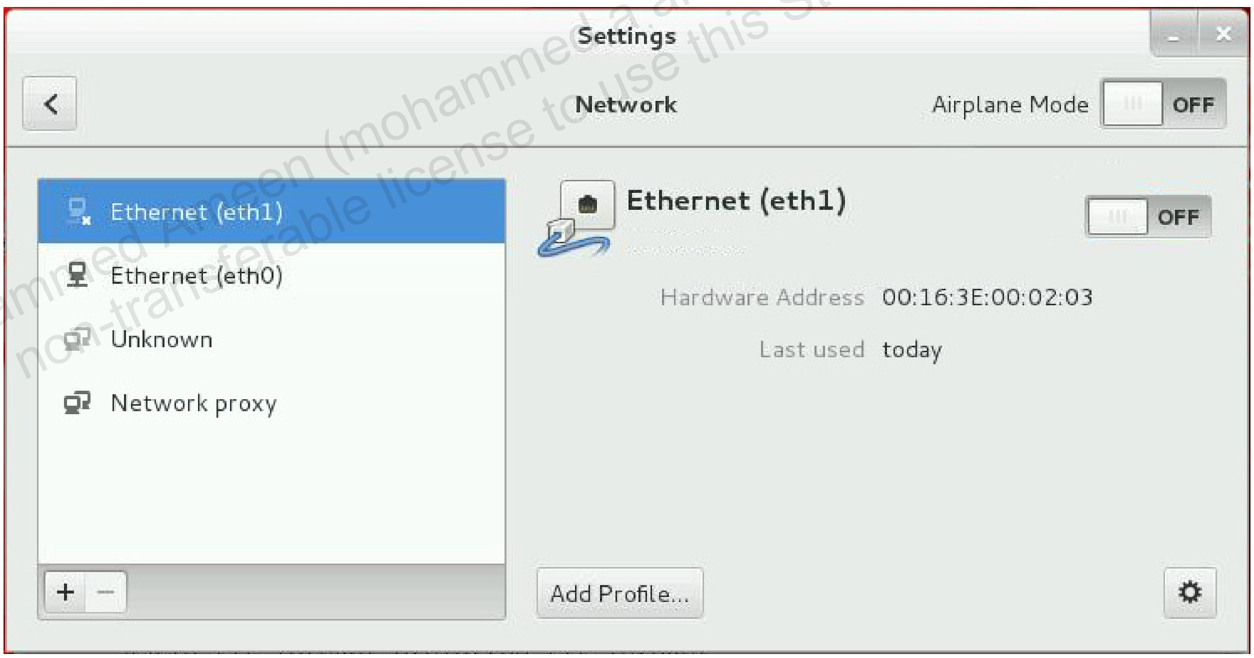


b. Select **Network Settings** to display the following screen:



c. Select the **Ethernet (eth1)** entry on the left and then toggle the **ON/OFF** switch to **OFF**.

- The window displays as shown.



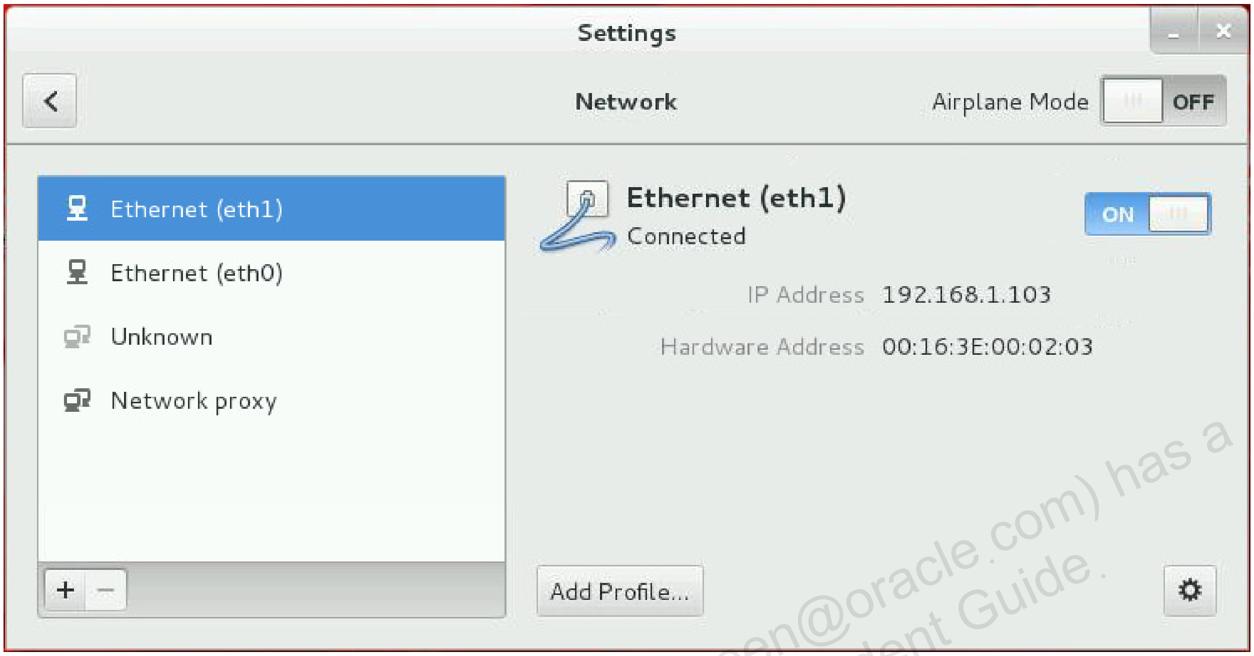
d. From a command window, use the `ip addr` command to display the status of the interfaces.

```
# ip addr
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

- Note that the `eth1` interface does not have IP addresses.

e. Select the **Ethernet (eth1)** entry on the left and then toggle the **ON/OFF** switch to **ON**.

- The window displays as shown.



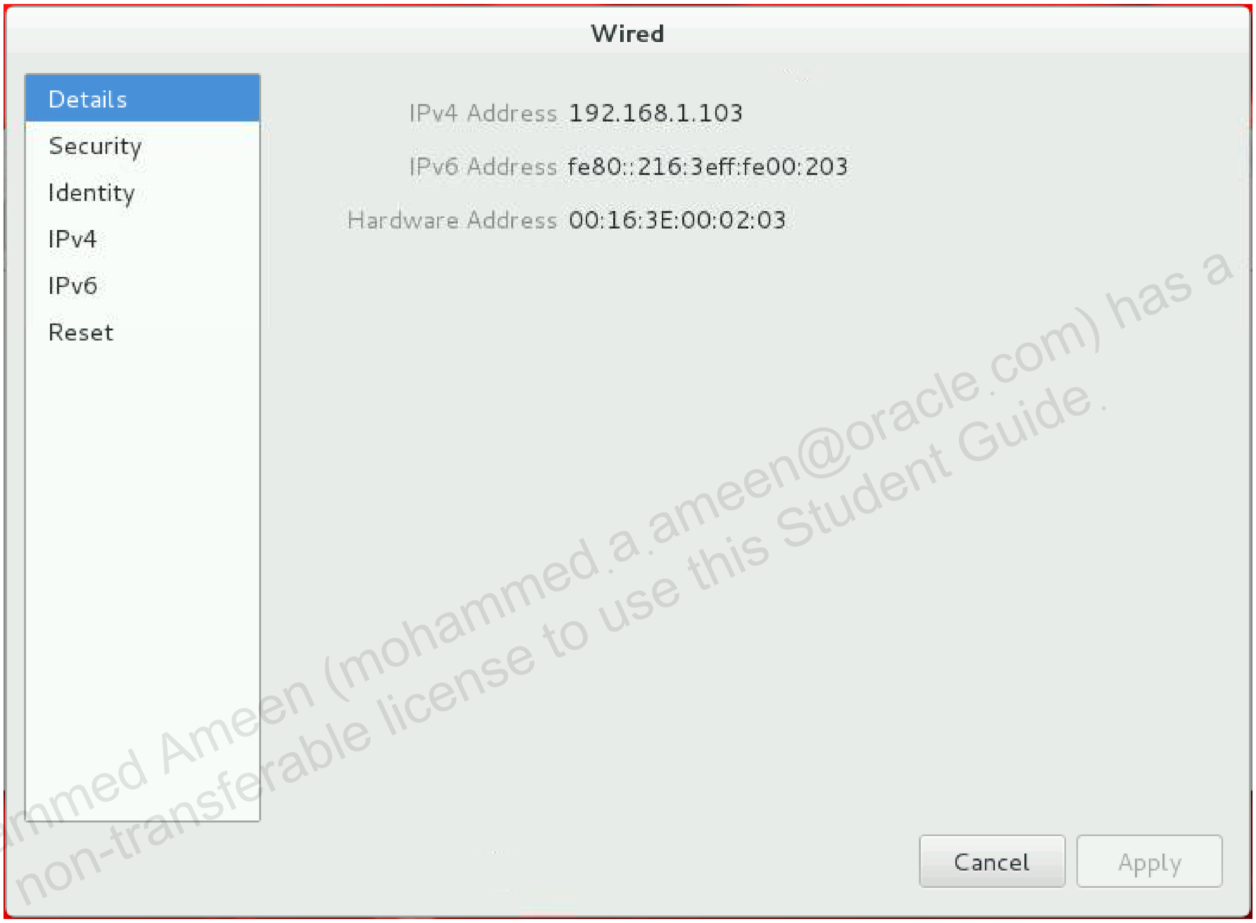
f. From a command window, use the `ip addr` command to display the status of the interfaces.

```
# ip addr
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

- Note that the `eth1` interface now has IP addresses.

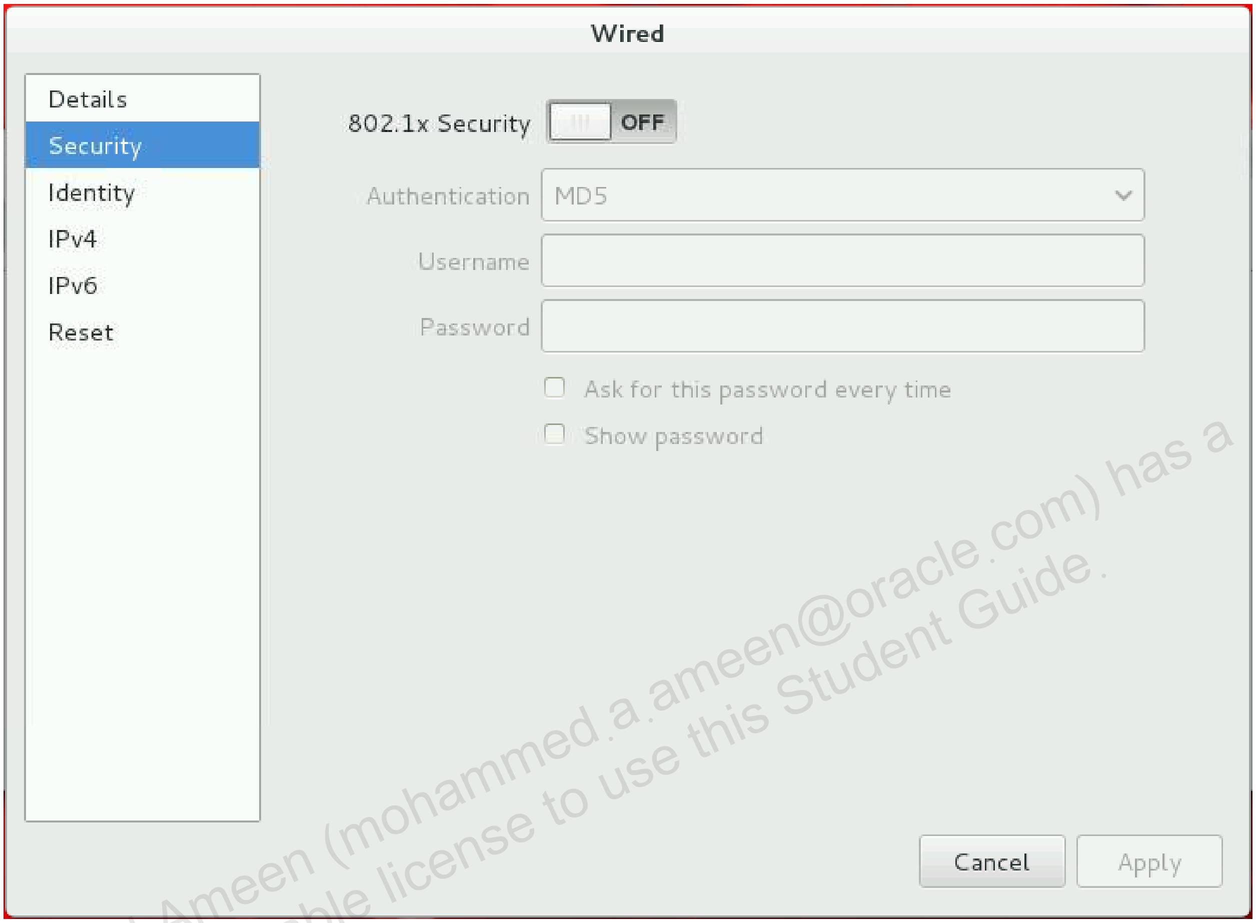
4. Use the **Network Settings** window to configure a network connection.
   a. Select the **Ethernet (eth1)** entry on the left and then select the gear icon in the lower right corner of the window as shown:

- The window displays as shown:



b. Select the **Security** entry on the left.

- The window displays as shown:



c. Toggle the **ON/OFF** switch to **ON** to enable 802.1x Security.

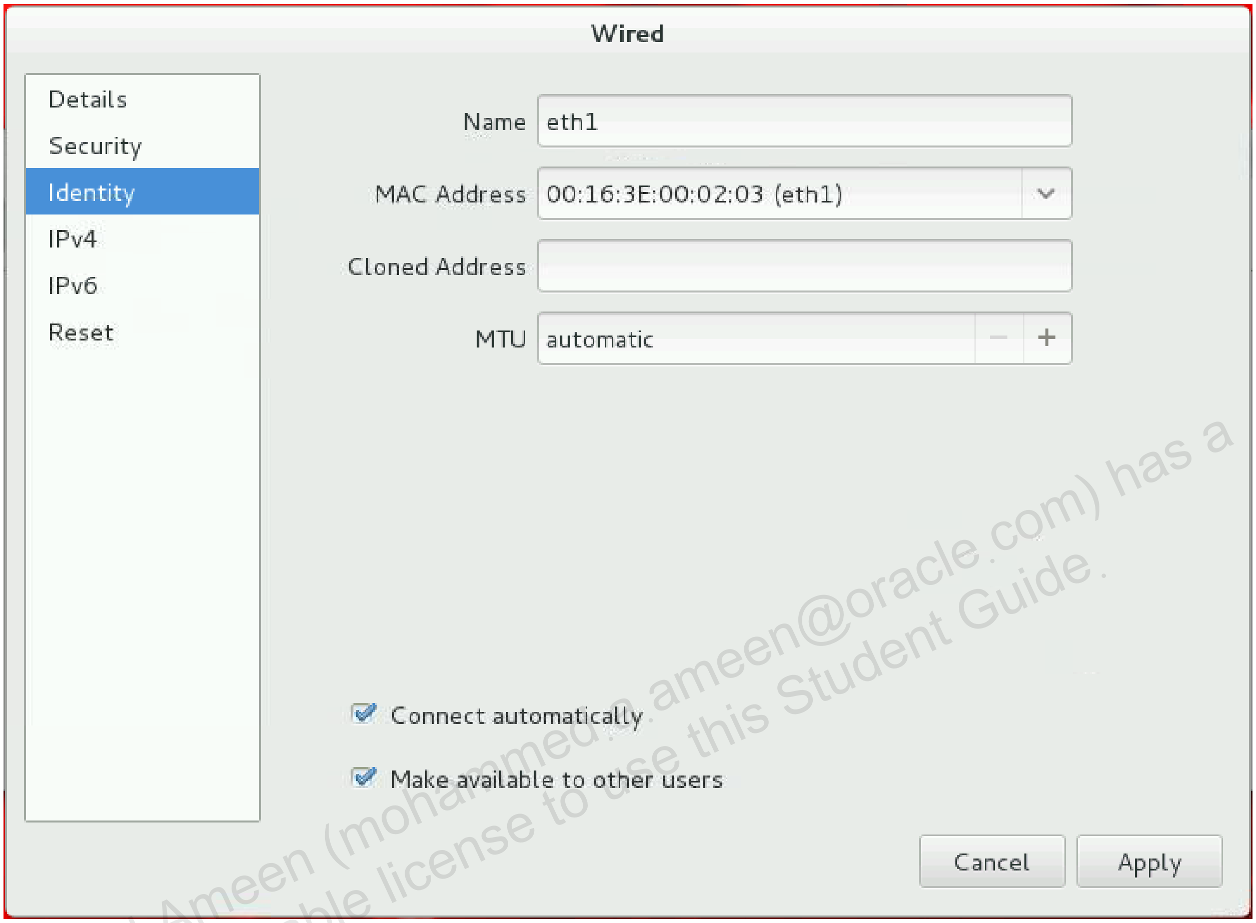
d. Select each Authentication option and view the configuration setting options:

- **MD5** (message-digest algorithm)
- **TLS** (Transport Layer Security)
- **FAST** (Flexible Authentication via Secure Tunneling)
- **Tunneled TLS**
- **PEAP** (Protected Extensible Authentication Protocol)

e. Toggle the **ON/OFF** switch to **OFF** to disable 802.1x Security.

f. Select the **Identity** entry on the left.

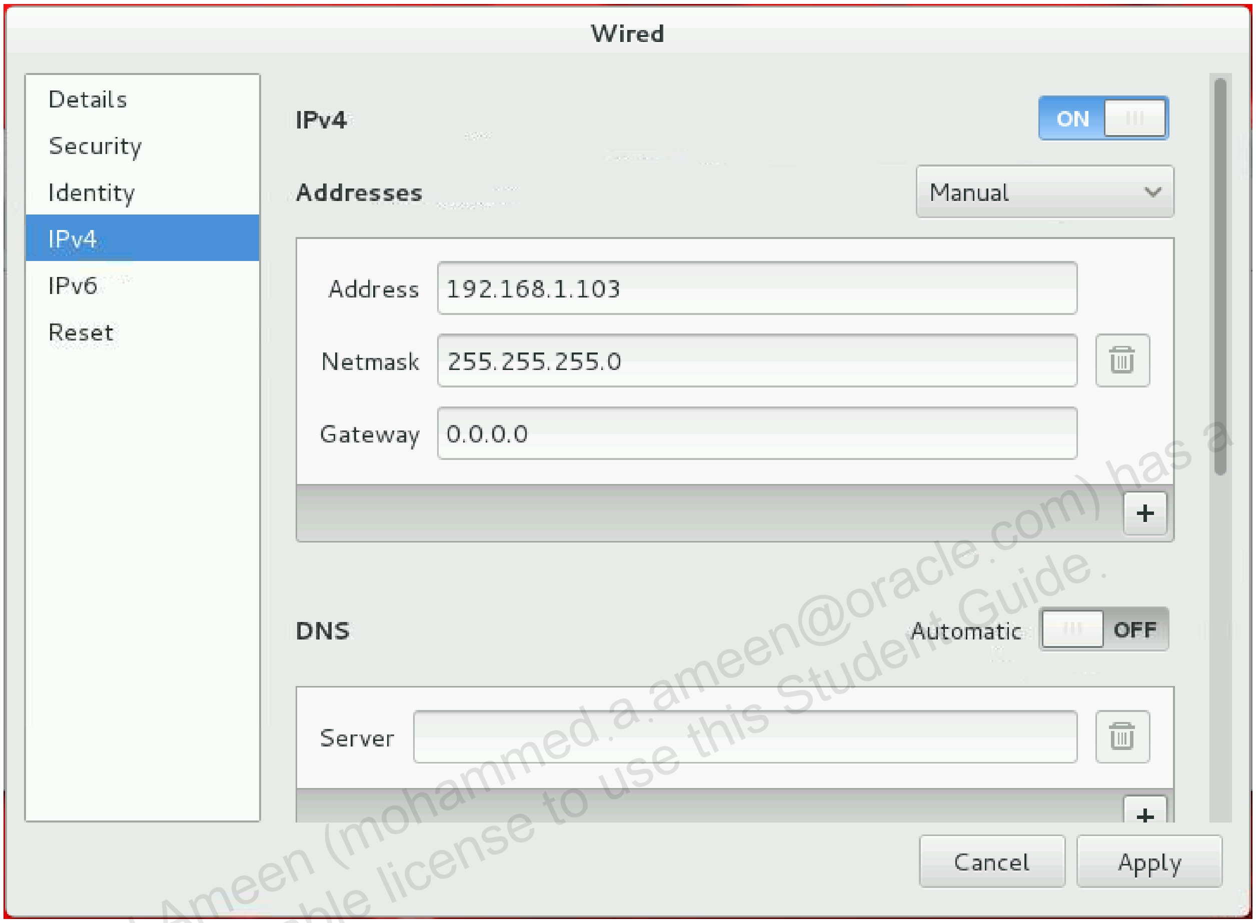
- The window displays as shown:



- Note the available settings on this window.

g. Select the **IPv4** entry on the left.

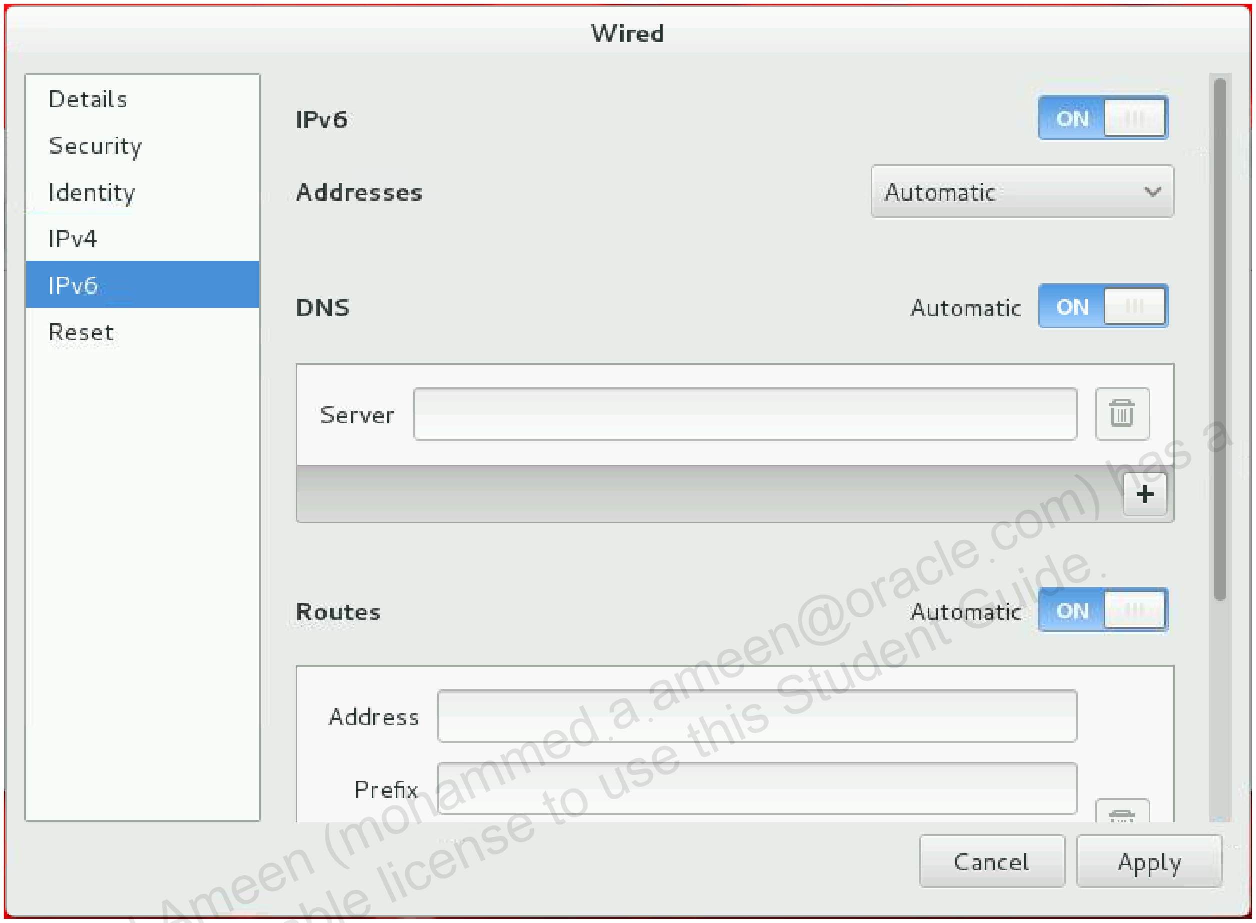
- The window displays as shown:



- Note the available settings on this window.
- Scroll down to view all settings.

h. Select the **IPv6** entry on the left.

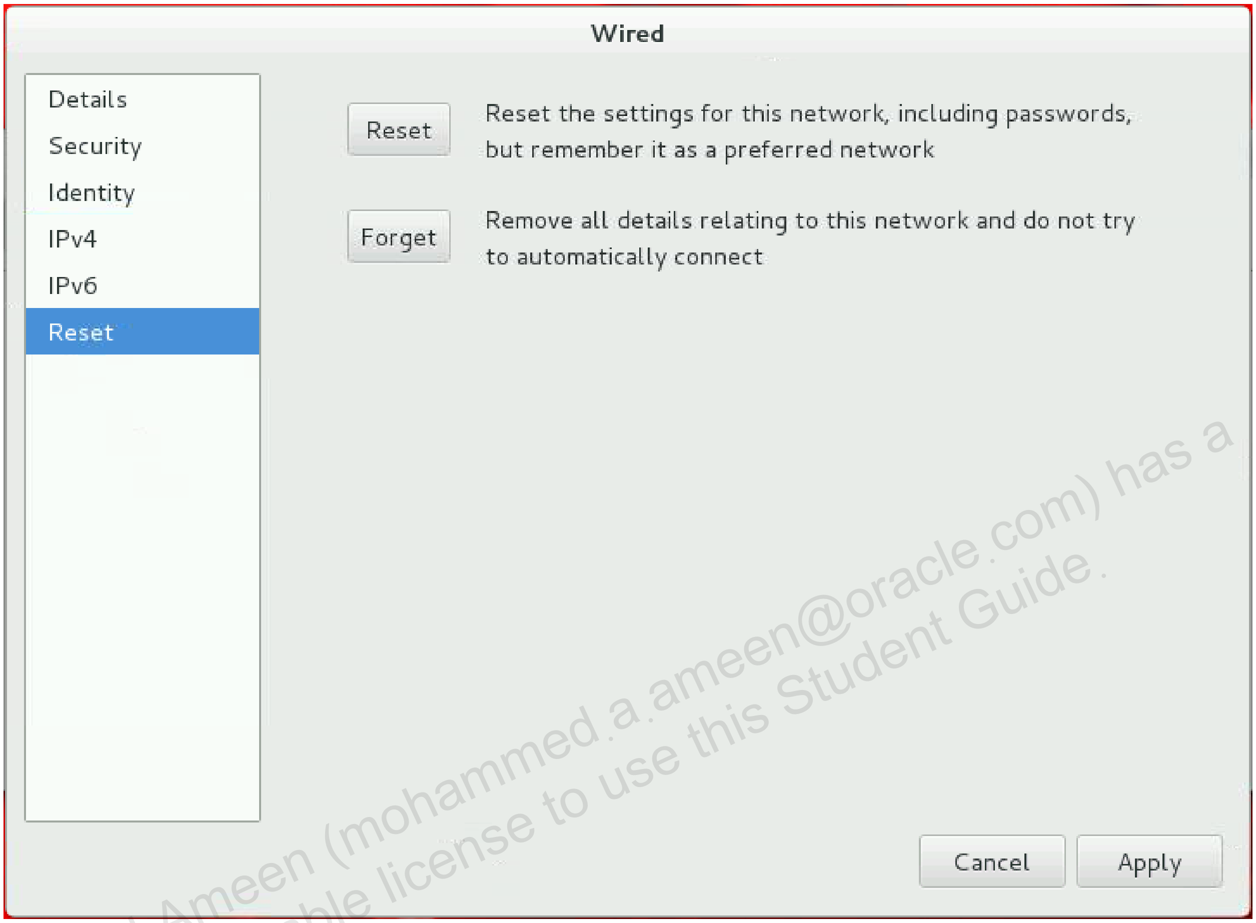
- The window displays as shown:



- Note the available settings on this window.
- Scroll down to view all settings.

i. Select the **Reset** entry on the left.

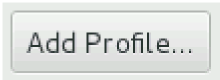
- The window displays as shown:



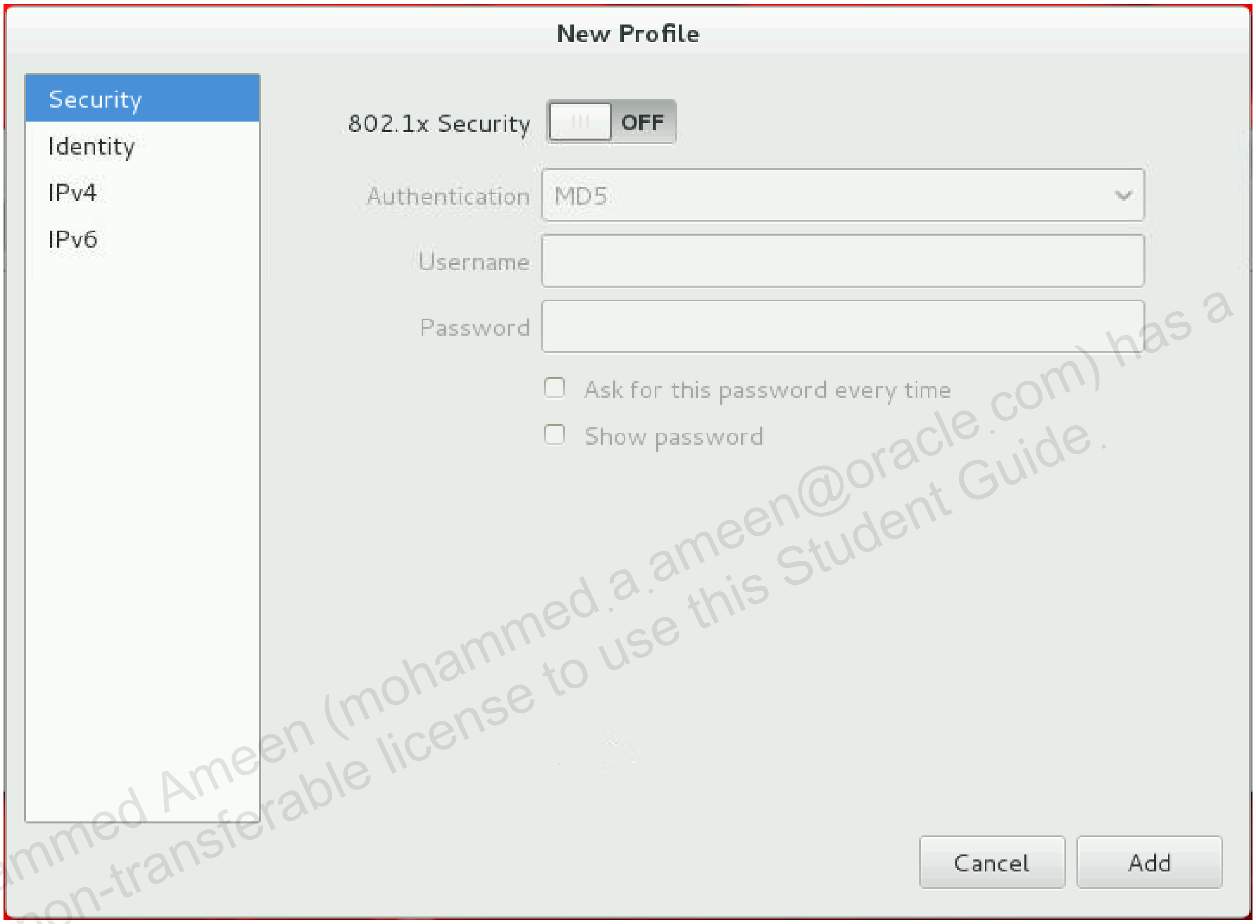
- Note the options on this window.

j. Select **Cancel** to close the window and return to the main **Network Settings** window.

5. Use the **Network Settings** window to add a profile.
   a. Select the **Add Profile** button as shown:

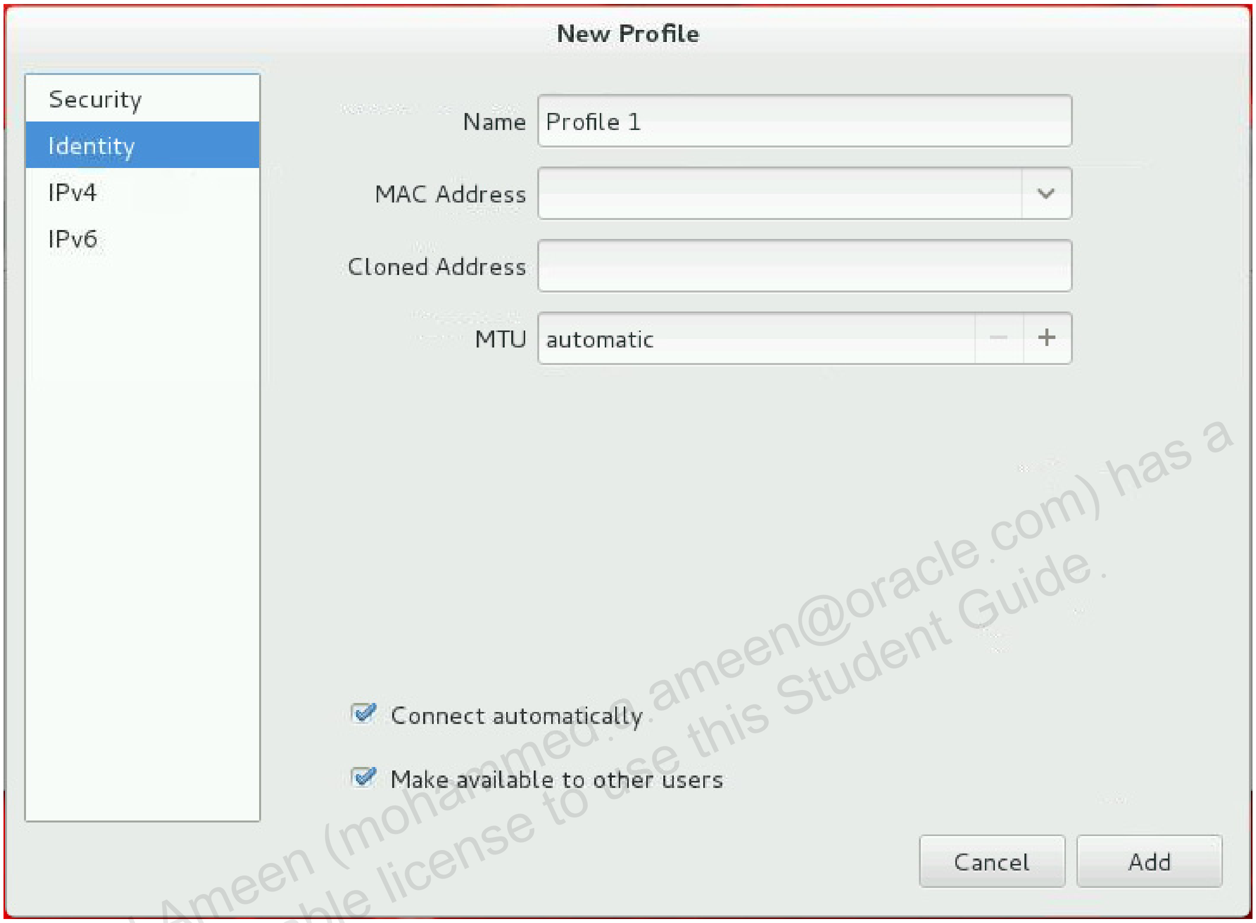


- The window displays as shown:



b. Select the **Identity** entry on the left.

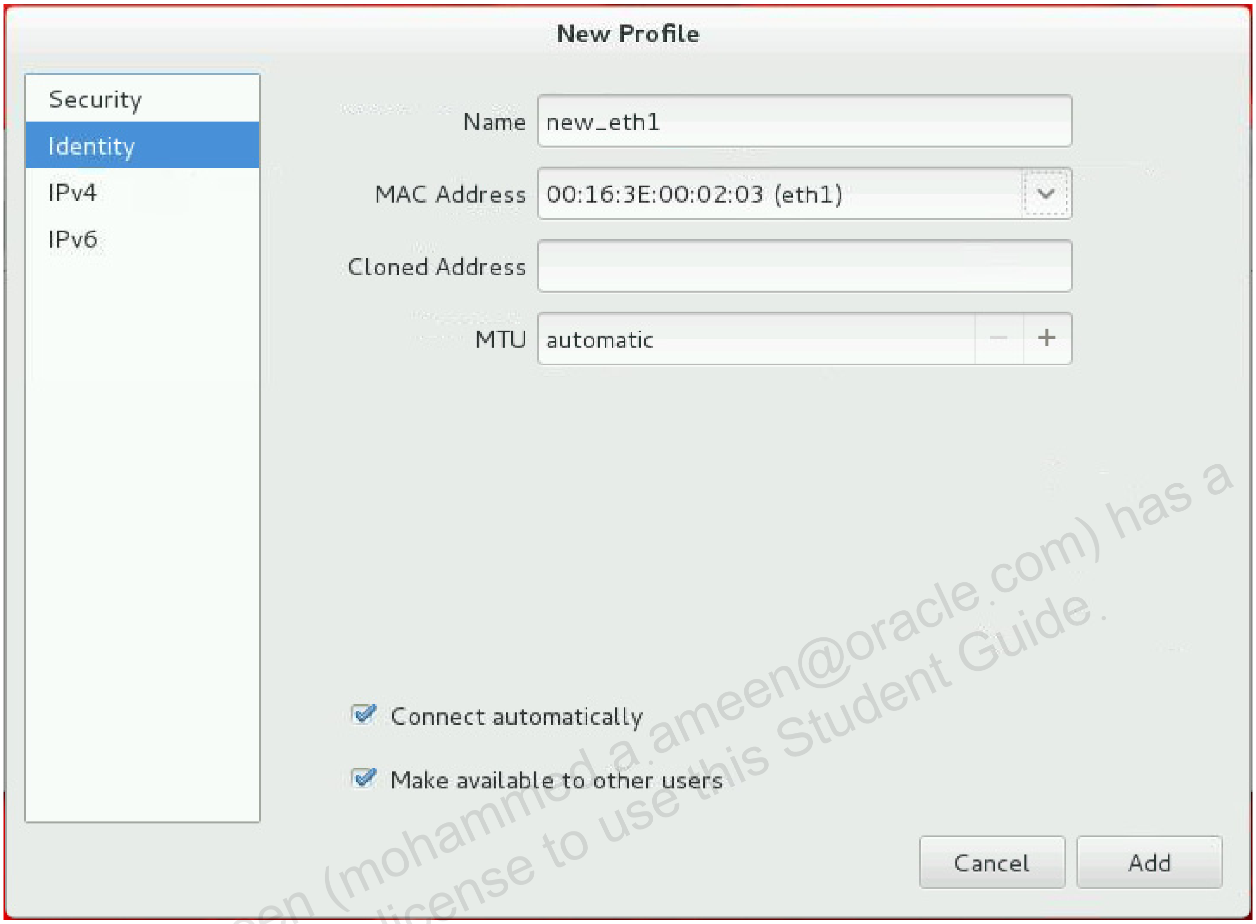
- The window displays as shown:



c. Provide the following Identity information:

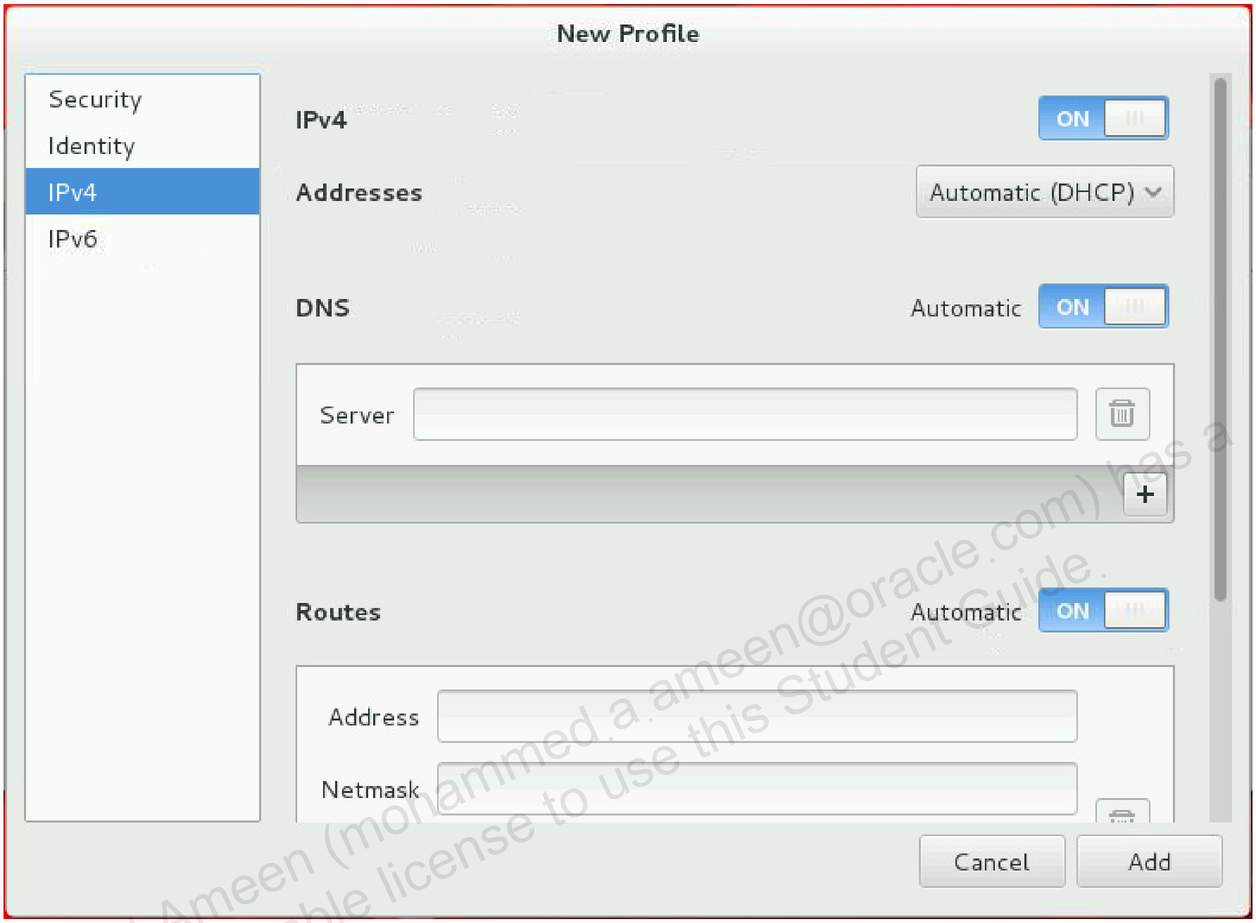
- **Name**: `new_eth1`
- **MAC Address**: Select `eth1` from the drop-down list.
- Accept remaining defaults.

- Your window appears as shown:



d. Select the **IPv4** entry on the left.

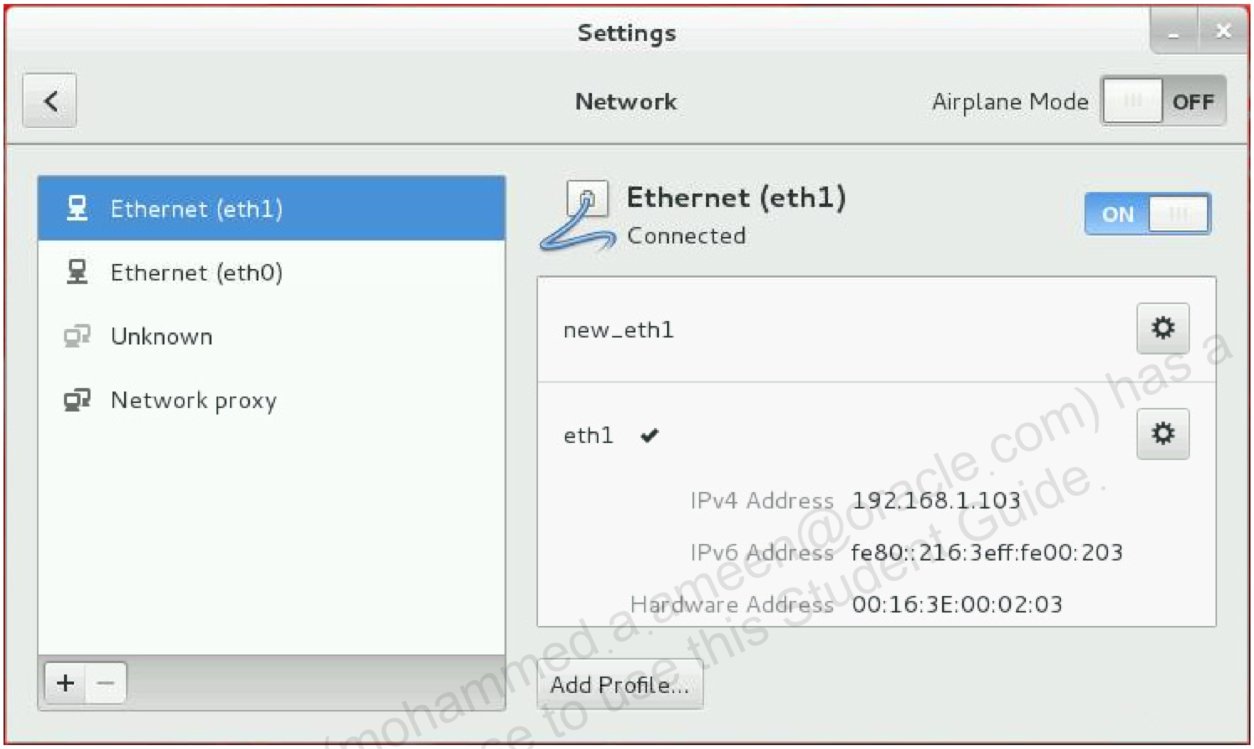
- The window displays as shown:



- Do not make any changes on this window.
- Use DHCP to obtain an IP address from the DHCP server, **dom0**.
- Dynamic Host Configuration Protocol (DHCP) is covered in another course.

e. Click **Add**.

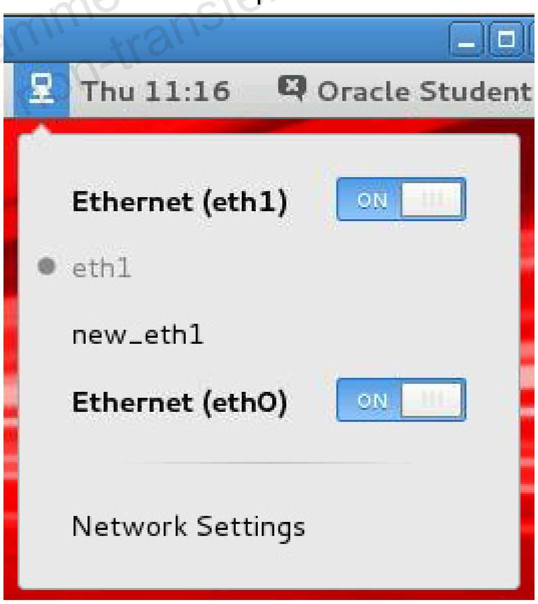
- The **New Profile** window closes.
- The **Network Settings** window displays as shown.
- Ensure the **Ethernet (eth1)** entry on the left is selected.



- Note that `eth1` has a check mark meaning it is currently the selected profile.

6. Use NetworkManager to select a different connection profile.

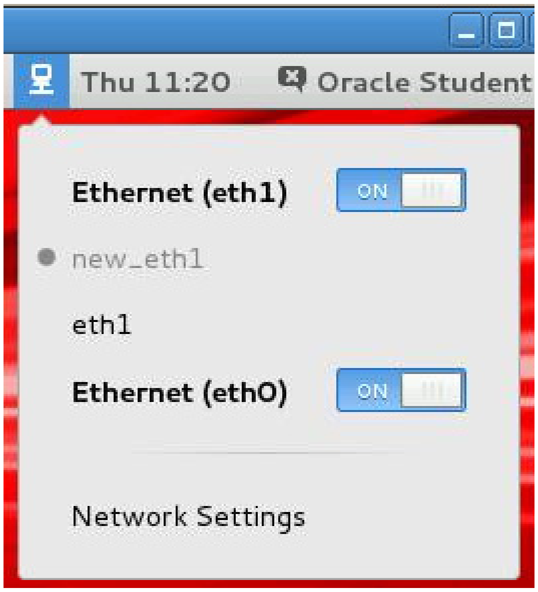
a. Select the computer screen icon to display the following:



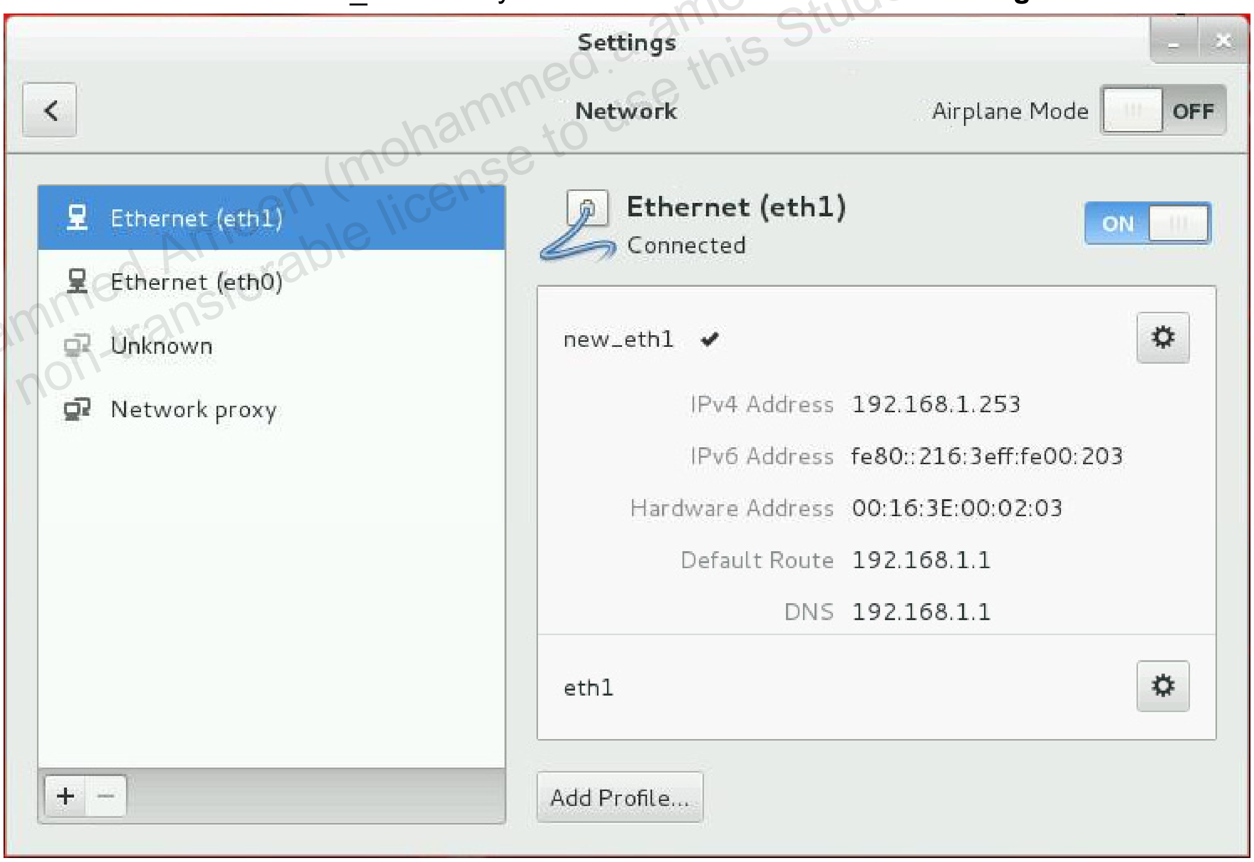
- Note that two connections are listed for the **Ethernet (eth1)** entry.
- Note that the `eth1` entry has a dot meaning it is currently selected.

b. Click the `new_eth1` entry.

- Select the computer screen icon again to show that `new_eth1` now has a dot meaning it is currently selected.



- Notice that the `new_eth1` entry is also selected in the **Network Settings** window.



c. From a command window, use the `ip addr` command to display the status of the interfaces.

```
# ip addr
```

```
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.254/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

- Note that the `eth1` interface now has different IPv4 address.
- In this example, the IPv4 address is `192.168.1.254`.
- This address might be different on your system because it was obtained from the DHCP server running on **dom0**.

7. View the network interface configuration file for the new connection profile.
   a. From a command window, use the `cd` command to change to the `/etc/sysconfig/network-scripts` directory.

```
# cd /etc/sysconfig/network-scripts
```

   b. Use the `ls` command to display files beginning with “`ifcfg`”.

```
# ls ifcfg*
ifcfg-eth0  ifcfg-eth1  ifcfg-lo  ifcfg-new_eth1
```

- Note that the `ifcfg-new_eth1` file exists for the new connection profile.

   c. Use the `cat` command to view the contents of the `ifcfg-new_eth1` file.

```
# cat ifcfg-new_eth1
HWADDR=00:16:3E:00:02:03
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=new_eth1
UUID=...
ONBOOT=yes
```

- Note that this Ethernet interface uses DHCP to obtain an IPv4 address:
  - `BOOTPROTO=dhcp`

8. Close the **Network Settings** window.
   Click the **X** in the upper-right corner of the **Network Settings** window.

Practice 15-3: Using the Network Connection Editor

Overview

In this practice, you:

- Ensure that the Network Connections package is installed
- Run the Network Connection Editor and view configuration settings for `eth0`

Assumptions

- You are connected to **host03** VM by using vncviewer.
- You are the `root` user on **host03** VM.

Tasks

1. Install the Network Connections package if necessary.
   - The Network Connections package name is `nm-connection-editor`.
   a. Use the `rpm` command to verify that the `nm-connection-editor` package is installed.

```
# rpm –q nm-connection-editor
nm-connection-editor-...
```

   - In this example, the package is installed.

   b. If the Network Connections package is not installed, use the `yum` command to install it.

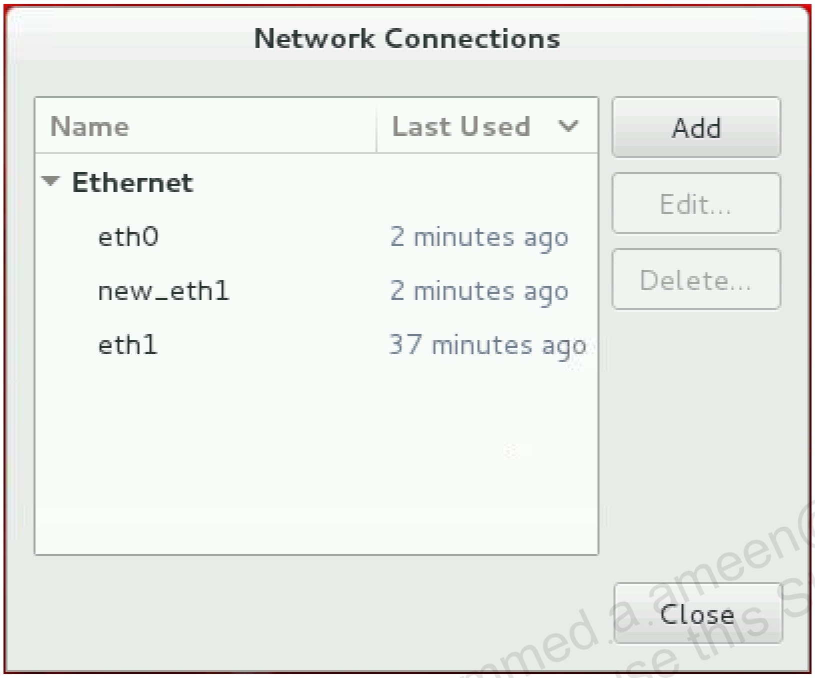
```
# yum install nm-connection-editor
...
```

2. Run the Network Connection Editor.

   a. Run the `nm-connection-editor` command.

```
# nm-connection-editor
```

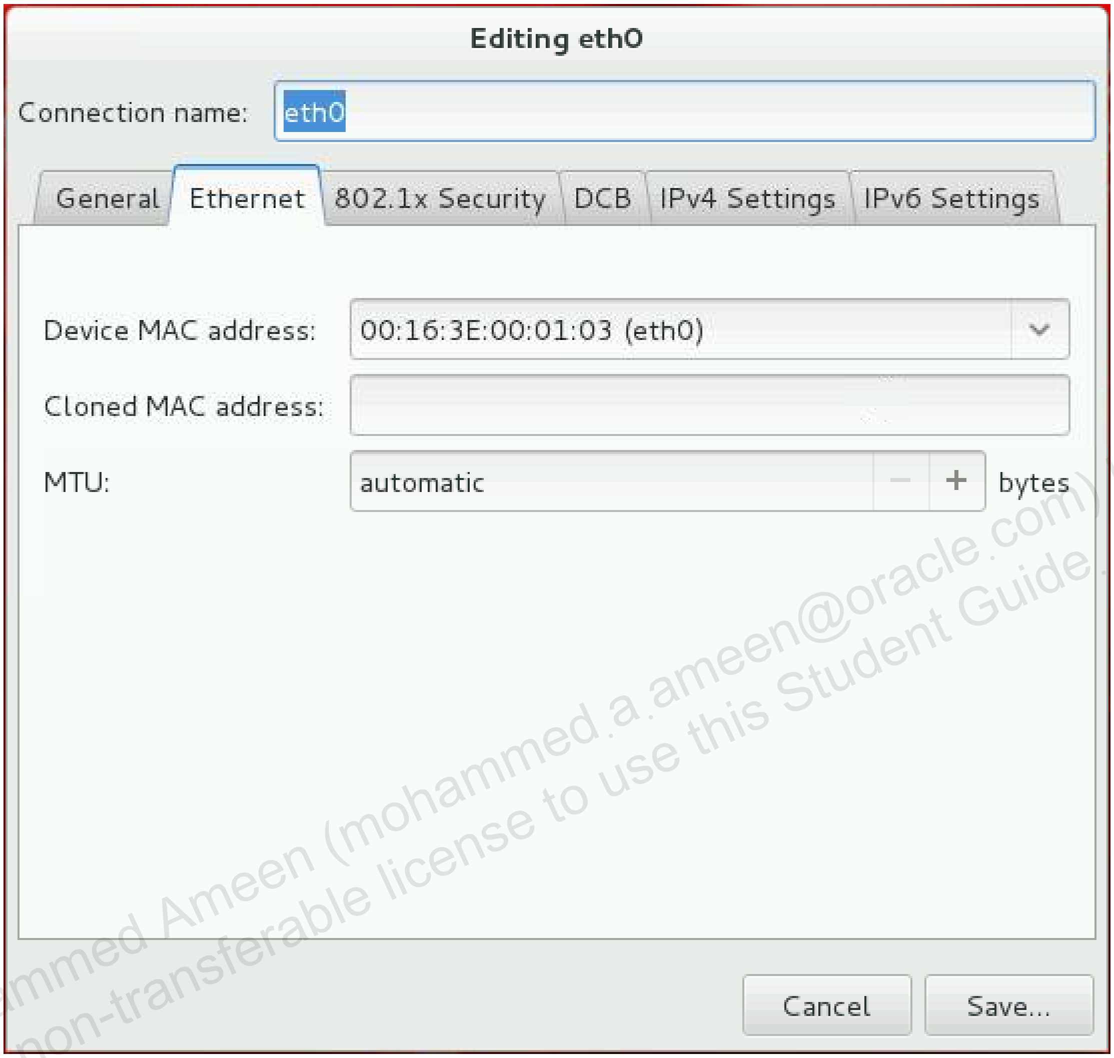
   - The window displays as shown:



- Note the three Ethernet connections, including the `new_eth1` connection profile.
- Note that you have the option to **Add** a new connection, or **Edit** and **Delete** an existing connection.

b. Select the `eth0` connection and then select **Edit**.

- The window displays as shown:



- Note that this is the same window that appeared when you configured the network during the installation of Oracle Linux 7.

c. Select the remaining tabs to view the configuration options for each window.

- **General**
- **802.1x Security**
- **DCB**
- **IPv4 Settings**
- **IPv6 Settings**

d. Select **Cancel** to close the editing window.

e. Select **Close** to close the Network Connections Editor window.

Practice 15-4: Using the nmcli Utility

Overview

In this practice, you use the nmcli command-line utility to view network status information, change the system host name, change the logging level, disable and enable networking, view connection information, add edit and delete a connection profile, and view device status information.

Assumptions

You are the root user on **host03** VM.

Tasks

1. Run the nmcli command without any options or arguments.

```
# nmcli
Usage: nmlci OPTIONS OBJECT { COMMAND | help }
OPTIONS
  -t[erse]                     terse output
  -p[retty]                    pretty output
  -m[ode] tabular|multiline    output mode
...
OBJECT
  g[eneral]     NetworkManager's general status and operations
  n[etworking]  overall networking control
  r[adio]       NetworkManager radio switches
  c[onnection]  NetworkManager's connections
  d[evice]      devices managed by NetworkManager
```

- Note that a number of options are available:
  - -t|--terse: This mode is designed and suitable for script processing.
  - -p|--pretty: This mode produces easily readable output with header.
  - -m|--mode tabular|multiline: Produces output in table format or in multiple lines.
- Refer to the nmcli(1) man page for a description of all options.
- Note that there are five different objects for the nmcli command.

2. Run the nmcli general object commands.
   a. Run the nmcli general help command.

```
# nmcli general help
Usage: nmcli general { COMMAND | help }

COMMAND := { status | hostname | permissions | logging }
...
```

   - Note that the nmcli general object provides four commands.

b. Run the `nmcli general status` command.

- Note that "`status`" is the default. That is, you can omit this argument.

```
# nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          enabled  enabled  enabled  disabled
```

- Note that the network status is "`connected`" with "`full`" connectivity.
- Full connectivity means the host is connected to a network and has full access to the Internet

c. Use the `systemctl` command to stop the NetworkManager service.

```
# systemctl stop NetworkManager
```

d. Run the `nmcli general status` command.

```
# nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
unknown    unknown       unknown  unknown  unknown  unknown
```

- With the NetworkManager service stopped, all columns are "`unknown`".

e. Run the `nmcli general hostname` command.

```
# nmcli general hostname
```

- This command reports nothing when the NetworkManager service is stopped.

f. Use the `systemctl` command to start the NetworkManager service.

```
# systemctl start NetworkManager
```

g. Run the `nmcli general hostname` command.

```
# nmcli general hostname
host03.example.com
```

- This command reports the host name when the NetworkManager service is running.
- You can also use this `nmcli general hostname` command to change the hostname.

h. Run the `nmcli general hostname` command and change the host name to "`myhost`".

```
# nmcli general hostname myhost
```

- The host name is stored in the `/etc/hostname` file.

i. Use the `cat` command to view the contents of the `/etc/hostname` file.

```
# cat /etc/hostname
myhost
```

j. Run the `nmcli general hostname` command and change the host name back to "`host03.example.com`".

```
# nmcli general hostname host03.example.com
```

k. Use the `cat` command to view the contents of the `/etc/hostname` file.

```
# cat /etc/hostname
host03.example.com
```

l. Run the `nmcli general permissions` command.

- In this example, all permissions are set to “yes” meaning you can enable and disable networking and modify all connections and settings.

```
# nmcli general permissions
PERMISSION                                              VALUE
org.freedesktop.NetworkManager.enable-disable-network   yes
org.freedesktop.NetworkManager.enable-disable-wifi      yes
org.freedesktop.NetworkManager.enable-disable-wwan      yes
org.freedesktop.NetworkManager.enable-disable-wimax     yes
org.freedesktop.NetworkManager.sleep-wake               yes
org.freedesktop.NetworkManager.network-control          yes
org.freedesktop.NetworkManager.wifi.share.protected     yes
org.freedesktop.NetworkManager.wifi.share.open          yes
org.freedesktop.NetworkManager.settings.modify.system   yes
org.freedesktop.NetworkManager.settings.modify.own      yes
org.freedesktop.NetworkManager.settings.modify.hostname yes
```

m. Run the `nmcli general logging` command.
   - With no arguments, this command shows the current logging level by domain.

```
# nmcli general logging
LEVEL  DOMAINS

INFO   PLATFORM,RFKIL,ETHER,WIFI,BT,MB,DHCP4,DHCP6,PPP,IP4,IP6,
AUTOIP4,DNS,VPN,SHARING,SUPPLICANT,AGENTS,SETTINGS,SUSPEND,CORE,
DEVICE,OLPC,WIMAX,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,BRIDGE,TEAM
,CONCHECK,DCB
```

   - In this example, the logging level is `INFO` for all domains.
   - Refer to the `NetworkManager.conf(5)` man page for information on logging levels and domain descriptions.

n. Use the `nmcli general logging` command to change the logging level to `DEBUG` for the `IP4` domain.

```
# nmcli general logging level DEBUG domains IP4
```

o. Run the `nmcli general logging` command to show the current logging level.

```
# nmcli general logging
LEVEL  DOMAINS
DEBUG  IP4
```

p. Use the `nmcli general logging` command to change the logging level to `INFO` for the `ALL` domains.
   - This command returns the logging level to the default setting.

```
# nmcli general logging level INFO domains ALL
```

3. Run the `nmcli networking` object commands.

   a. Run the `nmcli networking help` command.

```
# nmcli networking help
```

```
Usage: nmcli networking { COMMAND | help }

COMMAND := { [ on | off | connectivity ] }
...
```

- Note that the `nmcli networking` object provides three commands.

b. Run the `nmcli networking` command with no options or arguments to show the networking status.

```
# nmcli networking
enabled
```

- Note that the status is `enabled`.

c. Run the `nmcli networking off` command to disable networking.

```
# nmcli networking off
```

d. Run the `nmcli networking` command with no options or arguments to show the networking status.

```
# nmcli networking
disabled
```

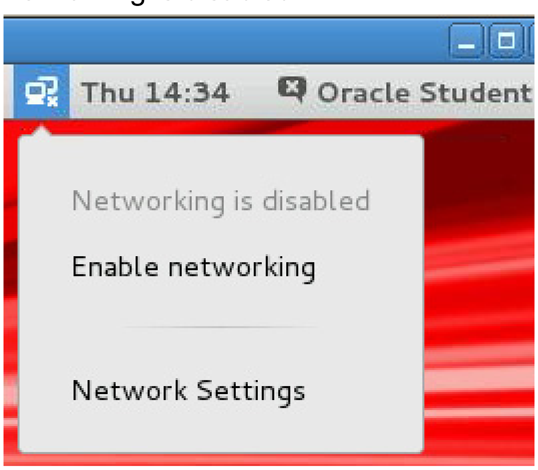
- Note that the status is `disabled`.

e. Use the `ip addr` command to display your available network interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc ... DOWN
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc ...  DOWN
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

- Note that both Ethernet interfaces are **DOWN** and have no IP addresses.

f. Select the computer screens icon on the GNOME notification area to show that networking is disabled.



- Note that you can select **Enable networking** from this screen.

g. Run the `nmcli networking on` command to enable networking.

```
# nmcli networking on
```

h. Run the `nmcli networking` command with no options or arguments to show the networking status.

```
# nmcli networking
enabled
```

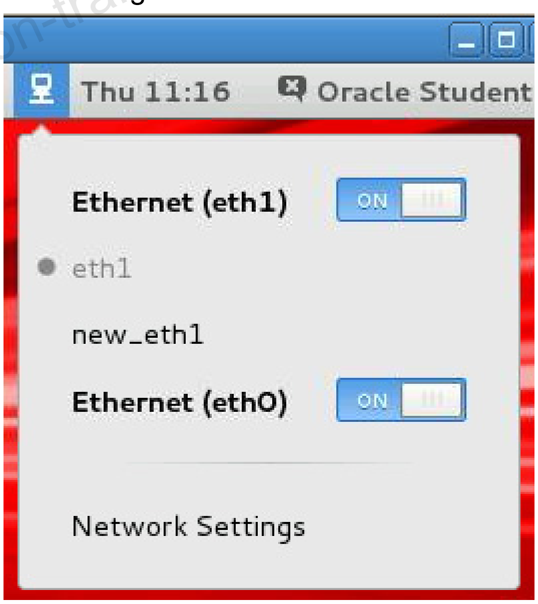
- Note that the status is now `enabled`.

i. Use the `ip addr` command to display the status of the interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.103/24 brd 192.0.2.255 scope global eth0
    inet6 ...
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

- Note that both `eth0` and `eth1` are now **UP** and have IP addresses.

j. Select the computer screen icon on the GNOME notification area to show that networking is enabled:



k. Run the `nmcli networking connectivity` command to show the network connectivity state.

- Include the `check` argument.
- Without the `check` argument, the command displays the most recent known connectivity state without re-checking.

```
# nmcli networking connectivity check
full
```

- Connectivity state `full` means the host is connected to a network and has full access to the Internet.

4. Run the `nmcli radio` object commands.

a. Run the `nmcli radio help` command.

```
# nmcli radio help
Usage: nmcli radio { COMMAND | help }

COMMAND := { [ all | wifi | wwan | wimax ] }
...
```

b. Run the `nmcli radio` command with no options or arguments to show radio switches status.

```
# nmcli radio
WIFI-HW  WIFI     WWAN-HW  WWAN
enabled  enabled  enabled  enabled
```

- Note that all switches are `enabled`.

c. Run the `nmcli radio wifi off` command to disable the WIFI radio switch.

```
# nmcli radio wifi off
```

d. Run the `nmcli radio` command to show radio switches status.

```
# nmcli radio
WIFI-HW  WIFI      WWAN-HW  WWAN
enabled  disabled  enabled  enabled
```

- Note that the WIFI switch is `disabled`.

e. Run the `nmcli radio wifi on` command to enable the WIFI radio switch.

- Run the `nmcli radio` command to show radio switches status.

```
# nmcli radio wifi on
# nmcli radio
WIFI-HW  WIFI     WWAN-HW  WWAN
enabled  enabled  enabled  enabled
```

- Note that all switches are `enabled`.

5. Run the `nmcli connection` object commands.

a. Run the `nmcli connection help` command.

```
# nmcli connection help
Usage: nmcli connection { COMMAND | help }
```

```
COMMAND := { show | up | down | add | modify | edit | delete |
reload | load }
...
```

- Note that the `nmcli connection` object provides nine commands.

b. Run the `nmcli connection show` command.
   - This command lists all the connection profiles.

```
# nmcli connection show
NAME         UUID    TYPE               DEVICE
eth0         ...     802-3-ethernet     eth0
new_eth1     ...     802-3-ethernet     --
eth1         ...     802-3-ethernet     eth1
```

c. Run the `nmcli connection show` command with the `--active` argument.
   - This command lists only active profiles.

```
# nmcli connection show --active
NAME         UUID    TYPE               DEVICE
eth0         ...     802-3-ethernet     eth0
eth1         ...     802-3-ethernet     eth1
```

d. Run the `nmcli connection show id eth0` command.
   - This command shows detailed information for a specific connection, `eth0`.
   - Only partial output is shown.

```
# nmcli connection show id eth0
connection.id:                 eth0
connection.uuid:               ...
connection.interface-name:     --
connection.type:               802-3-ethernet
connection.autoconnect:        yes
...
802-3-ethernet.mac-address:    00:16:3E:00:01:03
...
802-3-ethernet.mtu             auto
...
ipv4.method:                   manual
ipv4.dns:                      152.68.154.3, 10.216.106.3
ipv4.dns-search:               example.com
ipv4.addresses:                { ip = 192.0.2.103/24, gw = ... }
...
```

e. Run the `nmcli connection up id new_eth1` command.
   - This command activates a specific connection, `new_eth1`.

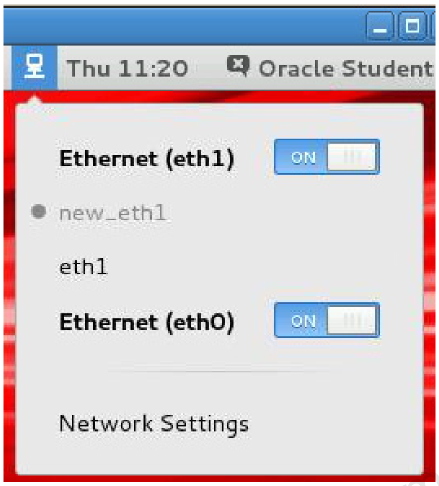
```
# nmcli connection up id new_eth1
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/4)
```

f. Run the `nmcli connection show` command.

```
# nmcli connection show
NAME          UUID     TYPE              DEVICE
eth0          ...      802-3-ethernet    eth0
new_eth1      ...      802-3-ethernet    eth1
eth1          ...      802-3-ethernet    --
```

- Note that the `new_eth1` connection profile is now active and `eth1` is not.

g. Select the computer screen icon in the GNOME notification area to display the following:



- Note that `new_eth1` has a dot meaning it is currently selected.

h. Run the `nmcli connection down id new_eth1` command.

- This command deactivates a specific connection, `new_eth1`.

```
# nmcli connection down id new_eth1
```

i. Run the `nmcli connection show` command.

```
# nmcli connection show
NAME          UUID     TYPE              DEVICE
eth0          ...      802-3-ethernet    eth0
new_eth1      ...      802-3-ethernet    --
eth1          ...      802-3-ethernet    eth1
```

- Note that the `new_eth1` connection profile is no longer active and `eth1` is active.
- The `eth1` connection automatically starts because the `autoconnect` parameter is set to “`yes`”.

j. Run the `nmcli connection show id eth1` command.

- Pipe the output to `grep` and search for the “`autoconnect`” string.

```
# nmcli connection show id eth1 | grep autoconnect
connection.autoconnect:        yes
```

- Note that the `autoconnect` parameter is set to “`yes`”.

6. Use the `nmcli connection` object to add, edit, and delete a connection profile.

   a. Run the `nmcli connection add` command to add a new connection profile. Use the following parameters:

      - Connection name (`con-name`): `new_eth0`
      - Interface name (`ifname`): `eth0`
      - Type (`type`): `ethernet`
      - IPv4 address (`ip4`): `192.0.2.111/24`
      - IPv4 Gateway (`gw4`): `192.0.2.1`

```
# nmcli connection add con-name new_eth0 ifname eth0 type
ethernet ip4 192.0.2.111/24 gw4 192.0.2.1
Connection 'new_eth0' (...) successfully added.
```

   b. Run the `nmcli connection show` command.

```
# nmcli connection show
NAME          UUID     TYPE              DEVICE
eth0          ...      802-3-ethernet    eth0
new_eth1      ...      802-3-ethernet    --
eth1          ...      802-3-ethernet    eth1
new_eth0      ...      802-3-ethernet    --
```

      - Note that the `new_eth0` connection now exists.

   c. Use the `ls` command to view network interface configuration files in the `/etc/sysconfig/network-scripts` directory.

```
# ls /etc/sysconfig/network-scripts/ifcfg*
/etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-eth1
/etc/sysconfig/network-scripts/ifcfg-lo
/etc/sysconfig/network-scripts/ifcfg-new_eth0
/etc/sysconfig/network-scripts/ifcfg-new_eth1
```

      - Note that the `nmcli connection add` command created a network interface configuration file, `ifcfg-new_eth0`, for the new connection profile, `new_eth0`.

   d. Run the `nmcli connection edit` command to edit connection parameters for the `new_eth0` connection.

      - This command uses an interactive editor.

```
# nmcli connection edit new_eth0

===| nmcli interactive connection editor |===

Editing existing '802-3-ethernet' connection: 'new_eth0'

Type 'help' or '?' for available commands.
Type 'describe [<settings>.<prop>]' for detailed property
description.
```

```
You may edit the following settings: connection, 802-3-ethernet
(ethernet), 802-1x, ipv4, ipv6, dcb
nmcli>
```

e. Enter `help` at the `nmcli>` prompt to list the available commands.
- Only partial output is shown.

```
nmcli> help
------------------------------------------------------------
---[ Main menu ]---
goto
remove    [<setting>.<prop>] | <prop> :: remove setting or ...
set       [<setting>.<prop> <value>]  :: set property value
describe  [<setting>.<prop>]          :: describe property
print     [all]                       :: print the connection
...
```

f. Enter `print` at the `nmcli>` prompt to print the connection profile details.
- Only partial output is shown.
- Note that the output is similar to the “`nmcli connection show id new_eth0`” command output.

```
nmcli> print
=============================================================
           Connection profile details (new_eth0)
=============================================================
connection.id:                 new_eth0
connection.uuid:               ...
connection.interface-name:     eth0
connection.type:               802-3-ethernet
connection.autoconnect:        yes
...
-------------------------------------------------------------
...
802-3-ethernet.mac-address:    --
...
802-3-ethernet.mtu             auto
...
-------------------------------------------------------------
ipv4.method:                   manual
ipv4.dns:
ipv4.dns-search:
ipv4.addresses:                { ip = 192.0.2.111/24, gw = ... }
...
```

- Note that the `connection.autoconnect` parameter is set to “`yes`”.

g. Use the `set` command from the `nmcli>` prompt to change the following parameter:

- `connection.autoconnect=no`

```
nmcli> set connection.autoconnect no
```

h. Use the `quit` command from the `nmcli>` prompt to exit the interactive editor.

- Answer **n** when prompted. Do not exit the `nmcli` interactive editor.

```
nmcli> quit
The connection is not saved. Do you really want to quit? [y/n] n
```

i. Use the `save` command from the `nmcli>` prompt to save your change.

```
nmcli> save
Connection 'new_eth0' (...) successfully updated.
```

j. Use the `quit` command from the `nmcli>` prompt to exit the interactive editor.

```
nmcli> quit
```

k. Run the `nmcli connection show id new_eth0` command.

- Pipe the output to `grep` and search for the "`autoconnect`" string.

```
# nmcli connection show id new_eth0 | grep autoconnect
connection.autoconnect:      no
```

- Note that the `connection.autoconnect` parameter is set to "`no`".

l. Run the `nmcli connection modify` command to modify a connection parameter for the `new_eth0` connection.

- This command does not use an interactive editor.
- Set the `ipv4.dns` parameter to `152.68.154.3`.

```
# nmcli connection modify new_eth0 ipv4.dns 152.68.154.3
```

m. Run the `nmcli connection show id new_eth0` command.

- Pipe the output to `grep` and search for the "`ipv4.dns`" string.

```
# nmcli connection show id new_eth0 | grep ipv4.dns
ipv4.dns:                    152.68.154.3
ipv4.dns-search:
```

- Note that the `ipv4.dns` parameter is set to "`152.68.154.3`".

n. Run the `nmcli connection delete` command to delete the `new_eth0` connection profile.

```
# nmcli connection delete new_eth0
```

o. Run the `nmcli connection show` command.

```
# nmcli connection show
NAME        UUID     TYPE                DEVICE
eth0        ...      802-3-ethernet      eth0
new_eth1    ...      802-3-ethernet      --
eth1        ...      802-3-ethernet      eth1
```

- Note that the `new_eth0` connection no longer exists.

p. Use the `ls` command to view network interface configuration files in the `/etc/sysconfig/network-scripts` directory.

```
# ls /etc/sysconfig/network-scripts/ifcfg*
/etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-eth1
/etc/sysconfig/network-scripts/ifcfg-lo
/etc/sysconfig/network-scripts/ifcfg-new_eth1
```

- Note that the network interface configuration file for the `new_eth0` connection profile no longer exists.

7. Run the `nmcli device` object commands.

a. Run the `nmcli device help` command.

```
# nmcli device help
Usage: nmcli device { COMMAND | help }

COMMAND := { status | show | connect | disconnect | wifi }
...
```

- Note that the `nmcli device` object provides five commands.

b. Run the `nmcli device status` command.

- This command displays the status of all the devices.

```
# nmcli device status
DEVICE      TYPE        STATE       CONNECTION
eth0        ethernet    connected   eth0
eth1        ethernet    connected   eth1
lo          loopback    unmanaged   --
```

c. Run the `nmcli device disconnect eth1` command.

```
# nmcli device disconnect eth1
```

d. Run the `nmcli device status` command.

- The `status` command is the default for the `nmcli device` object and so it is not required.

```
# nmcli device
DEVICE      TYPE        STATE          CONNECTION
eth0        ethernet    connected      eth0
eth1        ethernet    disconnected   --
lo          loopback    unmanaged      --
```

- Note the change in the `eth1` device.

e. Run the `nmcli device connect eth1` command to reconnect the `eth1` device.

```
# nmcli device connect eth1
Device 'eth1' successfully activated with '...'.
```

Practice 15-5: Using the `nmtui` Utility

Overview

In this practice, you use the `nmtui` text-based utility to view network connections.

Assumptions

You are the `root` user on **host03** VM.

Tasks

1. Install `NetworkManager-tui` if necessary.
   - The `numtui` utility is provided by the `NetworkManager-tui` software package.

   a. Use the `rpm` command to verify that the `NetworkManager-tui` package is installed.

   ```
   # rpm -qa | grep NetworkManager-tui
   NetworkManager-tui-...
   ```

   b. If the utility is not installed, use the `yum` command to install the package.

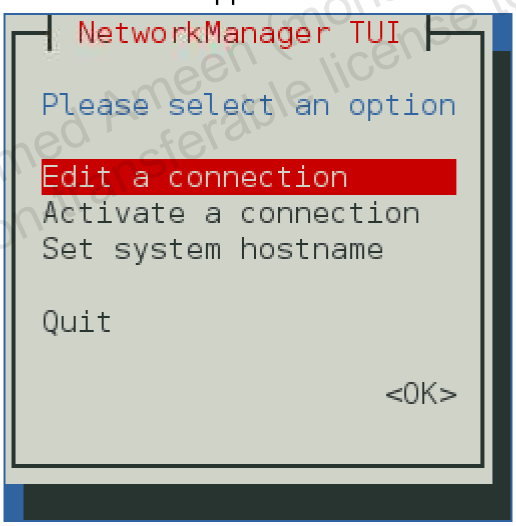
   ```
   # yum install NetworkManager-tui
   ...
   ```

2. Use `nmtui` to configure network interfaces.

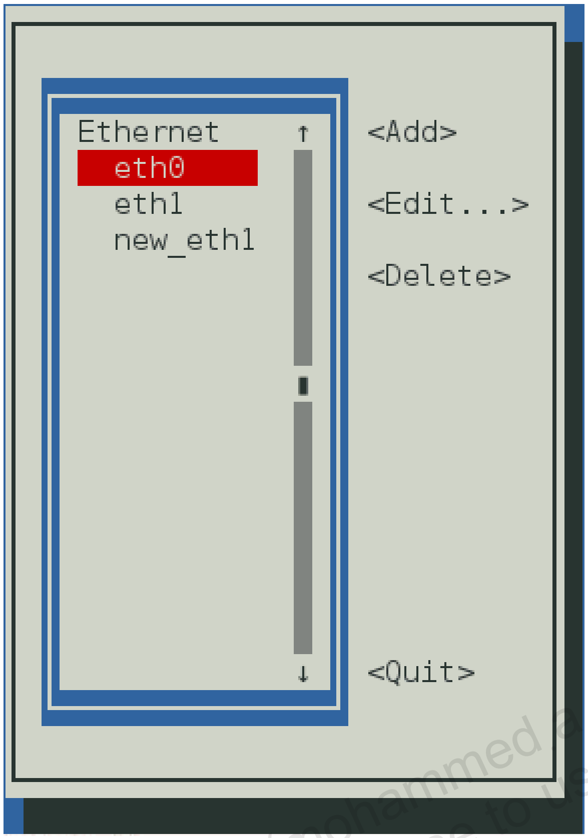
   a. Enter `nmtui` from the command line.

   ```
   # nmtui
   ```

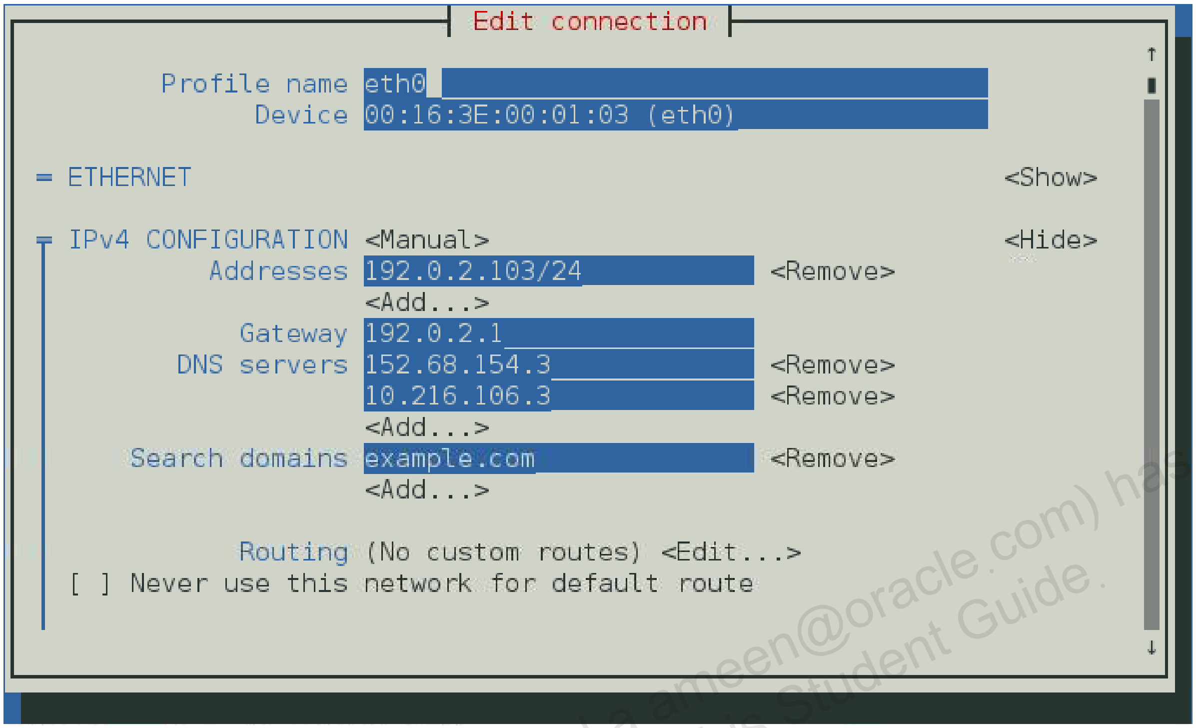
   - The screen appears as follows:



b. Use the up/down arrows to select **Edit a connection**. Use the TAB key to select **<OK>** and press Enter. The screen appears as follows:

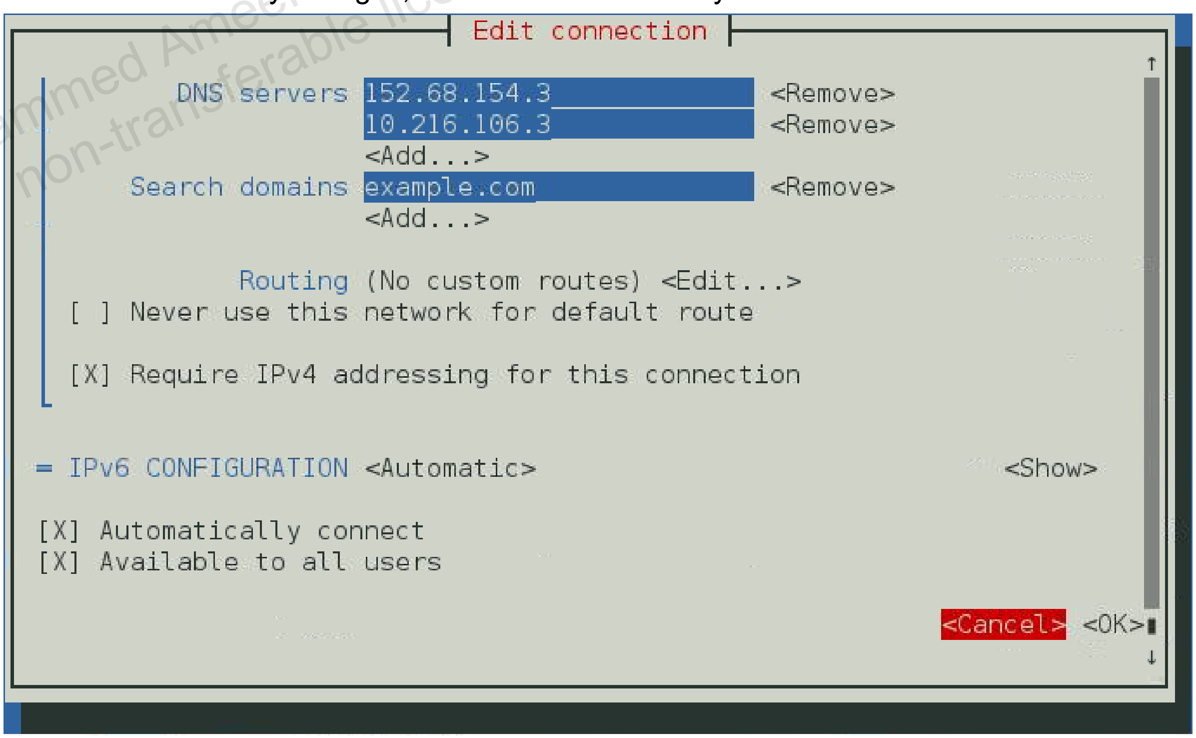


c. Use the up/down arrows to select **eth0**. Use the TAB key to select **<Edit…>** and press Enter. The screen appears as follows:



- Note that the information displayed is included in the configuration file, `/etc/sysconfig/network-scripts/ifcfg-eth0`.
- Any updates made from this screen are written to the configuration file.

d. Do not make any changes; use the down arrow keys to select **<Cancel>** as shown.



e. With **<Cancel>** selected as shown, press Enter.
f. Use the TAB key to select **<Quit>** and press Enter to exit the `nmtui` utility.

Practice 15-6: Using the `ip` Utility

Overview

In this practice, you use the `ip` utility to view network device information, add edit and delete a link, view IP addresses, add and delete an IPv4 address to a network device, and view and flush the ARP cache.

Tasks

1. Run the `ip` command without any options or arguments.
   - Only partial output is shown.

```
# ip
Usage: ip [OPTIONS] OBJECT {COMMAND | help}
       ip [ -force ] -batch filename
where OBJECT := { link | addr | addrlabel | route | rule |
                  neigh | ntable | tunnel | tuntap | maddr |
                  mroute | mrule | monitor | xfrm | netns |
                  l2tp | tcp_metrics | token }
...
```

   - Note that a number of options are available:
     - `-b|-batch <filename>`: Read and invoke commands from `<filename>`.
     - `-V|-Version`: Print the version of the `ip` utility.
     - `-s|-stats|-statistics`: Output more information.
   - Refer to the `ip(8)` man page for a description of all options.
   - Note that there are 18 different objects for the `ip` command.
2. Run the `ip link` object commands.
   - A “link” is a network device.
   - The “`ip link`” command is used to display and configure network devices.

   a. Run the `ip link help` command.
      - Only partial output is shown.

```
# ip link help
Usage: ip link add [link DEV] [name] NAME
       ...
       ip link delete DEV type TYPE [ARGS]
       ...
       ip link set {dev DEVICE | group DEVGROUP} [{up | down}]
       ...
       ip link show [DEVICE | group GROUP] [up]
...
```

      - Note that the `ip link` object provides four commands:
        – The four commands are `add`, `delete`, `set`, and `show`.

   b. Run the `ip link show` command.

- This command shows the existing network devices.
- The `show` command is the default for the `ip link` object so it is not required.

```
# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

- Note that you have three network devices; two Ethernet devices (`eth0` and `eth1`) and one loopback device (`lo`).
- These are the same devices that are listed by the `nmcli device status` command.

c. Run the `nmcli device status` command.

- This command lists the same devices as the `ip link show` command.

```
# nmcli device
DEVICE      TYPE        STATE        CONNECTION
eth0        ethernet    connected    eth0
eth1        ethernet    connected    eth1
lo          loopback    unmanaged    --
```

3. Use the `ip link` utility to add, edit, and delete a link.
   - The `ip link add` command adds a virtual link.
   - The virtual link type can be any of the following: `vlan`, `veth`, `vcan`, `dummy`, `ifb`, `macvlan`, `can`, `bridge`, `ipoib`, `ip6tnl`, `ipip`, `sit`, or `vxlan`.
   - Refer to the `ip-link(8)` man page for more information on link types.
   - VLANs and advanced networking topics are covered in another course.

a. Run the `ip link add` command to add a new link. Use the following parameters:

- Physical device to operate on (`link`): `eth0`
- Name (`name`): `eth0.10`
- Type (`type`): `vlan`
- VLAN ID (`id`): `10`

```
# ip link add link eth0 name eth0.10 type vlan id 10
```

b. Run the `ip link show` command.

```
# ip link
...
4: eth0.10@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
```

- Note that you now have four network devices, including the new link `eth0.10`.

c. Run the `nmcli device status` command.

```
# nmcli device
```

```
DEVICE       TYPE        STATE         CONNECTION
eth0         ethernet    connected     eth0
eth1         ethernet    connected     eth1
eth0.10      vlan        disconnected  --
lo           loopback    unmanaged     --
```

- Note the `eth0.10` device is listed.

d. Run the `ip link set` command to change device attributes.

- Change the MTU for the `eth0.10` device to `1400`.

```
# ip link set eth0.10 mtu 1400
```

e. Run the `ip link show` command to show the `eth0.10` device.

- The `show` command is required when specifying a device as an argument.

```
# ip link show eth0.10
4: eth0.10@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1400 ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
```

- Note that the MTU is `1400`.

f. Run the `ip link delete` command to delete the `eth0.10` device.

```
# ip link delete eth0.10
```

g. Run the `ip link show` command.

```
# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

- Note that the `eth0.10` device no longer exists.

4. Run the `ip addr` object commands.

- The "`ip addr`" command is used to display and manage IP addresses on network devices.

a. Run the `ip addr help` command.

- Only partial output is shown.

```
# ip addr help
Usage: ip addr {add|change|replace} IFADDR dev STRING [ LIFE...
       ip addr del IFADDR dev STRING
       ip addr {show|save|flush} [ dev STRING ] [ scope ...
       ip addr {showdump|restore}
IFADDR := PREFIX | ADDR peer PREFIX
...
```

- Note that the `ip addr` object provides nine commands:
  - The commands are `add`, `change`, `replace`, `del`, `show`, `save`, `flush`, `showdump`, and `restore`.

– Refer to the `ip-address(8)` man page for more information on using `ip addr` commands.

b. Run the `ip addr show` command.

- This command shows the same information as `ip link` but also shows IP address information.
- The `show` command is the default for the `ip addr` object so is not required.
- Only partial output is shown.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.103/24 brd 192.0.2.255 scope global eth0
    inet6 ...
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

- Note that IP address information is provided for each device.

c. Run the `ip addr add` command to add a second IP address to `eth1`.

- Use `10.1.1.1/24` as the second IPv4 address.

```
# ip addr add 10.1.1.1/24 dev eth1
```

d. Run the `ip addr show` command to show the `eth1` device.

- The `show` command is required when specifying a device as an argument.

```
# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet 10.1.1.1/24 scope global eth1
    inet6
...
```

- Note that the device now has two IPv4 addresses.

e. Run the `ip addr del` command to delete an IP address from `eth1`.

- Delete address `10.1.1.1/24`.

```
# ip addr del 10.1.1.1/24 dev eth1
```

f. Run the `ip addr show` command to show the `eth1` device.

```
# ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
    inet6
...
```

- Note that the `10.1.1.1/24` address no longer exists.

5. Run the `ip neigh` object commands.
   - The "`ip neigh`" command is used to display and manage ARP cache entries.

   a. Run the `ip neigh help` command.
   - Only partial output is shown.

```
# ip neigh help
Usage: ip neigh { add | del | change | replace } { ADDR ...
...
       ip neigh {show|flush} [ to PREFIX ]  [ dev DEV ] ...
```

   - Note that the `ip neigh` object provides six commands:
     – The commands are `add`, `del`, `change`, `replace`, `show`, and `flush`.

   b. Run the `ip neigh show` command.
   - The `show` command is the default for the `ip neigh` object so is not required.

```
# ip neigh
```

   c. If no output is produced from the `ip neigh show` command, use the `ping` command to communicate to **dom0** and the other VM guests.
   - Press **Ctrl + C** to kill the `ping` command.

```
# ping dom0
64 bytes from example.com (192.0.2.1)...
CTRL-C
# ping host01
64 bytes from host01.example.com (192.0.2.101)...
CTRL-C
# ping host02
64 bytes from host02.example.com (192.0.2.102)...
CTRL-C
```

   d. Run the `ip neigh show` command again to list ARP cache entries.

```
# ip neigh
192.0.2.102 dev eth0 lladdr 00:16:3e:00:01:02 REACHABLE
192.0.2.1 dev eth0 lladdr fe:ff:ff:ff:ff:ff REACHABLE
192.0.2.101 dev eth0 lladdr 00:16:3e:00:01:01 REACHABLE
```

e. Run the `ip neigh flush` command to clear all entries in the ARP cache.

```
# ip neigh flush all
```

f. Run the `ip neigh show` command.

```
# ip neigh
```

- No output is produced.

Practices for Lesson 16: File Sharing

Chapter 16

Practices for Lesson 16: Overview

Practices Overview

In these practices, you:

- Configure an NFS server and client and mount an exported file system
- Use automounter to mount the virtual CD drive
- Configure and use an FTP server

Practice 16-1: Configuring an NFS Server and an NFS Client

Overview

In this practice, you:

- Export a file system from **host03** VM and mount it on **host01** VM
- Ensure that the required package is installed and that services are running
- Use various NFS-related commands and files to share file systems using NFS

Assumptions

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

Tasks

1. Export a file system from **host03**.
   a. Use the `fdisk` command to display the partition table on `/dev/xvdb`.

```
[host03]# fdisk -l /dev/xvdb
Disk /dev/xvdb: 5368 MB,  5368709120 bytes, 10485760 sectors
...
    Device Boot    Start        End    Blocks    Id    System
/dev/xvdb1          2048    2099199   1048576    83    Linux
/dev/xvdb2       2099200    4196351   1048576    8e    Linux LVM
```

   - This shows two partitions on `/dev/xvdb`.

   b. Use the `mkfs.ext4` command to make an ext4 file system on `/dev/xvdb1`.

```
[host03]# mkfs.ext4 /dev/xvdb1
...
Writing superblocks and filesystem accounting information: done
```

   c. Use the `fdisk` command to display the partition table on `/dev/xvdd`.

```
[host03]# fdisk -l /dev/xvdd
Disk /dev/xvdb: 5368 MB,  5368709120 bytes, 10485760 sectors
...
    Device Boot    Start        End    Blocks    Id    System
/dev/xvdd1          2048    2099199   1048576    83    Linux
/dev/xvdd2       2099200    4196351   1048576    8e    Linux LVM
```

   - This shows two partitions on `/dev/xvdb`.

   d. Use the `mkfs.ext4` command to make an ext4 file system on `/dev/xvdd1`.

```
[host03]# mkfs.ext4 /dev/xvdd1
...
Writing superblocks and filesystem accounting information: done
```

   e. Use the `vi` editor to add the following entries to `/etc/fstab`.

```
[host03]# vi /etc/fstab
/dev/xvdb1   /Dev    ext4  defaults  0  0
/dev/xvdd1   /Test   ext4  defaults  0  0
```

f. Use the `mount -a` command to mount all the file systems defined in `/etc/fstab`.

```
[host03]# mount -a
```

g. Use the `df` command to display the mounted file system on `/dev/xvdb1`.

```
[host03]# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
...
/dev/xvdb1     976M  2.6M   907M    1%  /Dev
/dev/xvdd1     976M  2.6M   907M    1%  /Test
```

- This shows `/dev/xvdb1` is mounted on `/Dev`.
- This shows `/dev/xvdd1` is mounted on `/Test`.

h. Use the `vi` editor and edit `/etc/exports` to export `/Dev` to all client systems.

```
[host03]# vi /etc/exports
/Dev *
```

2. Start the NFS service on **host03**.

a. Use the `rpm` command to verify that the `rpcbind` package is installed.

```
[host03]# rpm -q rpcbind
rpcbind-...
```

- In this example, the package is already installed.

b. Use the `systemctl` command to verify that the `rpcbind` service is started.

```
[host03]# systemctl status rpcbind
rpcbind.service –RPC bind service
   Loaded: loaded (/usr/lib/systemd/system/rpcbind.service...
   Active: active (running) since ...
...
```

- In this example, the `rpcbind` service is enabled and running.

c. Use the `rpm` command to verify that the `nfs-utils` package is installed.

```
[host03]# rpm -q nfs-utils
nfs-utils-...
```

- In this example, the package is installed.

d. Use the `systemctl` command to verify that the `nfs` service is started.

```
[host03]# systemctl status nfs
nfs-server.service - NFS Server
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service...
   Active: inactive (dead)
```

- In this example, the `nfs-server` service is disabled and not running.

e. Use the `systemctl` command to start the `nfs` service and associated services.

```
[host03]# systemctl start nfs
```

f. Use the `systemctl` command to enable the `nfs-server` service to start at boot time.

- You must use the full name, `nfs-server`, to enable the NFS service.

```
[host03]# systemctl enable nfs
Failed to issue method call: No such file or directory
[host03]# systemctl enable nfs-server
ln -s '/usr/lib/systemd/system/nfs-server.service'
'/etc/systemd/system/nfs.target.wants/nfs-server.service'
```

g. Use the `systemctl` command to verify that the `nfslock` service is started.

```
[host03]# systemctl status nfslock
nfs-lock.service - NFS file locking service.
   Loaded: loaded (/usr/lib/systemd/system/nfs-lock.service...
   Active: active (running) since ...
...
```

- In this example, the `nfslock` service is running.

h. Use the `showmount -e` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Dev *
```

- 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.

3. Configure an NFS server from the command line.

a. Use the `exportfs` command to export `/Test` to all clients and allow read/write permission.

- Include the `-i` option to ignore `/etc/exports` entries.

```
[host03]# exportfs -i -o rw *:/Test
```

b. Use the `showmount` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Dev *
/Test *
```

- Note that both exported file systems are listed.
- You do not need to restart the `nfs` service when using `exportfs`.

c. Use the `cat` command to view the contents of `/var/lib/nfs/etab`.

```
[host03]# cat /var/lib/nfs/etab
/Dev  *(ro,sync,wdelay,hide,nocrossmnt,secure,root_squash,...
/Test *(rw,sync,wdelay,hide,nocrossmnt,secure,root_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.

d. Use the `vi` editor to edit `/etc/exports`.

- Change the entry exporting `/Dev` to the following:

```
[host03]# vi /etc/exports
/Dev *(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.

e. 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`.

f. Use the `cat` command to view the contents of `/var/lib/nfs/etab`.

```
[host03]# cat /var/lib/nfs/etab
/Dev  *(rw,sync,wdelay,hide,nocrossmnt,secure,no_root_squash...
```

- Note the new options on the `/Dev` NFS share.
- Also note that `/Test` is no longer listed.

g. Use the `showmount -e` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Dev *
```

- This confirms that the `/Test` file system is no longer exported.

4. Mount the exported `/Dev` file system on **host01**.

a. Use the `rpm` command to verify that the `nfs-utils` package is installed.

```
[host01]# rpm -q nfs-utils
nfs-utils-...
```

- In this example, the package is installed.

b. Use the `mkdir` command to create a mountpoint named `/remote_dev` on **host01**.

```
[host01]# mkdir /remote_dev
```

c. Use the `mount` command to mount the exported file system from **host03**, `/Dev`, 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.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
mount.nfs: mount system call failed
```

- In this example, the `mount` command fails.
- It takes several seconds for the `mount` command to fail. Rather than wait for the command to fail, you can press Ctrl + C to abort the `mount` command.

d. 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 **host01** to mount the exported file system from **host03**.

```
[host03]# systemctl stop firewalld
```

- `firewalld` and other system security–related issues are covered in the lesson titled “Security Administration.”
- You can allow NFS connectivity without disabling firewalls, which is covered in lesson titled “Security Administration.”

e. Re-issue the `mount` command from **host01**, if it is necessary to mount the NFS share.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
```

- In this example, the `mount` command was successful.

f. Run the mount command to view the mount information for the NFS share.

```
[host01]# mount | grep nfs
...
host03:/Dev on /remote_dev type nfs4 (rw,nosuid,relatime,ver...
clientaddr=192.0.2.101,local_lock=none,addr=192.0.2.103)
```

- Note the mount options (`rw,nosuid`), the NFS version (`4`), and the server and client IP addresses.

5. Verify that the NFS file system is mounted with `rw` (read/write) permissions.

a. On **host01**, use the `df` command to display the mounted file systems.

```
[host01]# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvda2    5.7G  1.1G   4.3G   21%  /
...
host03:/Dev   976M  2.5M   907M    1%  /remote_dev
```

- Note that the `host03:/Dev` file system is mounted on the local file system `/remote_dev`.

b. Use the `ls` command to list the contents of `/remote_dev` on **host01**.

```
[host01]# ls /remote_dev
lost+found
```

c. Use the `ls` command to list the contents of `/Dev` on **host03**.

```
[host03]# ls /Dev
lost+found
```

- Note that the contents of `/Dev` on **host03** are the same as `/remote_dev` on **host01** because they are the same directories.

d. From **host03**, use the `vi` command to create a file on `/Dev`. Enter some content in the file.

```
[host03]# vi /Dev/test
<enter some content>
```

e. From **host01**, use the `vi` command to edit the file created from **host03**. Make some changes to the content.

```
[host01]# vi /remote_dev/test
<change the content>
```

f. From **host03**, use the `cat` command to view the contents of the file in `/Dev`.

```
[host03]# cat /Dev/test
<content has changed>
```

- This again confirms `/Dev` on **host03** is the same as `/remote_dev` on **host01** and that the file system has read/write permissions.

6. Unmount the NFS file system on **host01**.

On **host01**, use the `umount` command to unmount `/remote_dev`.

- Use the `cd` command to ensure you are not in `/remote_dev` before unmounting.

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

Practice 16-2: Using Automounter

Overview

In this practice, you:

- Use automounter to mount the virtual CD drive
- Use the -hosts map to automount all exports from **host03**

Assumptions

- This practice is performed on **host01** VM, but you are asked to run a single command from **dom0**.
- Open a terminal window on each system.
- Log in as the root user on each system.

Tasks

1. From **dom0**, use the grep command to search for "cdrom" in the vm.cfg file for **host01**.

```
[dom0]# grep cdrom /OVS/running_pool/host01/vm.cfg
'file:/OVS/seed_pool/OracleLinux-R7-GA-Server-x86_64-
dvd.iso,hdc:cdrom,r']
```

- Note that the dvd.iso image of Oracle Linux 7 is on the virtual cdrom drive.

All remaining commands are run from the **host01** VM.

2. Start the autofs service.
   a. Use the rpm command to verify that the autofs package is installed.

```
# rpm -q autofs
autofs-5.0.7-40.0.1.el7.x86_64
```

   - In this example, the package is already installed.

   b. Use the systemctl command to verify that the autofs service is running.

```
# systemctl status autofs
autofs.service - Automounts filesystems on demand
   Loaded: loaded (/usr/lib/systemd/system/autofs.service...
   Active: inactive (dead)
```

   - In this example, the autofs service is disabled and not running.

   c. Use the systemctl command to start the autofs service.

```
# systemctl start autofs
```

d. Display the automount user-space daemon.

- Starting the autofs service also starts the automount user-space daemon.
- To view the automount user-space daemon, run the ps -ef command, pipe the output to grep, and search for the "auto" string.

```
# ps -ef | grep auto
root  <pid> ... /usr/sbin/automount -pid-file /run/autofs.pid
...
```

3. Automount /dev/cdrom on **host01** by using an indirect map file.

a. Use the grep command to search for "misc" in the /etc/auto.master master map file.

- The entry shown is an indirect map entry.

```
# grep misc /etc/auto.master
/misc     /etc/auto.misc
...
```

- Note that the /misc mount point is associated with the /etc/auto.misc map file.

b. Use the grep command to search for "cd" in the /etc/auto.misc file.

```
# grep cd /etc/auto.misc
cd        -fstype=iso9600,ro,nosuid,nodev         :/dev/cdrom
```

- Note that the cd mount point is associated with the /dev/cdrom device.

c. Use the df command to display the mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvda2    5.7G  1.1G   4.3G   20%  /
...
```

- Note that the cdrom is not mounted.

d. Use the ls /misc/cd command, combining the /misc mountpoint from /etc/auto.master with the cd mountpoint from /etc/auto.misc.

```
# ls /misc/cd
addons  EULA    GPL     isolinux  Packages  RPM-GPG-KEY ...
EFI     Extras  images  LiveOS    repodata  RPM-GPG-KEY-oracle
```

- Note that this is the contents of the Oracle Linux 7 dvd-iso image.

e. Use the df command to display the mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
/dev/xvda2    5.7G  1.1G   4.3G   20%  /
...
/dev/sr0      3.9G  3.9G      0  100%  /misc/cd
```

- Note that the cdrom was "automounted" simply by accessing it with the ls command.

4. Automount the exported file systems from **host03** by using a host map file.
   a. From **host01**, use the `grep` command to search for "`host`" in the `/etc/auto.master` master map file.

```
# grep host /etc/auto.master
...
/net     -hosts
```

   - Note that the `/net` mount point is associated with the `-hosts` map.

   b. From **host01**, use the `cd` command to change to the `/net/host03` directory to automount all exports from **host03**.

   - Use the `ls` command to list the contents of the directory.

```
# cd /net/host03
# ls
Dev
```

   - Note that the `/net/host03` directory contains the exported file system from **host03**, `/Dev`.

   c. From **host01**, use the `ls` command to list the contents of `/net/host03/Dev`.

```
# ls /net/host03/Dev
lost+found     test
```

Practice 16-3: 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
...
Package vsftpd-3.0.2-9.el7.x86_64 already installed and ...
Nothing to do.
```

   - In this example, the package is already installed.

   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
ln-s '/usr/lib/systemd/system/vsftpd.service'
'/etc/systemd/system/multi-user.target.wants/vsftpd.service'
```

2. Use the `yum` command to install the `ftp` (client) package on **host03**.
   - Answer '**y**' when prompted.

```
# yum install ftp
...
Transaction Summary
============================================================
Install       1 Package(s)
Total download size: 60 k
Installed size: 96 k
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 (127,0,0,1,54,166).
150 Here comes the directory listing.
drwxr-xr-x  2 0    0   4096  Apr 29  2012 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 the `/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 16-4: Downloading a File from an FTP Server

Overview

In this practice, you install the `ftp` package on **host01** VM and download a file from the FTP server on **host03**.

Assumptions

You are the `root` user on **host01** VM.

Tasks

1. Install the `ftp` package on **host01**.
   a. From **host01**, use the `cd /misc/cd` command to automount the virtual `cdrom`, which contains Oracle Linux `dvd.iso` image.

```
# cd /misc/cd
```

   b. Use the `cd` command to change to the `Packages` directory.

```
# cd Packages
```

   c. Use the `ls` command to display the `ftp` package name.

```
# ls ftp*
ftp-0.17-66.el7.x86_64.rpm
```

   d. Use the `rpm` command to install the `ftp` package.

```
# rpm –Uvh ftp-0.17-66.el7.x86_64.rpm
...
```

2. Download a file using `ftp` to **host01** from the FTP server, **host03**.
   a. From **host01**, use the `cd` command to change back to your home directory.

```
# cd
# pwd
/root
```

   b. Use the `ftp` utility to connect to the FTP server, **host03**, as `anonymous` user.
   - Use any password.

```
# ftp host03
Connected to localhost (192.0.2.103).
220 (vsFTPd 2.2.2)
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 (192,0,2,103,26,76).
150 Here comes the directory list.
-rw-r--r--    1 0       0       1636 <date_time> test_file
226 Directory send OK.
```

- Note that the `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 **host01**.

```
ftp> !ls
anaconda-ks.cfg
```

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 (192,0,2,103,40,249).
150 Opening BINARY mode data connection for test_file ...
226 Transfer complete.
1636 bytes received in ...
```

i. Use the `!ls` command to list the contents of the local directory on **host01**.

```
ftp> !ls
anaconda-ks.cfg   test_file
```

- Note that the `test_file` exists in the local directory on **host01**.

j. Use the `quit` command to exit `ftp`.

```
ftp> quit
221 Goodbye.
```

Practices for Lesson 17: OpenSSH

Chapter 17

Practices for Lesson 17: Overview

Practices Overview

In these practices, you do the following:

- You verify that the OpenSSH packages are installed and that the `sshd` service is running.
- You use the `ssh` and `scp` utilities.
- You use the `ssh-keygen` utility to generate keys enabling connectivity without supplying a password.

Practice 17-1: Connecting to a Remote Server by Using ssh

Overview

In this practice, you verify that the OpenSSH packages are installed, verify that the `sshd` service is started on the server, and use the `ssh` utility to establish a connection and execute a command on a remote system.

Assumptions

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

Tasks

1. Verify that the `sshd` service is running on **host03**.
   a. Use the `rpm` command to verify that the openssh packages are installed.

```
[host03]# rpm -qa | grep openssh
openssh-clients-...
openssh-server-...
openssh-...
```

   - In this example, the packages are already installed.

   b. Use the `systemctl` command to verify that the `sshd` service is started.

```
[host03]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service...
   Active: active (running) since ...
...
```

   - In this example, the service is enabled and running.

2. Log in to remote **host03** from **host01**.
   a. On **host01**, use the `rpm` command to verify that the openssh packages are installed.

```
[host01]# rpm -qa | grep openssh
openssh-clients-...
openssh-server-...
openssh-...
```

   - In this example, the packages are already installed.

   b. Use the `exit` command to log off as `root` on **host01**. Log back on as user `oracle`.
   - The password is `oracle`.

```
[host01]# exit
host01 login: oracle
Password: oracle
```

c. Use the ls command to display a long listing of all files in the home directory of user oracle.

```
[oracle@host01 ~]$ ls -la
...
```

- Notice that there is no ~/.ssh directory.

d. Perform a remote login to **host03** by using the ssh command.

- Answer "**yes**" when asked, "Are you sure."
- The oracle user password on **host03** is oracle.

```
[oracle@host01 ~]$ ssh host03
The authenticity of host 'host03 (192.0.2.103)' can't be
established. 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.
oracle@host03's password: oracle
[oracle@host03 ~]$
```

e. Use the hostname command to display the host name to confirm that you successfully logged on to **host03**.

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

f. Use the logout command to close the ssh connection to **host03**. Use the hostname command to confirm that you are back to **host01**.

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

3. View the .ssh directory in the oracle user's home directory.

a. Use the ls command to display a long listing of all files in the home directory of user oracle.

```
[oracle@host01 ~]$ ls -la
...
drwx------ 2 oracle oracle 4096 <date_time> .ssh
```

- Notice that there is now a ~/.ssh directory.

b. Use the cd command to change to the ~/.ssh directory, and then use ls to view the contents of the directory.

```
[oracle@host01 ~]$ cd .ssh
[oracle@host01 .ssh]$ ls
known_hosts
```

- Notice that the known_hosts file was created.

c. Use the `cat` command to view the contents of the `known_hosts` file.

```
[oracle@host01 .ssh]$ cat known_hosts
host03,192.0.2.103 ecdsa-sha2-nistp256 ...
```

- Notice that **host03** is now a “known host”.

4. Log in to remote **host03** from **host01**.

a. Perform a remote login to **host03** using the `ssh` command.

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

- Notice that you are not asked to confirm this time, because of the existence of the `known_hosts` file.

b. Use the `logout` command to close the `ssh` connection to **host03**.

- Use the `hostname` command to confirm that you are back to **host01**.
- Use the `cd` command to change back to the user's home directory.

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

c. Log on to **host03** as user `root` and run the `ls` command with a single `ssh` command.

- The `root` user password on **host03** is `oracle`.

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

- Note that the `ls` command ran on the remote system displaying the contents of the remote directory, and then the remote connection closed.

d. Use the `hostname` command to confirm that you are back to **host01**.

```
[oracle@host01 ~]$ hostname
host01.example.com
```

Practice 17-2: Configuring OpenSSH to Connect Without a Password

Overview

In this practice, you use the `ssh-keygen` command to generate an RSA key pair and configure OpenSSH to connect to a remote system without supplying a password. You also use the `scp` command in this practice.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- Open a terminal window on each system.
- Log in as the `root` user on **host03**.
- Log in as the `oracle` 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. Use the `ssh-keygen` command to create the public and private parts of an RSA key.
   a. From **host03**, use the `su` command to become the `oracle` user.

```
[root@host03 ~]# su – oracle
[oracle@host03 ~]$ whoami
oracle
[oracle@host03 ~]$ pwd
/home/oracle
```

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

```
[oracle@host03 ~]$ ls ~/.ssh
ls: cannot access /home/oracle/.ssh: No such file or directory
```

   - Notice that the directory does not exist.

   c. Use the `ssh-keygen –t rsa` command to create the RSA key.
   - Accept all the defaults.

```
[oracle@host03 ~]$ ssh-keygen –t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/oracle/.ssh/id_rsa):
ENTER
Created directory '/home/oracle/.ssh'.
Enter passphrase (empty for no passphrase): ENTER
Enter same passphrase again: ENTER
Your identification has been saved in /home/oracle/.ssh/id_rsa.
Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
...
The key's randomart image is:
...
```

d. Use the `ls` command to view the contents of the `~/.ssh` directory.

```
[oracle@host03 ~]$ ls ~/.ssh
id_rsa     id_rsa.pub
```

- Note that the `ssh-keygen` command generated two keys.

2. Use the `scp` command to copy `~/.ssh/id_rsa.pub` on the local system (**host03**) to `~/.ssh/authorized_keys` on the remote system (**host01**).
   - Answer "**yes**" to continue.
   - Password is `oracle`.

```
[oracle@host03 ~]$ scp .ssh/id_rsa.pub
host01:~/.ssh/authorized_keys
The authenticity of host 'host01 (192.0.2.101)' can't be
established. 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.
oracle@host01's password: oracle
id_rsa.pub                       100%  407   0.4KB/s   00:00
[oracle@host03 ~]$
```

   - Because you are connecting to this OpenSSH server for the first time, you are asked to confirm the connection.
   - Note that a password is required to make the connection.
   - Note that the file is copied but you are still connected to the local system (**host03**).

3. Log in to remote **host01** from **host03**.

a. Perform a remote login to **host01** by using the `ssh` command.

```
[oracle@host03 ~]$ ssh host01
Last login:...
```

- Note that you no longer need to enter a password.

b. Use the `hostname` command to confirm that you successfully logged on to **host01**.

```
[oracle@host01 ~]$ hostname
host01.example.com
```

c. Use the `ls` command to view the contents of the `~/.ssh` directory.

```
[oracle@host01 ~]$ ls ~/.ssh
authorized_keys   known_hosts
```

- Note the existence of the `authorized_keys` file, which allowed you to connect without supplying a password.

d. Use the `logout` command to close the connection to **host01**.

- Use the `hostname` command to confirm that you are back to **host03**.

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

e. Use the `exit` command to log out as `oracle` user and return to the `root` logon.

- Use the `whoami` command to confirm that you are logged on as `root`.

```
[oracle@host03 ~]$ exit
logout
[root@host03 ~]$ whoami
root
```

Practices for Lesson 18: Security Administration

Chapter 18

Practices for Lesson 18: Overview

Practices Overview

In these practices, you:

- Configure a `chroot` jail
- Configure a `chroot` jail for `ftp` users
- Explore and configure `firewalld`
- Configure `iptables`
- Configure a TCP wrapper

Practice 18-1: Configuring a `chroot` Jail

Overview

In this practice, you configure a `chroot` jail and copy all files required to run the `/bin/bash` shell in the `chroot` jail.

Assumptions

You are the `root` user on **host03** VM.

Tasks

1. Become the `oracle` user.
   a. Use the `su - oracle` command to become the `oracle` user.

```
# su – oracle
```

   b. Use the `whoami` command to confirm that you are logged in as the `oracle` user.

```
$ whoami
oracle
```

   c. Use the `cd` command to change to the `oracle` user's home directory.
      - Use the `pwd` command to ensure you are in the `/home/oracle` directory.

```
$ cd
$ pwd
/home/oracle
```

2. Create a `chroot` jail.
   a. As the `oracle` user, use the `mkdir` command to make a `jail` directory in the current directory.

```
$ mkdir jail
```

   b. Use the `exit` command to log off as the `oracle` user and return to being the `root` user.
      - Use the `whoami` command to confirm that you are the `root` user.

```
$ exit
logout
# whoami
root
```

   c. Use the `echo` command to display the value of the `SHELL` variable.

```
# echo $SHELL
/bin/bash
```

      - In this example, `SHELL=/bin/bash`.

   d. Use the `chroot` command to create a `chroot` jail in the `/home/oracle/jail` directory.

```
# chroot /home/oracle/jail
chroot: failed to run command '/bin/bash': No such file or
directory
```

- Note that if you do not specify a command as an argument, `chroot` attempts to run the value of the `SHELL` variable, `/bin/bash`, in the `chroot` jail directory, `/home/oracle/jail`.
- The command failed because `/bin/bash` was not found in `/home/oracle/jail`.

3. As the `oracle` user, copy `/bin/bash` and other necessary files into the `~/jail` directory.
   a. Use the `su - oracle` command to become the `oracle` user.

```
# su - oracle
```

   b. Use the `cd` command to change to the ~/`jail` directory.
      - Use the `mkdir` command to make a `bin` directory.

```
$ cd ~/jail
$ mkdir bin
```

   c. Use the `cp` command to copy `/bin/bash` into `~/jail/bin`.

```
$ cp /bin/bash ~/jail/bin
```

   d. Use the `ldd` command to determine which shared libraries are required by `/bin/bash`.

```
$ ldd /bin/bash
linux-vdso.so.1 => (0x0000...)
libtinfo.so.5 => /lib64/libtinfo.so.5 (0x0000...)
libdl.so.2 => /lib64/libdl.so.2 (0x0000...)
libc.so.6 => /lib64/libc.so.6 (0x0000...)
/lib64/ld-linux-x86-64.so.2 (0x0000...)
```

      - In this example, there are four shared library files in `/lib64` used by `/bin/bash`.

   e. Within the `jail` directory in your home directory, use the `mkdir` command to make a `lib64` directory.

```
$ cd ~/jail
$ mkdir lib64
```

   f. Use the `cp` command to copy the four shared library files required for `/bin/bash` from `/lib64` to `~/jail/lib64`.

```
$ cp /lib64/libtinfo.so.5 ~/jail/lib64
$ cp /lib64/libdl.so.2 ~/jail/lib64
$ cp /lib64/libc.so.6 ~/jail/lib64
$ cp /lib64/ld-linux-x86-64.so.2 ~/jail/lib64
```

4. Create a `chroot` jail.
   a. Use the `exit` command to log off as the `oracle` user and return to being the `root` user.

```
$ exit
logout
```

b. As the `root` user, use the `chroot` command to create a `chroot` jail in the `/home/oracle/jail` directory.

```
# chroot /home/oracle/jail
```

- Note that the `chroot` command was successful—no errors occurred and the `/bin/bash` program executed.

c. Use the `pwd` command to display the current directory.

```
# pwd
/
```

- Note that the output indicates that the current directory is the root-level directory even though the actual directory is `/home/oracle/jail`.

d. Use the `exit` command to exit the `chroot` jail.

```
# exit
```

Practice 18-2: Configuring a `chroot` Jail for `ftp` Users

Overview

In this practice, you:

- Confirm that `anonymous` ftp users are placed in a `chroot` jail on a `vsftpd` server by default
- Configure local users to be placed in a `chroot` jail

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- Open a terminal window on each system.
- Log in as the `root` user on each system.
- The prompts in the solution section include either **host01** or **host03** to indicate which system to enter the command from.
- You completed Practice 16-3: Configuring an FTP Server.

Tasks

1. Confirm that `anonymous` ftp users are placed in a `chroot` jail by default.
   a. From **host03**, use the `ls` command to list the contents of the `/var/ftp` directory.

```
[host03]# ls –l /var/ftp
drwxr-xr-x...  pub
```

   - Note that `/var/ftp` contains a single directory, `pub`.

   b. From **host03**, use the `ls` command to list the contents of `/var/ftp/pub`.

```
[host03]# ls /var/ftp/pub
test_file
```

   - Note that the `/var/ftp/pub` directory contains a single file, `test_file`.

   c. From **host01**, `ftp` to **host03** as `anonymous` user.
   - Press Enter when prompted for a password.

```
[host01]# ftp host03
Connected to host03: (192.0.2.103).
220 (vsFTPd 3.0.2)
Name...: anonymous
331 Please specify the password.
Password: ENTER
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

   d. If you get an “`ftp: connect: No route to host`” error when running the “`ftp host03`” command, perform the following steps:
   - Type `quit` at the `ftp>` prompt on **host01** to exit `ftp`.

- Stop the `firewalld` service on **host03** by running the `systemctl` command shown as follows.

```
[host03]# systemctl stop firewalld
```

- Repeat step 1c to establish an `ftp` connection from **host01** to **host03**.

e. From **host01**, use the `ls` command to list the contents of the current directory.

```
ftp> ls
...
drwxr-xr-x ... pub
226 Directory send OK.
ftp>
```

- Note that the current directory contains a single directory, `pub`.

f. Use the `ls` command to list the contents of the `pub` directory.

```
ftp> ls pub
...
-rw-rw-rw- ... test_file
226 Directory send OK.
ftp>
```

- Note that the `pub` directory contains a single file, `test_file`.
- This confirms that the current location of the `anonymous` FTP user is `/var/ftp` on **host03**.

g. Use the `pwd` command to display the current directory.

```
ftp> pwd
257 "/"
ftp>
```

- Note that the output indicates that the current directory is the `root`-level directory even though the actual directory is `/var/ftp`.
- This confirms that `anonymous` users are placed in a `chroot` jail by default.

h. Use the `quit` command to exit `ftp`.

```
ftp> quit
221 Goodbye.
```

2. Confirm that local ftp users are placed in their home directory by default.

a. From **host01**, `ftp` to **host03** as the `oracle` user.

- Password is `oracle`.

```
[host01]# ftp host03
Connected to host03: (192.0.2.103).
220 (vsFTPd 3.0.2)
Name...: oracle
331 Please specify the password.
Password: oracle
230 Login successful.
...
ftp>
```

b. Use the pwd command to display the current directory.

- Use the ls command to display the contents of the current directory.

```
ftp> pwd
257 "/home/oracle"
ftp> ls
...
drwxr-xr-x ...  Desktop
drwxr-xr-x ...  Documents
drwxr-xr-x ...  Downloads
...
ftp>
```

- The output indicates that the oracle user was placed in its home directory.
- This is the case for all local users that access a vsftpd server; they are placed in their home directory by default, and not in a chroot jail.

c. Use the quit command to exit ftp.

```
ftp> quit
221 Goodbye.
```

3. On **host03**, enable options in the /etc/vsftpd/vsftpd.conf file to put local users in a chroot jail.

a. Use the vi editor to make the following changes:

- Remove the # sign to uncomment the chroot_local_user directive and set to YES.
- Add the allow_writeable_chroot directive and set to YES.
- Ensure the following chroot directives are commented out (preceded with a # sign, as shown):
  - #chroot_list_enable=YES
  - #chroot_list_file=/etc/vsftpd/chroot_list

```
[host03]# vi /etc/vsftpd/vsftpd.conf
chroot_local_user=YES
allow_writeable_chroot=YES
#chroot_list_enable=YES
#chroot_list_file=/etc/vsftpd/chroot_list
```

- After making changes to the vsftpd.conf file, you need to restart the vsftpd service.

b. Use the systemctl command to restart the vsftpd service.

```
[host03]# systemctl restart vsftpd
```

c. On **host03**, use the setenforce 0 command to change SELinux to "Permissive" mode. Use the getenforce command before and after to confirm the change.

- SELinux is covered in another course.

- For purposes of this practice, set SELinux to "Permissive" mode.

```
[host03]# getenforce
Enforcing
[host03]# setenforce 0
[host03]# getenforce
Permissive
```

4. Verify `chroot` settings are working.
   a. From **host01**, `ftp` to **host03** as the `oracle` user.
      - Password is `oracle`.

```
[host01]# ftp host03
Connected to host03: (192.0.2.103).
220 (vsFTPd 3.0.2)
Name...: oracle
331 Please specify the password.
Password: oracle
230 Login successful.
...
ftp>
```

   b. Use the `pwd` command to display the current directory.
      - Use the `ls` command to list the contents of the current directory.

```
ftp> pwd
257 "/"
ftp> ls
...
drwxr-xr-x ...  Desktop
drwxr-xr-x ...  Documents
drwxr-xr-x ...  Downloads
...
ftp>
```

      - The output indicates that the current directory is the root-level directory even though the actual directory is `/home/oracle` on the `vsftpd` server, **host03**.

   c. Use the `quit` command to exit `ftp`.

```
ftp> quit
221 Goodbye.
```

5. Restore `vsftpd.conf` settings on **host03** to their original state.
   a. On **host03**, disable options in the `/etc/vsftpd/vsftpd.conf` file to put local users in a `chroot` jail. Use the `vi` editor to make the following changes:
      - Insert # sign to comment out the `chroot_local_user` directive.
      - Insert # sign to comment out the `allow_writeable_chroot` directive.

```
[host03]# vi /etc/vsftpd/vsftpd.conf
#chroot_local_user=YES
```

```
#allow_writeable_chroot=YES
```

- After making changes to the `vsftpd.conf` file, you need to restart the `vsftpd` service.

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

```
[host03]# systemctl restart vsftpd
```

c. On **host03**, use the `setenforce 1` command to change SELinux to “Enforcing” mode. Use the `getenforce` command to confirm the change.

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

d. From **host01**, `ftp` to **host03** as the `oracle` user.

- Password is `oracle`.

```
[host01]# ftp host03
Connected to host03: (192.0.2.103).
220 (vsFTPd 3.0.2)
Name...: oracle
331 Please specify the password.
Password: oracle
230 Login successful.
...
ftp>
```

e. Use the `pwd` command to display the current directory.

```
ftp> pwd
257 "/home/oracle"
ftp>
```

- Note that the `oracle` user is now placed in their home directory and not in a `chroot` jail.

f. Use the `quit` command to exit `ftp`.

```
ftp> quit
221 Goodbye.
```

Practice 18-3: Exploring `firewalld`

Overview

In this practice, you:

- Start the `firewalld` service
- Start the Firewall Configuration GUI
- Explore `firewalld` zones
- Change the default zone
- Explore **Runtime** and **Permanent** configuration modes
- Explore `firewalld` services

Assumptions

- You are the `root` user on **host03** VM.
- This practice assumes the `firewalld` service is not running.

Tasks

1. Start the `firewalld` service on **host03**.
   a. Use the `systemctl` command to check if the `firewalld` service is running.

```
# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service...
   Active: inactive (dead) since ...
...
```

   - In this example, `firewalld` is enabled but is not running.

   b. Use the `firewall-cmd` command-line utility to check if the `firewalld` service is running.
   - This command provides another method to check the status of `firewalld`.

```
# firewall-cmd --state
not running
```

   - This confirms that the `firewalld` service is not running.

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

```
# systemctl start firewalld
```

   d. Use the `systemctl` command to check if the `firewalld` service is running.

```
# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service...
   Active: active (running) since ...
...
```

   - The `firewalld` service is now running.

   e. Use the `firewall-cmd` command-line utility to check if the `firewalld` service is running.

```
# firewall-cmd --state
running
```

- This confirms that the `firewalld` service is running.

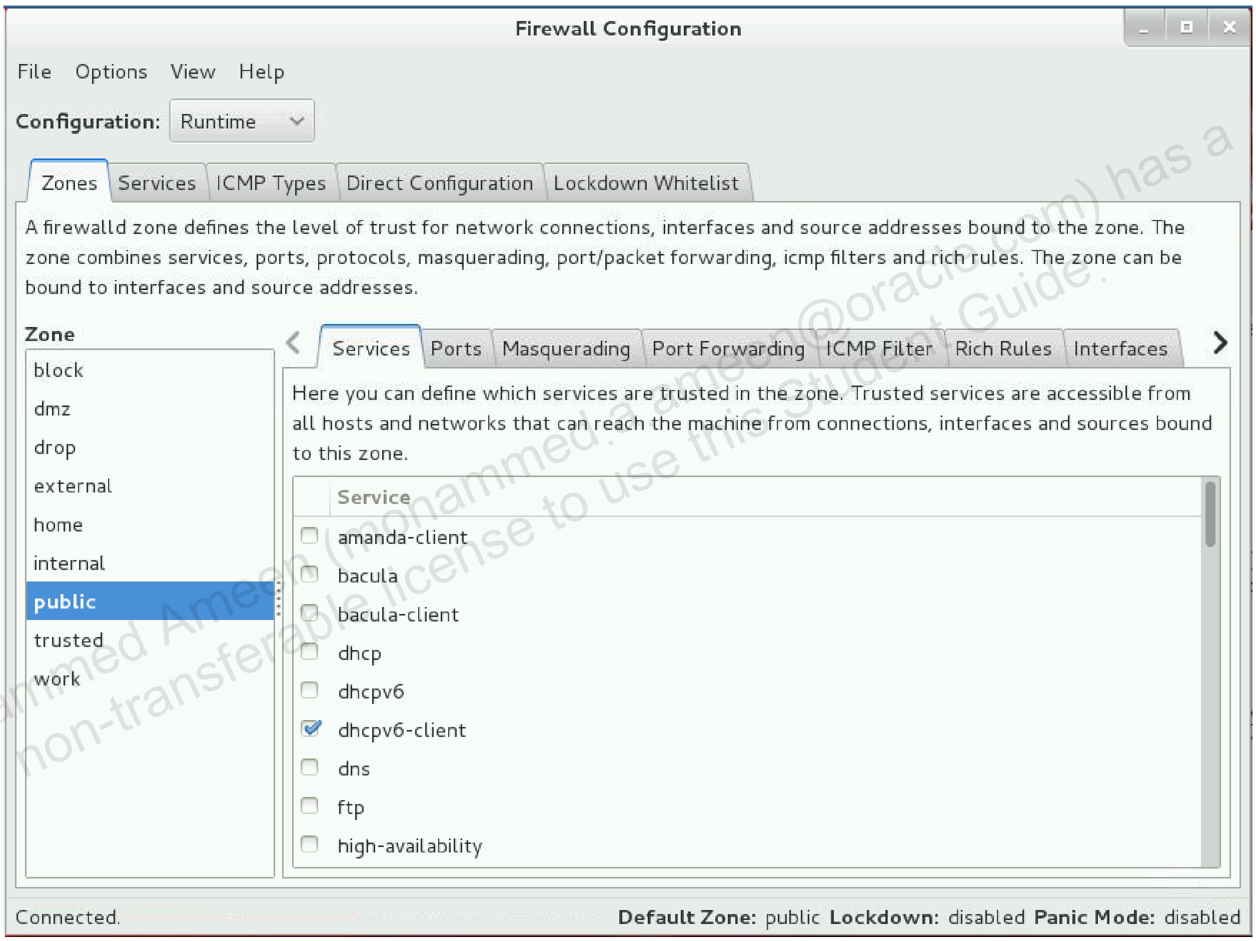
2. Start the Firewall Configuration GUI.

   a. Use the `firewall-config` command to start the GUI.

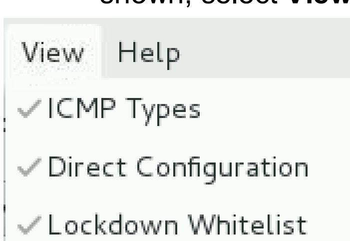
   - Run the command in the background so that you can continue to enter commands from the command line.

```
# firewall-config&
```

   - The GUI appears as shown:

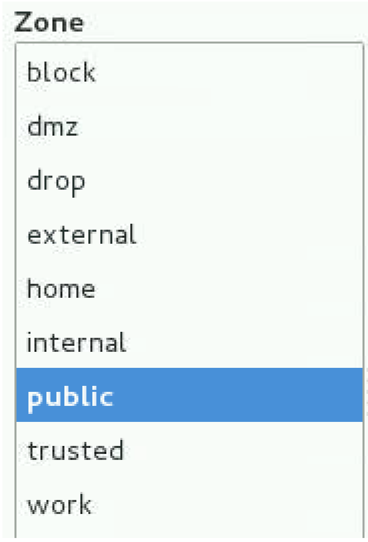


   - Notice the word "Connected" at the lower-left of the window. This indicates that the `firewalld` service is running.
   - If the **ICMP Types**, **Direct Configuration**, and **Lockdown Whitelist** tabs are not shown, select **View** from the menu bar and select each option.



3. Explore `firewalld` zones.

- Note that there are nine predefined zones displayed on the GUI as shown:



- Each of these zones is defined by individual files in the `/usr/lib/firewalld/zones` directory.

a. From a command prompt, use the `ls` command to display the contents of the `/usr/lib/firewalld/zones` directory.

```
# ls /usr/lib/firewalld/zones
block.xml   drop.xml       home.xml       public.xml   work.xml
dmz.xml     external.xml   internal.xml   trusted.xml
```

- Note that there is an XML configuration file for each of the nine predefined zones.

b. Use the `firewall-cmd` command to display the available zones.

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

- Note that this command shows the same nine predefined zones.

c. From the GUI, note that for each zone you can select the lower row of tabs and configure the associated parameters.

- Select the **Services** tab.
  – Note that you can select which services are trusted in a specific zone.
- Select the **Ports** tab.
  – Click the **Add** button.
  – Note that you can add additional ports or port ranges that need to be accessible from hosts or networks.
  – Click the **Cancel** button.
- Select the **Masquerading** tab.
  – Note that you can enable IPv4 masquerading, which causes hosts on your local network to appear as a single IP address on the Internet.
- Select the **Port Forwarding** tab.

- Click the **Add** button.
- Note that you can forward inbound traffic to different ports or different systems.
- Click the **Cancel** button.

- Select the **ICMP Filter** tab.
  - Note that you can block selected types of ICMP messages.
- Select the **Rich Rules** tab.
  - Click the **Add** button.
  - Note that you can extend existing `firewalld` rules to include additional source and destination addresses and logging and auditing actions.
  - Click the **Cancel** button.
- Select the **Interfaces** tab.
  - Note that the `eth0` and `eth1` interfaces are bound to the `public` zone.
  - From a command prompt, run the following `firewall-cmd` command to display the active zones.

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

  - Note that the output of this `firewall-cmd` command confirms that the `eth0` and `eth1` interfaces are bound to the `public` zone.
  - From the GUI, select either the `eth0` or `eth1` interface.
  - Click the **Edit** button.
  - Note that you can assign an interface to a different zone.
  - Click the **Close** button.
  - Note that you can also assign an interface to a different zone by selecting **Options->Change Zones of Connections** from the menu bar.
- Select the **>** character, which selects the **Sources** tab.
  - Click the **Add** button.
  - Note that you can bind source addresses to a specific zone.
  - Click the **Cancel** button.
- Select the **<** character until the **Services** tab is selected.

d. From a command prompt, use the `cat` command to view the main `firewalld` configuration file, `/etc/firewalld/firewalld.conf`.

```
# cat /etc/firewalld/firewalld.conf
# firewalld config file
# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=public
...
```

- Note that the default zone, `public`, is defined in this configuration file.

- Note that the **Default Zone: public** is displayed on the bottom of the GUI as shown:

e. Use the `cat` command to view the contents of the `public.xml` zone file:

```
# cat /usr/lib/firewalld/zones/public.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
    <short>Public</short>
    <description>For use in public areas. You do not trust the
other computers on networks to not harm your computer. Only
selected incoming connections are accepted.</description>
    <service name="ssh"/>
    <service name="dhcpv6-client"/>
</zone>
```

- Note that the description states, "Only selected incoming connections are accepted."
- In this example, the selected incoming connections (trusted services) are `ssh` and `dhcpv6-client`.

f. From the Firewall Configuration GUI, scroll down the **Service** window and note that the following two services are trusted for the `public` zone by default:

- `dhcpv6-client`
- `ssh`
- These services correspond with the service entries in the `public.xml` file.

4. Change the default zone.

a. From a command prompt, use the `firewall-cmd` command to display the current default zone.

```
# firewall-cmd --get-default-zone
public
```

- Note that the `public` zone is the default zone.

b. Use the `firewall-cmd` command to change the default zone to the `work` zone.

```
# firewall-cmd --set-default-zone=work
success
```

c. Use the `firewall-cmd` command to display the current default zone.

```
# firewall-cmd --get-default-zone
work
```

- Note that the `work` zone is the default zone.

d. Use the `grep –i` command to search for the string "`defaultzone`" in the `firewalld` configuration file.

```
# grep –i defaultzone /etc/firewalld/firewalld.conf
DefaultZone=work
```

- Note that the command in step 4b updated the `DefaultZone` setting in the configuration file.
- Note that the **Default Zone: work** is displayed on the bottom of the GUI as shown:

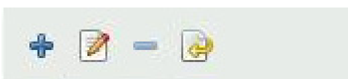
e. Use the `firewall-cmd` command to change the default zone back to the `public` zone.

```
# firewall-cmd --set-default-zone=public
success
```

5. Explore **Runtime** and **Permanent** configuration modes.

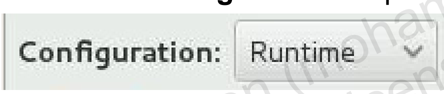
a. From the GUI, click the **Configuration** drop-down menu and select **Permanent**.

- When **Permanent** is selected, changes are applied when the `firewalld` service restarts.
- Note that you can restart `firewalld` from the GUI by selecting **Options->Reload Firewall** from the menu bar.
- Note that in **Permanent** configuration mode, the following menu appears under the list of zones:



- These options allow you to Add, Edit, and Remove a Zone, a Service, or an ICMP Type.

b. Click the **Configuration** drop-down menu and select **Runtime**.

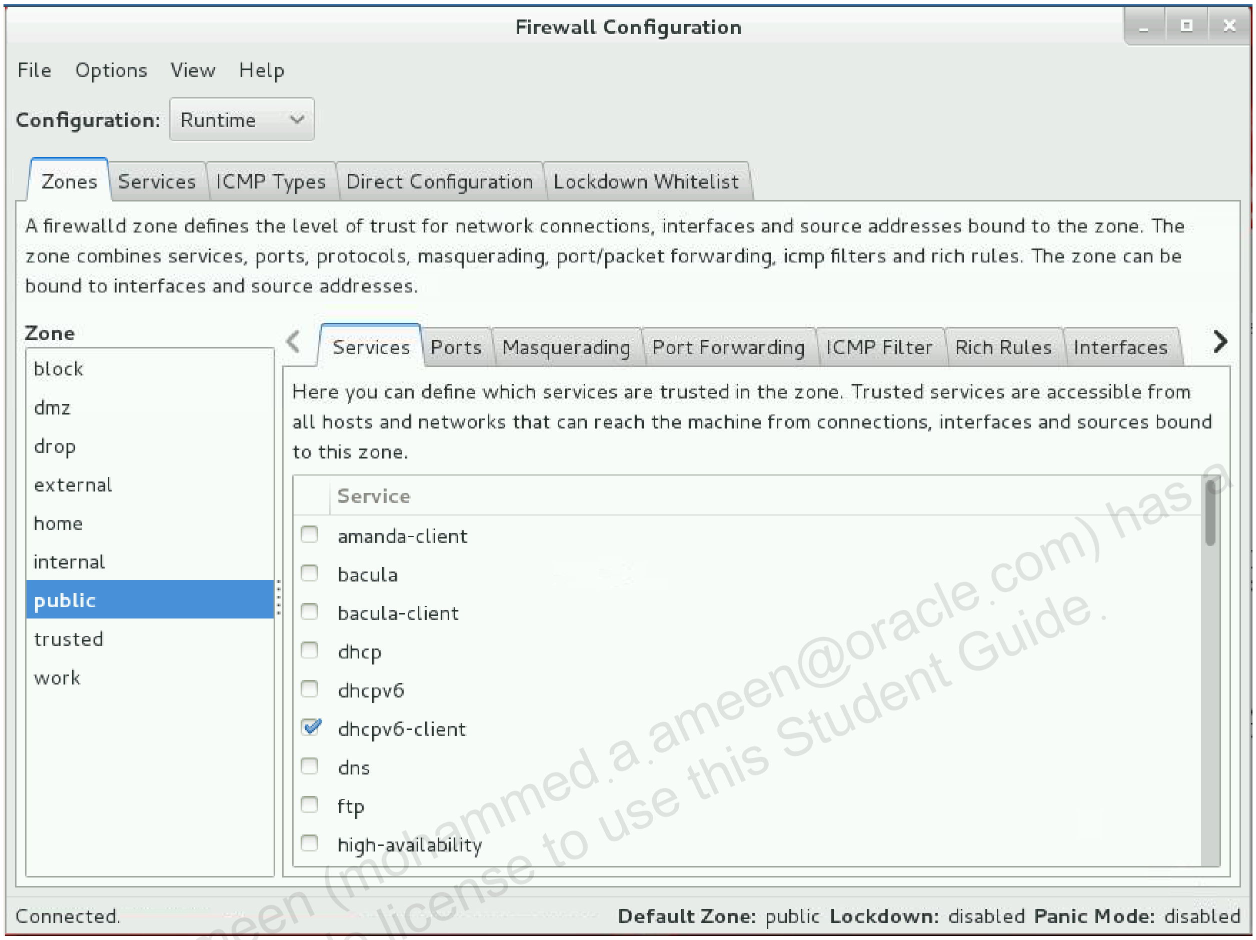


- Notice the menu under the list of zones is no longer displayed.
- When **Runtime** is selected, changes to current firewall settings take effect immediately.

c. Make the following selections from the Firewall Configuration GUI:

- Select the **Runtime** configuration option.
- Select **Zones** from the top row of tabs.
- Select **Services** from the lower row of tabs.
- Select the `public` zone from the list of Zones.

- The Firewall Configuration window appears as shown:



d. From a command prompt, use the `grep` command to search for the "`service`" string in the `/usr/lib/firewalld/zones/public.xml` zone file.

```
# grep service /usr/lib/firewalld/zones/public.xml
    <service name="ssh"/>
    <service name="dhcpv6-client"/>
```

- Note that these services correspond with the trusted services for the `public` zone.

e. In the Firewall Configuration **Service** window, select `dhcp` to trust this service.

- The **Service** window appears as shown:

- With the **Runtime** configuration selected, this service is trusted immediately by the network connections in the selected zone.
- It is not necessary to restart the `firewalld` service and disrupt the existing network connections and services.

f. From a command prompt, rerun the `grep` command from step 5d.

```
# grep service /usr/lib/firewalld/zones/public.xml
    <service name="ssh"/>
    <service name="dhcpv6-client"/>
```

- Note that this file was not updated. There is no entry for the `dhcp` service.

g. Use the `grep` command to search for the string "`service`" in the `/etc/firewalld/zones/public.xml` zone file.

```
# grep service /etc/firewalld/zones/public.xml
    <service name="ssh"/>
    <service name="dhcpv6-client"/>
```

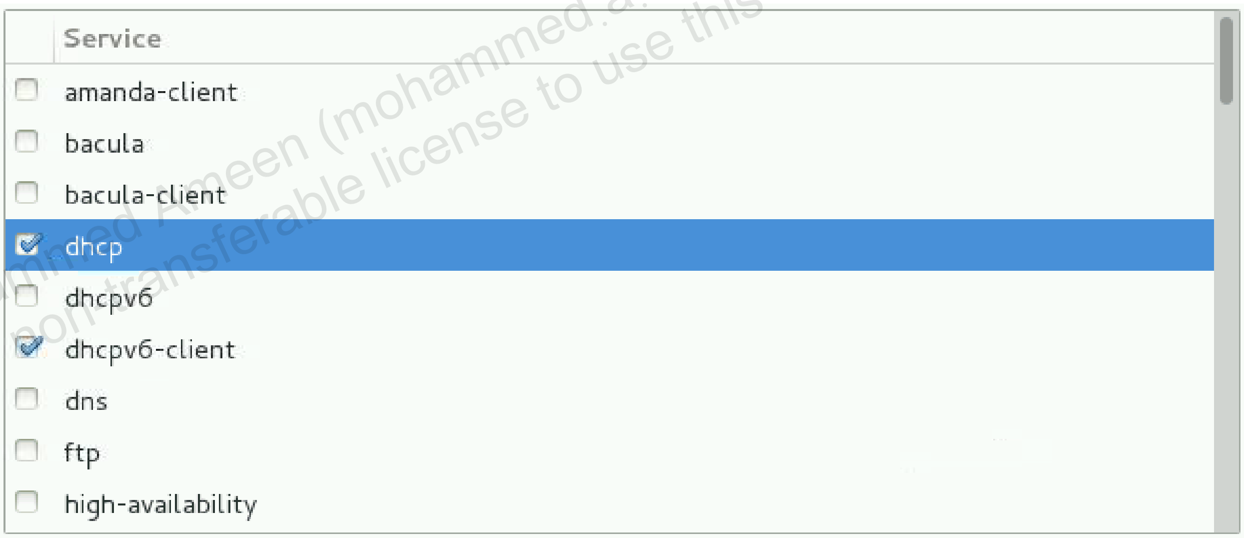
- Note that this file was not updated. There is no entry for the `dhcp` service.
- Configuration files are not updated in **Runtime** configuration mode.
- Configuration changes made in **Runtime** configuration mode are lost when the `firewalld` service is restarted.

h. From the Firewall Configuration GUI, select the **Permanent** configuration option.

- Note that in the **Service** window, the `dhcp` service is no longer trusted.

i. In the Firewall Configuration **Service** window, select `dhcp` to trust this service.

- The **Service** window appears as shown:



- In the lower left corner of the GUI, the message "Changes applied" appears briefly. This only means that the configuration file was update. The change does not take effect until the `firewalld` service is restarted.

j. From a command prompt, rerun the `grep` command from steps 5d and 5f.

```
# grep service /usr/lib/firewalld/zones/public.xml
    <service name="ssh"/>
    <service name="dhcpv6-client"/>
```

- Note that there is no entry for the `dhcp` service.
- Configuration files in `/usr/lib/firewalld` are not updated when changes are made.

k. Rerun the `grep` command from step 5g.

```
# grep service /etc/firewalld/zones/public.xml
    <service name="dhcp"/>
    <service name="dhcpv6-client"/>
    <service name="ssh"/>
```

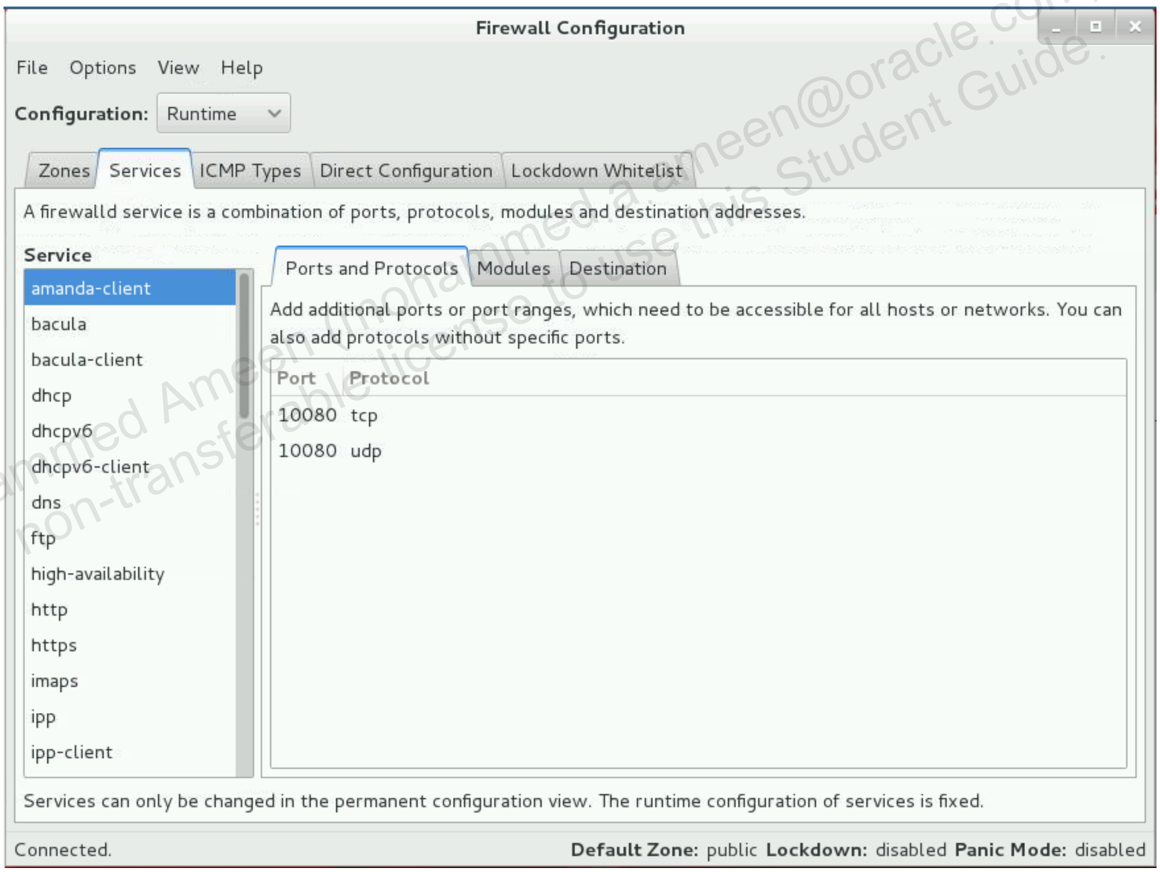
- Note that there is an entry for the `dhcp` service.
- The configuration files in `/etc/firewalld` are updated when changes are made.
- In this example, the `dhcp` service is trusted after restarting the `firewalld` service.

l. Click the **Configuration** drop-down menu and select **Runtime**.

6. Explore `firewalld` services.

a. Select the **Services** tab from the top row of tabs.

- The Firewall Configuration GUI appears as shown:



- Note that there are several predefined services.
- Each of these zones is defined by individual files in the `/usr/lib/firewalld/services` directory.

b. Use the `ls` command to display the contents of the `/usr/lib/firewalld/services` directory.

```
# ls /usr/lib/firewalld/services/
amanda-client.xml  ipp-client.xml  mysql.xml  rpc-bind.xml
```

```
bacula-client.xml  ipp.xml          nfs.xml     samba-client.xml
bacula.xml         ipsec.xml        ntp.xml     samba.xml
...
```

- Note that there is an XML configuration file for each of the predefined services.

c. Use the `firewall-cmd` command to display the available services.

```
# firewall-cmd --get-services
amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns
ftp high-availablity http https imaps ipp ipp-client ipsec
...
```

- Note that this command shows the same predefined services.

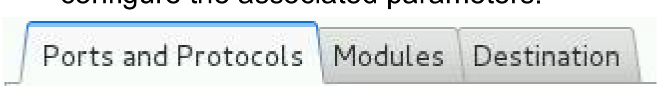
d. Use the `cat` command to display the contents of the `/usr/lib/firewalld/services/samba.xml` file.

```
# cat /usr/lib/firewalld/services/samba.xml
...
<service>
  <short>Samba</short>
  <description>This option allows you to access and participate
in Windows file and printer sharing networks. You need the samba
package installed for this option to be useful.</description>
  <port protocol="udp" port="137"/>
  <port protocol="udp" port="138"/>
  <port protocol="tcp" port="139"/>
  <port protocol="tcp" port="445"/>
  <module name="nf_conntrack_netbios_ns"/>
</service>
```

e. From the Firewall Configuration GUI, select the **Permanent** configuration option.

- Services can only be changed in **Permanent** configuration mode.

f. From the GUI, note that for each service you can select the lower row of tabs and configure the associated parameters.

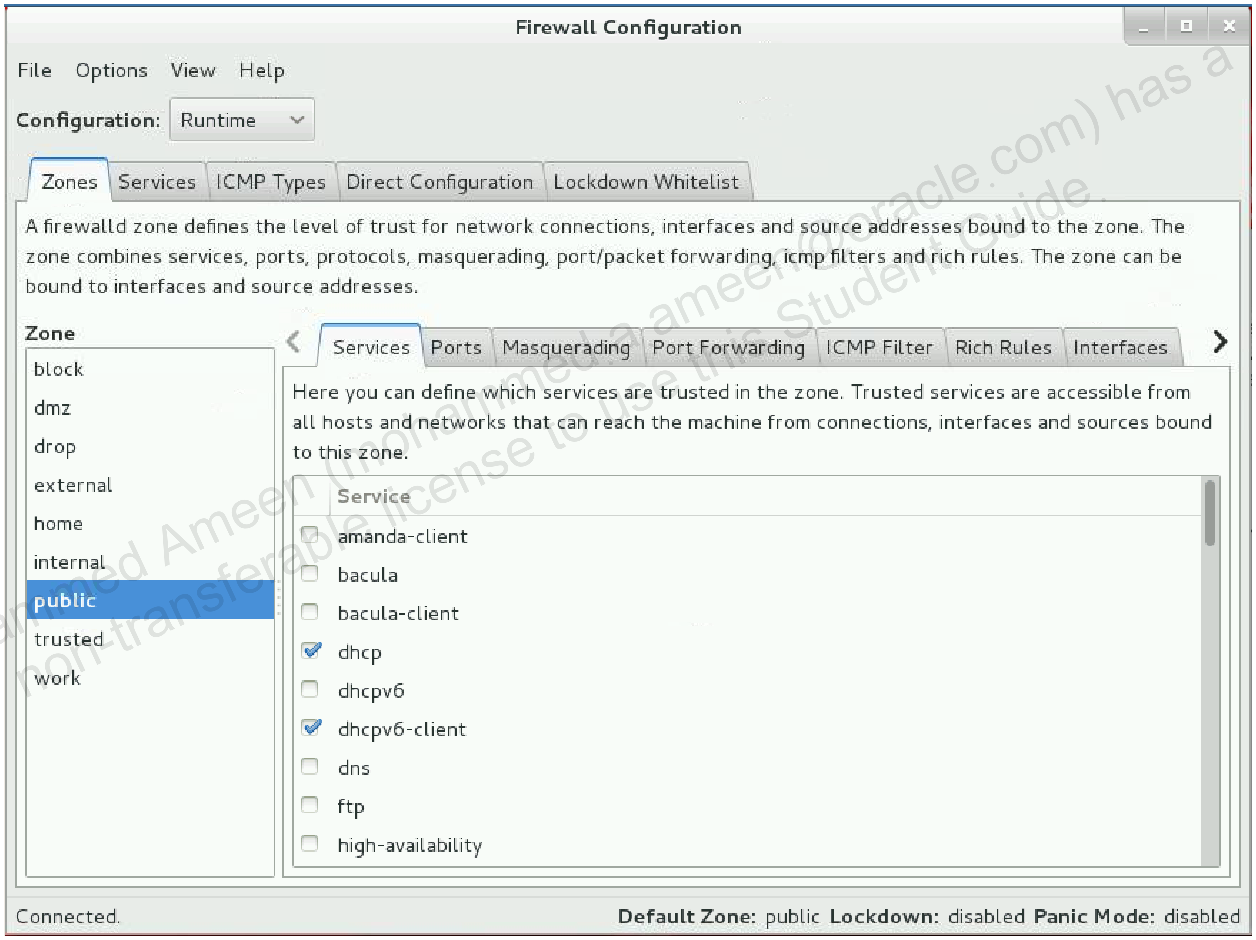


- Select the **Ports and Protocols** tab.
  – Click the **Add** button.
  – Note that you can add, change, or remove ports and protocols for the selected service.
  – Click the **Cancel** button.
- Select the **Modules** tab.
  – Click the **Add** button.
  – Note that you can add, change, or remove Netfilter helper modules for the selected service.

- Click the **Cancel** button.
- Select the **Destination** tab.
  - Note that you can limit traffic to a particular destination address and Internet Protocol (IPv4 or IPv6).

g. To prepare for the next practice, make the following selections from the Firewall Configuration GUI:

- Select the **Runtime** configuration option.
- Select **Zones** from the top row of tabs.
- Select **Services** from the lower row of tabs.
- Select the `public` zone from the list of Zones.
- The Firewall Configuration window appears as shown:



Practice 18-4: Configuring `firewalld`

Overview

In this practice, you create a `firewalld` rule to trust NFS.

- In Practice 16-1, you configured an NFS server and NFS client.
- However, you needed to stop the `firewalld` service before the NFS client was able to mount the exported file system.
- In this practice, you create a `firewalld` rule to trust NFS so that the NFS client can mount the exported file system with the `firewalld` service running.

Assumptions

- This practice is performed on **host01** and **host03** VMs.
- Open a terminal window on each system.
- Log in as the `root` user on each system.
- The prompts in the solution section include either **host01** or **host03** to indicate which system to enter the command from.
- This practice assumes you completed Practice 16-1.

Tasks

1. Review the NFS server configuration on **host03**.
   a. Use the `cat` command to view the `/etc/exports` file on **host03**.

```
[host03]# cat /etc/exports
/Dev *(rw,no_root_squash)
```

- The `/Dev` file system is exported to all client systems.
- 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.

   b. Use the `showmount -e` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Dev *
```

- This confirms that the `/Dev` file system is exported to all client systems.

2. Attempt to mount the exported `/Dev` file system on **host01**.
   a. On **host01**, use the `ls` command to list the contents of the `/remote_dev` directory.

```
[host01]# ls /remote_dev
```

- In this example, the `/remote_dev` directory is empty.

   b. Use the `mount` command to mount the exported file system from **host03**, `/Dev`, 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.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
mount.nfs: mount system call failed
```

- In this example, the `mount` command fails because of the firewall on **host03**.
- It takes several seconds for the `mount` command to timeout. Rather than wait, you can press Ctrl + C to abort the `mount` command.

3. From **host03**, use the Firewall Configuration GUI to trust the `nfs` service.
   a. In the Firewall Configuration **Service** window, scroll down if necessary and select `nfs` to trust this service.
      - The **Service** window appears as shown:

      - In **Runtime** configuration mode, this change takes effect immediately.

4. Attempt to mount the exported `/Dev` file system on **host01**.
   a. Re-issue the `mount` command from step 2b on **host01**.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
```

      - The `mount` command is successful.

   b. On **host01**, use the `df` command to display the mounted file systems.

```
[host01]# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvda2     5.7G  1.1G   4.3G   20%  /
...
host03:/Dev    976M  2.5M   907M    1%  /remote_dev
```

      - Note that the `host03:/Dev` file system is mounted on the local file system `/remote_dev`.

5. Unmount the exported file system on **host01**.
   a. Use the `umount` command to unmount `/remote_dev` on **host01**.

```
[host01]# umount /remote_dev
```

   b. On **host01**, use the `df` command to display the mounted file systems.

```
[host01]# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvda2     5.7G  1.1G   4.3G   20%  /
...
```

- Note that the `host03:/Dev` file system is no longer mounted on the local file system `/remote_dev`.

6. From **host03**, use the `firewall-cmd` command to permanently trust the `nfs` service.
   a. From a command prompt on **host03**, use the `firewall-cmd` command to permanently trust the `nfs` service.

```
[host03]# firewall-cmd --permanent --zone=public --add-service=nfs
success
```

   b. Use the `systemctl` command to restart the `firewalld` service on **host03**.

```
[host03]# systemctl restart firewalld
```

7. Attempt to mount the exported `/Dev` file system on **host01**.
   a. Re-issue the `mount` command from step 4a on **host01**.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
```

   - The `mount` command is successful.

   b. On **host01**, use the `df` command to display the mounted file systems.

```
[host01]# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvda2     5.7G  1.1G   4.3G   20%  /
...
host03:/Dev    976M  2.5M   907M    1%  /remote_dev
```

   - Note that the `host03:/Dev` file system is mounted on the local file system `/remote_dev`.

8. Restore initial configuration in preparation for next practice.
   a. Use the `umount` command to unmount `/remote_dev` on **host01**.

```
[host01]# umount /remote_dev
```

   b. From **host03**, use the `grep` command to search for the string "`service`" in the `/etc/firewalld/zones/public.xml` file.

```
[host03]# grep service /etc/firewalld/zones/public.xml
<service name="dhcp"/>
<service name="dhcpv6-client"/>
<service name="nfs"/>
<service name="ssh"/>
```

   c. Remove the "`dhcp`" and "`nfs`" service entries from the `/etc/firewalld/zones/public.xml` file.

   - You can use the `vi` editor to edit the file and delete the entries, or you can use the `firewall-cmd` command as shown.

```
[host03]# firewall-cmd --permanent --zone=public --remove-service=nfs
success
[host03]# firewall-cmd --permanent --zone=public --remove-service=dhcp
success
```

d. From **host03**, use the `grep` command to search for the "`service`" string in the `/etc/firewalld/zones/public.xml` file.

```
[host03]# grep service /etc/firewalld/zones/public.xml
<service name="dhcpv6-client"/>
<service name="ssh"/>
```

- Note that the entries for `ssh` and `dhcp` no longer exist.

e. Use the `systemctl` command to restart the `firewalld` service on **host03**.

```
[host03]# systemctl restart firewalld
```

f. From the Firewall Configuration GUI on **host03**, view the entries in the **Service** window.

- Note that the only trusted services are `dhcpv6-client` and `ssh`.

g. Quit the Firewall Configuration GUI on **host03** by selecting **File->Quit** from the menu bar.

Practice 18-5: Configuring `iptables`

Overview

In this practice, you use the `iptables` command to allow a client system to mount an NFS file system.

Assumptions

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

Tasks

1. From **host03**, stop the `firewalld` service and start `iptables`.
   a. Use the `systemctl` command to stop the `firewalld` service.

```
[host03]# systemctl stop firewalld
```

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

```
[host03]# systemctl start iptables
```

   c. Use the `iptables -L` command to list all the rules in all the chains.

```
[host03]# iptables -L
Chain INPUT (policy ACCEPT)
target   prot  opt  source     destination
ACCEPT   all   --   anywhere   anywhere        state ...
...
Chain FORWARD (policy ACCEPT)
target   prot  opt  source     destination
REJECT   all   --   anywhere   anywhere        reject...
...
Chain OUTPUT (policy ACCEPT)
target   prot  opt  source     destination
```

   d. Run the `iptables -L` command again, but this time pipe the output to `grep` and search for “`nfs`”.

```
[host03]# iptables -L | grep nfs
```

   - Note that there are currently no rules containing the “`nfs`” string.

2. Attempt to mount the NFS exported file system on **host03** from a remote host, **host01**.
   a. From **host03**, use the `showmount -e` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/Dev *
```

   - This confirms that the `/Dev` file system is exported to all client systems.

b. From **host01**, use the `mount` command to mount the exported `/Dev` NFS file system from **host03**.

- Mount with `rw` and `nosuid` options on the local mountpoint, `/remote_dev`.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
```

- With the `iptables` service enabled, and no firewall rule to trust `nfs`, the `mount` command fails.
- Press Ctrl + C to abort the `mount` command.

3. From **host03**, view and modify the `iptable` rules.

a. Run `iptables -h` to display all options.

```
[host03]# iptables -h
iptables v1.4.21

Usage: iptables -[ACD] chain rule-specification [options]
       iptables -I chain [rulenum] rule-specification ...
...
  --line-numbers      print line numbers when listing
...
```

- From the help, note that the `--line-numbers` option displays line numbers.

b. Run the `iptables` command to list only those rules in the `INPUT` chain and include line numbers.

```
[host03]# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target   prot  opt  source     destination
1    ACCEPT   all   --   anywhere   anywhere     state ...
...
4    ACCEPT   tcp   --   anywhere   anywhere     state ...ssh
...
```

- Note that line number 4 contains a rule to accept `ssh` traffic.
- You need to create a similar entry to accept `nfs` traffic.

c. Use the `iptables` command to insert the “`nfs`” rule before line 4 with the following characteristics from the command line:

- Chain = `INPUT`
- Protocol = `tcp`
- State = `NEW`
- Destination port = `nfs`
- Target = `ACCEPT`

```
[host03]# iptables -I INPUT 4 -p tcp -m state --state NEW --
dport nfs -j ACCEPT
```

- This rule accepts incoming `tcp` traffic for `nfs`.

d. Repeat step 3b to list the new nfs rule.

```
 [host03]# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target   prot  opt  source     destination
1    ACCEPT   all   --   anywhere   anywhere     state ...
...
4    ACCEPT   tcp   --   anywhere   anywhere     state ...nfs
5    ACCEPT   tcp   --   anywhere   anywhere     state ...ssh
...
```

- Note that there now is a rule to accept `nfs` traffic.

e. Use the `cat` command to view the `/etc/sysconfig/iptables` file.

```
[host03]# cat /etc/sysconfig/iptables
...
-A INPUT -p tcp -m state -state NEW -m tcp -dport 22 -j ACCEPT
...
```

- Note that there is a rule for `ssh` (port `22`) but not for `nfs`.
- You still need to save the `iptables` rules to the `/etc/sysconfig/iptables` file.

f. Use the `service` command to save the `iptables` rules.

```
[host03]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables...
```

g. Use the `cat` command to view the `/etc/sysconfig/iptables` file.

```
[host03]# cat /etc/sysconfig/iptables
...
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2049 -j
ACCEPT
...
```

- Note that the new rule has the actual port number, `2049`, for `nfs`.

h. Use the `grep` command to search for `2049` in the `/etc/services` file.

```
[host03]# grep 2049 /etc/services
nfs  2049/tcp ...
```

i. Use the `systemctl` command to restart the `iptables` service.

```
[host03]# systemctl restart iptables
```

4. From **host01**, attempt to mount the NFS file system.

a. Use the `mount` command to mount `host03:/Dev` on `/remote_dev`.

```
[host01]# mount -t nfs -o rw,nosuid host03:/Dev /remote_dev
```

- The `mount` command is successful this time.

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

```
[host01]# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvda2     5.7G  1.1G   4.3G   20%  /
...
host03:/Dev    976M  2.5M   907M    1%  /remote_dev
```

- Note that the `host03:/Dev` file system is mounted on local file system `/remote_dev`.

c. From **host01**, use the `umount` command to unmount `/remote_dev`.

```
[host01]# umount /remote_dev
```

Practice 18-6: Configuring a TCP Wrapper

Overview

In this practice, you configure a TCP wrapper to deny one system from using OpenSSH utilities to connect to another system. You also create a custom log file to capture connection attempts that are denied.

Assumptions

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

Tasks

1. From **host01**, confirm that you can use the `ssh` command to connect to **host03**.
   - Password is `oracle`.
   - Use the `logout` command to log off after confirming that you can connect.

```
[host01]# ssh host03
root@host03's password: oracle
Last login: ...
[root@host03 ~]# logout
Connection to host03 closed.
[host01]#
```

2. On **host03**, configure a TCP wrapper to deny host01 from using OpenSSH utilities to connect.
   a. Use the `vi` editor to edit `/etc/hosts.deny` and add the following entry.

```
[host03]# vi /etc/hosts.deny
sshd : 192.0.2.101
```

   - This entry denies **host01** (`192.0.2.101`) from using the OpenSSH utilities to connect to **host03**.

   b. From **host01**, attempt to use the `ssh` command to connect to **host03**.

```
[host01]# ssh host03
ssh_exchange_identification: Connection closed by remote host
```

   - This time you are denied a connection.

3. On **host03**, modify the TCP wrapper to write a message to a log file.
   a. Use the `vi` editor to edit `/etc/hosts.deny` and modify the entry as follows.

```
[host03]# vi /etc/hosts.deny
sshd : 192.0.2.101 : spawn /bin/echo "%c tried to connect to %d
and was blocked." >> /var/log/tcpwrappers.log
```

b. From **host01**, attempt to use the ssh command to connect to **host03**.

```
[host01]# ssh host03
ssh_exchange_identification: Connection closed by remote host
```

- You are still denied a connection and a message is written to a log file.

c. On **host03**, use the cat command to view the /var/log/tcpwrappers.log file.

```
[host03]# cat /var/log/tcpwrappers.log
192.0.2.101 tried to connect to sshd and was blocked.
```

4. Restore the system.

a. On **host03**, use the vi editor to edit /etc/hosts.deny and delete the entry created earlier in this practice.

- Delete the strike-through line as follows:

```
[host03]# vi /etc/hosts.deny
sshd : 192.0.2.101 : spawn /bin/echo "%c tried to connect to %d
and was blocked." >> /var/log/tcpwrappers.log
```

b. From **host01**, confirm that you can use the ssh command to connect to **host03**.

- Password is oracle.
- Use the logout command to log off after confirming that you can connect.

```
[host01]# ssh host03
root@host03's password: oracle
Last login: ...
[root@host03 ~]# hostname
host03.example.com
[root@host03 ~]# logout
Connection to host03 closed.
[host01]#
```

Practices for Lesson 19: Oracle on Oracle

Chapter 19

Practices for Lesson 19: Overview

Practices Overview

In these practices, you install and run Oracle RDBMS Pre-Install RPM for Oracle Linux 7. You also configure ASMLib.

Practice 19-1: Using `sftp` to Upload `oracle` Packages

Overview

In this practice, you use `sftp` to upload the `oracle-rdbms-server-12cR1-preinstall` package and the `oracleasmlib` package from the **dom0** to the **host03** VM. Normally, you obtain these packages from ULN or from the Public Yum Server.

Assumptions

- You are the `root` user on the **host03** VM.
- You are the `root` user on **dom0**.

Tasks

1. Use the `sftp` to transfer the `oracle` packages from **dom0** to **host03**.
   a. From **dom0**, use the `cd` command to change to the `/OVS/seed_pool/sfws` directory.

   ```
   [dom0]# cd /OVS/seed_pool/sfws
   ```

   b. Use the `ls` command to view the directory for the `oracle*` package.

   ```
   [dom0]# ls oracle*
   oracleasmlib-2.0.8-2.el7.x86_64.rpm
   oracle-rdbms-server-12cR1-preinstall-1.0-1.el7.x86_64.rpm
   ```

   c. Use the `sftp` command to connect to **host03** as `root`.
      - Password is `oracle`.

   ```
   [dom0]# sftp host03
   root@host03's password: oracle
   sftp>
   ```

   d. Use the `mput` command to copy the `oracle*` packages to **host03**.

   ```
   sftp> mput oracle*
   Uploading oracle-rdbms-server-12cR1-preinstall-1.0-
   1.el7.x86_64.rpm to /root/...
   Uploading oracleasmlib-2.0.8-2.el7.x86_64.rpm to /root/...
   ...
   ```

   e. Use the `quit` command to exit `sftp`.

   ```
   sftp> quit
   ```

Practice 19-2: Installing and Running Oracle RDBMS Pre-Install

Overview

In this practice, you install the `oracle-rdbms-server-12cR1-preinstall` package, run the "`verify`" program, and view the results.

Assumptions

You are the `root` user on **host03** VM.

Tasks

1. Install the `oracle-rdbms-server-12cR1-preinstall` package and view the results of the installation.
   a. Use the `cd` command to change to the `root` user's home directory, and then use the `yum` command to install the `oracle-rdbms-server-12cR1-preinstall` package.
      - Answer **`y`** when prompted.

```
# cd
# yum install oracle-rdbms-server-12cR1-preinstall-1.0-
1.el7.x86_64.rpm
...
Transaction Summary
=========================================================
Install  1 Package (+6 Dependent packages)
Total size: 9.8 M
Total download size: 9.7 M
Installed size: 28 M
Is this ok [y/d/N] y
Downloading Packages:
...
Installed:
  oracle-rdbms-server-12cR1-preinstall.x86_64 0:1.0-1.el7

Dependency Installed:
...
Complete!
```

   b. Use the `find` command to locate all `oracle-rdbms*` files.

```
# find / -name "*oracle-rdbms*"
/usr/bin/oracle-rdbms-server-12cR1-preinstall-verify
/etc/sysconfig/oracle-rdbms-server-12cR1-preinstall
/etc/sysconfig/oracle-rdbms-server-12cR1-preinstall/oracle-
rdbms-server-12cR1-preinstall.param
/etc/sysconfig/oracle-rdbms-server-12cR1-preinstall/oracle-
rdbms-server-12cR1-preinstall.conf
```

```
/etc/sysconfig/oracle-rdbms-server-12cR1-preinstall/oracle-
rdbms-server-12cR1-preinstall-verify
/etc/rc.d/init.d/oracle-rdbms-server-12cR1-preinstall-firstboot
/etc/security/limits.d/oracle-rdbms-server-12cR1-preinstall.conf
/var/log/oracle-rdbms-server-12cR1-preinstall
...
```

c. Use the `ls -l` command to display the file type of the `/usr/bin/oracle-rdbms*` file.

```
# ls –l /usr/bin/oracle-rdbms*
lrwxrwxrwx ... /usr/bin/oracle... -> /etc/sysconfig/oracle...
```

- Note that this file is a symbolic link to a file in the `/etc/sysconfig/oracle-rdbms*` directory.

d. Change to the `/etc/sysconfig/oracle-rdbms*` directory and view the contents of the directory.

```
# cd /etc/sysconfig/oracle-rdbms*
# ls
oracle-rdbms-server-12cR1-preinstall.conf
oracle-rdbms-server-12cR1-preinstall.param
oracle-rdbms-server-12cR1-preinstall-verify
```

e. Use the `less` command to view each of the files in the `/etc/sysconfig/oracle-rdbms*` directory.

```
# less oracle-rdbms-server-12cR1-preinstall.conf
...
# less oracle-rdbms-server-12cR1-preinstall.param
...
# less oracle-rdbms-server-12cR1-preinstall-verify
...
```

- Note that the `*.param` file is the main configuration file.
- Note that the `*verify` file is the Bash script that modifies settings.

2. View the modifications made by the `oracle-rdbms-server-12cR1-preinstall-verify` script.
   - Note that previous versions of the `oracle-rdbms-server-...-verify` script needed to be run manually. With this version, however, the script is executed automatically when the RPM package is installed.

a. Use the `less` command to view the Oracle RDBMS Pre-install log file, `/var/log/oracle-rdbms-server-12cR1-preinstall/results/orakernel.log` (sample output shown).

```
# cd /var/log/oracle-rdbms-server-12cR1-preinstall/results
# less orakernel.log
Adding group oinstall with gid 54321
Adding group dba
User oracle is already present
```

```
uid=1000(oracle) gid=1000(oracle) ...
Creating oracle user passed

Verifying kernel parameters as per Oracle recommendations...
Adding fs.file-max = 6815744
Adding kernel.sem = 250 32000 100 128
Adding kernel.shmmni = 4096
Adding kernel.shmall = 1073741824
Adding kernel.shmmax = 4398046511104
Adding kernel.panic_op_oops = 1
...
Setting kernel parameters as per oracle recommendations...
Altered file /etc/sysctl.conf
Original file backed up at /etc/sysctl.conf.orabackup
Verifying & setting of kernel parameters passed

Setting user limits using /etc/security/limits.d/oracle...

Verifying oracle user OS limits as per Oracle recommendations...
Adding oracle soft nofile 1024
Adding oracle hard nofile 65536
Adding oracle soft nproc 16384
Adding oracle hard nproc 16384
Adding oracle soft stack 10240
Adding oracle hard stack 32768
Setting oracle user OS limits as per oracle recommendations...
Altered file /etc/security/limits.d/oracle-rdbms...
Original file backed up at /var/log/oracle-rdbms...
<date_time>
Verifying & setting of user limits passed

Verifying kernel boot parameters as per Oracle ...
old boot params: ...
...

Setting kernel boot parameters as per Oracle recommendations...
Generating grub configuration file ...
...
done
Boot parameters will be effected on next reboot
Altered file /etc/default/grub
Original file backed up at /etc/default/grub.orabackup
```

```
Verifying & setting of boot parameters passed

Taking a backup of old config files under /var/log/oracle...
```

- Note that the required user and groups are created if necessary.
- Note that kernel parameters are set and `/etc/sysctl.conf` is backed up beforehand.
- Note the "`kernel.panic_on_oops = 1`" setting. This changes a kernel oops into a panic. See Oracle Bug 19212317 at https://bug.oraclecorp.com/ for more information.
- Note that `oracle` user OS limits are set in `/etc/security/limits.d/oracle-rdbms-server-12cR1-preinstall.conf`.
- Note that kernel boot parameters are set and `/etc/default/grub` is backed up beforehand.
- While not recorded in the `orakernel.log` file, the `/boot/grub2/grub.cfg` file is also backed up.

b. Use the `find` command to list the files backed up before settings were changed, `*orabackup`.

```
# find / -name "*orabackup"
/boot/grub2/grub.cfg.orabackup
/etc/sysctl.conf.orabackup
/etc/default/grub.orabackup
```

c. Use the `diff` command to view the changes made in the `grub.cfg` file.

```
# diff /boot/grub2/grub.cfg /boot/grub2/grub.cfg.orabackup
...
```

- Note that the "`numa=off`" boot parameter is added to the `grub.cfg` file.
- Note that the "`transparent_hugepage=never`" boot parameter is added to the `grub.cfg` file.

d. Use the `grep` command and search for the string "`numa`" and "`transparent`" in the `grub.cfg` and `grub.cfg.orabackup` and note that these boot parameters were added.

```
# grep numa /boot/grub2/grub.cfg
... numa=off transparent_hugepage=never
...
# grep transparent /boot/grub2/grub.cfg.orabackup
```

e. Use the `wc -l` command to display the number of lines in the `/etc/sysctl.conf` files.

```
# wc -l /etc/sysctl.conf /etc/sysctl.conf.orabackup
41 /etc/sysctl.conf
 4 /etc/sysctl.conf.orabackup
...
```

- Note that 37 new lines are added to the `sysctl.conf` file.

f. Use the `diff` command to view the changes made in the `sysctl.conf` file.

```
# diff /etc/sysctl.conf /etc/sysctl.conf.orabackup
...
```

g. Use the `diff` command to view the changes made in the `/etc/default/grub` file.

```
# diff /etc/default/grub /etc/default/grub.orabackup
...
```

- Note that the "`transparent_hugepage=never`" boot parameter is added to the `etc/default/grub` file.

h. Use the `cat` command to view the limits set in the `/etc/security/limits.d/oracle-rdbms-server-12cR1-preinstall.conf` file.

```
# cat /etc/security/limits.d/oracle-rdbms-server-12cR1-
preinstall.conf
...
```

Practice 19-3: Preparing Disk for ASM Use

Overview

In this practice, you:

- Remove all NFS file systems
- Unmount all file systems on /dev/xvdb and /dev/xvdd
- Delete all existing partitions to use the disks for ASM
- Create one partition using the entire disk on /dev/xvdb
- Create one partition using the entire disk on /dev/xvdd

Assumptions

You are the root user on **host03** VM.

Tasks

1. Remove all NFS file systems.
   a. Use the vi editor to delete all entries from /etc/exports.

```
# vi /etc/exports
Delete all entries and save the file
```

   b. Use the systemctl command to restart the nfs service.

```
# systemctl restart nfs
```

   c. Use the showmount command to confirm that there are no exported file systems.

```
# showmount -e
Export list for host03.example.com
```

   - No NFS file systems are being exported.

2. Unmount all file systems on /dev/xvdb and /dev/xvdd.
   a. Use the df command to list the mounted partitions.

```
# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
/dev/xvda2     5.7G  4.7G   760M   87%  /
...
/dev/xvdb1     976M  2.6M   907M    1%  /Dev
/dev/xvdd1     976M  2.6M   907M    1%  /Test
```

   - In this example, the following partitions need to be unmounted (your system might be different).
     - /dev/xvdb1 mounted on /Dev
     - /dev/xvdd1 mounted on /Test

   b. Use the umount command to unmount file systems on /dev/xvdb1 and /dev/xvdd1.

```
# umount /Dev
# umount /Test
```

3. Delete all partitions on /dev/xvdb and /dev/xvdd.
   a. Use the fdisk command to display the partition table on /dev/xvdb and then delete all the partitions.

```
# fdisk /dev/xvdb
...
Command (m for help): p
Disk /dev/xvdb: 5368 MB, 5368709120 bytes, 10485760 sectors
...
    Device Boot   Start        End      Blocks    Id    System
/dev/xvdb1          ...
/dev/xvdb2          ...

Command (m for help): d
Partition number (1,2, default 2): ENTER
Partition 2 is deleted

Command (m for help): d
Selected partition 1
Partition 1 is deleted

Command (m for help): p
...
    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.
```

   - In this example, two partitions are deleted (your system might be different).

   b. Use the fdisk command to display the partition table on /dev/xvdd and then delete all the partitions.

```
# fdisk /dev/xvdd
...
Command (m for help): p
Disk /dev/xvdb: 5368 MB, 5368709120 bytes, 10485760 sectors
...
    Device Boot   Start        End      Blocks    Id    System
/dev/xvdd1          ...
/dev/xvdd2          ...

Command (m for help): d
```

```
Partition number (1,2, default 2): ENTER
Partition 2 is deleted

Command (m for help): d
Selected partition 1
Partition 1 is deleted

Command (m for help): p
...
    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.
```

- In this example, two partitions are deleted (your system might be different).

c. Use the `vi` editor to delete entries pertaining to file systems on `/dev/xvdb` and `/dev/xvdd` from the `/etc/fstab` file.

```
# vi /etc/fstab
/dev/xvdb1    /Dev    ... (delete this entry)
/dev/xvdd1    /Test   ... (delete this entry)
```

- In this example, delete the two entries shown (your system might be different).

4. In preparation for the ASMLib lab, you must disable SELinux completely and reboot.
   - ASMlib is not supported with SElinux enabled (it is a known issue).

Use the `vi` editor to edit `/etc/selinux/config` as follows:

```
# vi /etc/selinux/config
SELINUX=enforcing                                 (old value)
SELINUX=disabled                                  (new value)
```

5. Reboot your system and log back in.

a. Use the `systemctl` command to reboot your system.
   - It might take a couple minutes for the reboot to complete.

```
# systemctl reboot
...
```

- After you reboot your system, your vnc session closes.

b. Connect to **host03** by using VNC.

1) Run the `vncviewer&` command.

```
# vncviewer&
```

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

2) Enter the command, `localhost:<port_number>`, substituting the correct port number for the **host03** guest. For example, if the port number is `5904`, enter the following and click Connect:

```
localhost:5904
```

c. Select `Oracle Student` from the GNOME login window; password is `oracle`.

d. Right-click the GNOME desktop and select **Open in Terminal** from the pop-up menu.

e. In the terminal window, become the `root` user by entering the `su -` command followed by the `root` password, `oracle`.

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

6. Create a partition on `/dev/xvdb`.

a. Use the `fdisk` command to partition `/dev/xvdb`.

```
# fdisk /dev/xvdb
...
Command (m for help):
```

b. Add a new **primary** partition, giving the partition number **1**.

```
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
```

c. Continue adding the new partition, using the entire disk as follows:

```
First sector (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): ENTER
Using default value 10485759
Partition 1 of type Linux and of size 5 GiB is set
```

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.
```

7. Create a partition on `/dev/xvdd`.

a. Use the `fdisk` command to partition `/dev/xvdd`.

```
# fdisk /dev/xvdd
...
```

```
Command (m for help):
```

b. Add a new **primary** partition, giving the partition number **1**.

```
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
```

c. Continue adding the new partition, using the entire disk as follows:

```
First sector (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): ENTER
Using default value 10485759
Partition 1 of type Linux and of size 5 GiB is set
```

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 19-4: Installing and Configuring ASMLib

Overview

In this practice, you:

- Install the `oracleasmlib` package
- Install the `oracleasm-support` package
- Configure ASMLib
- Load and initialize the ASMLib driver
- Mark disk partitions for ASM use
- View information about the ASM disk partitions

Assumptions

You are the `root` user on **host03** VM.

Tasks

1. Install the `oracleasm` packages.
   a. Use the `cd` command to change to the `root` user's home directory, and then use the `yum` command to install the `oracleasmlib` package.
      - Answer **y** when prompted.

```
# cd
# yum install oracleasmlib-2.0.8-2.el7.x86_64.rpm
...
Transaction Summary
==========================================================
Install  1 Package

Total size: 35 k
Installed size: 35 k
Is this ok [y/d/N] y
Downloading Packages:
...
Installed:
  oracleasmlib.x86_64 0:2.0.8-2.el7

Complete!
```

   b. Use the `yum` command to install the `oracleasm-support` package.
      - Answer **y** when prompted.

```
# cd
# yum install oracleasm-support.x86_64
...
Transaction Summary
==========================================================
```

```
Install  1 Package

Total download size: 79 k
Installed size:242 k
Is this ok [y/d/N] y
Downloading Packages:
...
Installed:
  oracleasm-support.x86_64 0:2.1.8-3.el7

Complete!
```

2. Use the `oracleasm` utility to configure ASMLib.
   a. Run the `oracleasm -h` command to display the usage and commands.

```
# oracleasm –h
Usage: oracleasm [--exec-path=<exec_path>] <command> [ <args> ]
       oracleasm –exec-path
       oracleasm –h
       oracleasm –V

The basic oracleasm commands are:
    configure        Configure the Oracle Linux ASMLib driver
    init             Load and initialize the ASMLib driver
    exit             Stop the ASMLib driver
    scandisks        Scan the system for Oracle ASMLib disks
    status           Display the status of the Oracle ASMLib ...
    listdisks        List known Oracle ASMLib disks
    querydisk        Determine if a disk belongs to Oracle AS...
    createdisk       Allocate a device for Oracle ASMLib use
    deletedisk       Return a device to the operating system
    renamedisk       Change the label of an Oracle ASMLib disk
    update-driver    Download the latest ASMLib driver
```

   b. Use the `oracleasm configure –i` command to configure the ASMLib driver.

```
# oracleasm configure –i
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM
library driver. The following questions will determine whether
the driver is loaded on boot and what permissions it will have.
The current values will be shown in brackets (‘[]’). Hitting
<ENTER> without typing an answer will keep that current value.
Ctrl-C will abort.

Default user to own the driver interface []: oracle
Default group to own the driver interface []: dba
```

```
Scan for Oracle ASM disks on boot (y/n) [y]: ENTER
Writing Oracle ASM library driver configuration: done
```

c. Use the `oracleasm init` command to load and initialize the ASMLib driver.

```
# oracleasm init
Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
```

d. Use the `oracleasm configure` command without the `-i` option.

- This command shows the current configuration.

```
# oracleasm configure
ORACLEASM_UID=oracle
ORACLEASM_GID=dba
ORACLEASM_SCANBOOT=true
ORACLEASM_SCANORDER=""
ORACLEASM_SCANEXCLUDE=""
ORACLEASM_USE_LOGICAL_BLOCK_SIZE="false"
```

e. Use the `oracleasm status` command.

```
# oracleasm status
Checking if ASM is loaded: yes
Checking if /dev/oracleasm is mounted: yes
```

f. Use the `oracleasm createdisk` command to mark `/dev/xvdb1` for ASM use.

- Give the disk a label of `VOL1`.

```
# oracleasm createdisk VOL1 /dev/xvdb1
Writing disk header: done
Instantiating disk: done
```

g. Use the `oracleasm createdisk` command to mark `/dev/xvdd1` for ASM use.

- Give the disk a label of `VOL2`.

```
# oracleasm createdisk VOL2 /dev/xvdd1
Writing disk header: done
Instantiating disk: done
```

3. View ASM disks.

a. Use the `ls` command to display a long list of `/dev/oracleasm/disks` directory.

```
# ls -l /dev/oracleasm/disks
brw-rw----. 1 oracle  dba  ... VOL1
brw-rw----. 1 oracle  dba  ... VOL2
```

b. Use the `oracleasm listdisks` command to list the disk names of marked ASMLib disks.

```
# oracleasm listdisks
VOL1
VOL2
```

c. Use the `oracleasm scandisks` command to enable cluster nodes to identify which shared disks have been marked as ASMLib disks on another node.

```
# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
```

d. Use the `oracleasm querydisk` command to determine whether a disk name or disk device is being used by ASMLib.

```
# oracleasm querydisk VOL1
Disk "VOL1" is a valid ASM disk

# oracleasm querydisk /dev/xvdb1
Device "/dev/xvdb1" is marked an ASM disk with the label "VOL1"

# oracleasm querydisk /dev/xvdd1
Device "/dev/xvdd1" is marked an ASM disk with the label "VOL2"
```

Practices for Lesson 20: System Monitoring

Chapter 20

Practices for Lesson 20: Overview

Practices Overview

In these practices, you:

- Use the `sosreport` utility to collect system information
- Use standard Linux utilities to monitor system resource usage
- Use OSWatcher Black Box and OSWatcher Analyzer

Practice 20-1: Using `sosreport` to Collect System Information

Overview

In this practice, you:

- Use the `sosreport` utility to collect system information
- Extract the compressed TAR file and view the collected information
- View the status of the `sosreport` plug-ins

Assumptions

You are the `root` user on **host03** VM.

Tasks

1. Use the `rpm` command to verify that the `sos` packing is installed.

```
# rpm -q sos
sos-3.0-23.0.2.el7.noarch
```

   In this example, the package is installed.

2. Run the `sosreport` command.
   a. Press **Enter** when prompted to continue.
   b. Press **Enter** when prompted to enter your first initial and last name.
   c. Enter number `1` as the case number for which you are generating the report.

```
# sosreport
sosreport (version 3.0)

This command will collect diagnostic and configuration
information from this Oracle Linux system and installed
applications.

An archive containing the collected information will be
generated in /var/tmp and may be provided to a Oracle America
support representative.

Any information provided to Oracle America will be treated in
accordance with the published support policies at:

  https://linux.oracle.com/

The generated archive may contain data considered sensitive and
its content should be reviewed by the originating organization
before being passed to any third party.

No changes will be made to system configuration.

Press ENTER to continue, or CTRL-C to quit. ENTER
```

```
Please enter your first initial and last name [host03...]: ENTER
Please enter the case number that you are generating... 1
...
Running plugins. Please wait ...
...
Creating compressed archive...

Your sosreport has been generated and saved in:
  /var/tmp/sosreport-host03.example.com.1-...tar.xz

The checksum is: ...

Please send this file to your support representative.
```

3. View the `sosreport` file.
   a. Use the `cd` command to change to the `/tmp` directory.

```
# cd /var/tmp
```

   b. Use the `ls` command to display a long listing of the `sos*` files.

```
# ls -l sos*
-rw------- sosreport-host03.example.com.1-...tar.xz
-rw-r--r-- sosreport-host03.example.com.1-...tar.xz.md5
```

   - Note the two `sosreport` files, one with the `.xz` extension and one with the `.md5` extension.
   - The `.xz` file is the compressed data file.
   - Note the size of the `.xz` file.

   c. Use the `xz -d` command to uncompress the `.xz` file.

```
# xz -d sosreport-host03.example.com.1-...tar.xz
```

   d. Use the `ls` command to display a long listing of the `sos*` files.

```
# ls -l sos*
-rw------- sosreport-host03.example.com.1-...tar
-rw-r--r-- sosreport-host03.example.com.1-...tar.xz.md5
```

   - Note that the `sosreport` file with the `.tar.xz` extension now has a `.tar` extension.
   - Note the size of the `.tar` file—it is considerably larger than the compressed (`.xz`) file.

   e. Use the `tar` command to extract the `.tar` file.

```
# tar xvf sosreport-host03.example.com.1-...tar
...
```

   Note that the `tar` file is extracted in a `sosreport-host03...` directory.

   f. Use the `cd` command to change to the `sosreport-host03...` directory.

```
# cd sosreport-host03...
```

g. Use the `ls` command to display a long listing of the `sosreport-host03...` directory.

```
# ls -l
...
drwxr-xr-x          boot
lrwxrwxrwx          chkconfig -> sos_commands/startup/...
lrwxrwxrwx          date -> sos_commands/general/date
...
drwxr-xr-x          etc
lrwxrwxrwx          free -> sos_commands/memory/free
lrwxrwxrwx          hostname -> sos_commands/general/...
...
drwxr-xr-x          lib
...
```

- Note that a number of directories that contain data collected from the system exist.
- Note that a number of symbolic links that contain the output of several status-related commands exist.

4. Use the `sosreport -l` command to list the plug-ins.

```
# sosreport -l
sosreport (version 3.0)

The following plugins are currently enabled:

 abrt              ABRT log dump
 acpid             acpid related information
 anaconda          Anaconda / Installation information
...
The following plugins are currently disabled:
 apache            inactive  Apache related information ...
 ceph              inactive  information on CEPH
 cloudforms        Cloudforms related information
...
The following plugin options are available:
 abrt.backtraces     off   collect backtraces for every ...
 auditd.syslogsize   15    max size (MiB) of logs to collect
...
```

Practice 20-2: Using Standard Linux Performance Monitoring Tools

Overview

In this practice, you use standard Linux system resource monitoring utilities to observe the following:

- CPU statistics
- Memory statistics
- Disk I/O statistics
- Network statistics

Assumptions

- You are the `root` user on **dom0**.
- The **host03** VM is running.
- The output shown in the tasks is only a sample. Your output will be different.

Tasks

1. From **dom0**, log on to **host03** from two terminal windows.
   a. Open two terminal windows on **dom0** and use the `su -` command to become the `root` user (the `root` user's password on **dom0** is `oracle`) in both windows.
      - You have two terminal windows open:
        – One window to run the Linux system resource monitoring utilities
        – One window to run a command-line utility to generate a system load

```
[dom0]$ su –
Password: oracle
[dom0]#
```

   b. Use the `ssh` command to log in to **host03**. The `root` password is `oracle`.
      - Log in to **host03** from both terminal windows.

```
[dom0]# ssh host03
root@host03's password: oracle
[root@host03 ~]#
```

2. Observe the CPU statistics.
   a. In one window, use the `top` command to display CPU usage and load averages.
      - The `top` command also monitors process statistics and memory usage.

```
# top
```

   b. In the second window, run the following command to generate a system load:

```
# dd if=/dev/zero of=/dev/null bs=1024
```

- A sample output of `top` is shown as follows:

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
top - 08:41:12 up 7 days, 22:15,  5 users,  load average: 0.61, 0.20, 0.10
Tasks: 173 total,   2 running, 171 sleeping,   0 stopped,   0 zombie
%Cpu(s): 19.5 us, 80.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   2050960 total,  1928140 used,   122820 free,   170360 buffers
KiB Swap:  3876860 total,        0 used,  3876860 free.   970624 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1540 root      20   0  107920    624    532 R 99.7  0.0   0:56.49 dd
 1509 root      20   0  123648   1616   1132 R  0.7  0.1   0:00.18 top
    1 root      20   0   53108   7064   3796 S  0.0  0.3   0:23.25 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.13 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:01.25 ksoftirqd/0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:+
    6 root      20   0       0      0      0 S  0.0  0.0   0:00.54 kworker/u:0
    7 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/u:+
    8 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
   10 root      20   0       0      0      0 S  0.0  0.0   0:02.79 rcu_sched
   11 root      rt   0       0      0      0 S  0.0  0.0   0:08.45 watchdog/0
   12 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 cpuset
   13 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 khelper
   14 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   15 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
   16 root      20   0       0      0      0 S  0.0  0.0   0:00.00 xenwatch
```

- In this example, CPU usage is high as indicated by the `0.0 id` (idle) statistic.
- The biggest consumer of the CPU is the `dd` process, which has a PID of `1540`.
- Load average is the average number of processes in the run queue or the number of processes waiting to run on the CPU.
  - The load average over the last 1 minute is `0.61`, over the last 5 minutes is `0.20`, and over the last 15 minutes is `0.10`.
  - A high load average is an indication that your system does not have sufficient CPU capacity.
- Press **q** to quit the `top` command.

c. Use the `mpstat -P ALL 1` command to display CPU statistics.

- The `1` argument provides statistics at one-second intervals.
- The `-P ALL` option provides statistics for each individual processor and globally for all processors.

```
[root@host03 ~]# mpstat -P ALL 1
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_        (1 CPU)

08:44:23 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:44:24 AM  all   19.00    0.00   81.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
08:44:24 AM    0   19.00    0.00   81.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

08:44:24 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:44:25 AM  all   20.00    0.00   80.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
08:44:25 AM    0   20.00    0.00   80.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

08:44:25 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:44:26 AM  all   21.00    0.00   79.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
08:44:26 AM    0   21.00    0.00   79.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

08:44:26 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:44:27 AM  all   20.00    0.00   80.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
08:44:27 AM    0   20.00    0.00   80.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

08:44:27 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:44:28 AM  all   21.78    0.00   78.22    0.00    0.00    0.00    0.00    0.00    0.00    0.00
08:44:28 AM    0   21.78    0.00   78.22    0.00    0.00    0.00    0.00    0.00    0.00    0.00

08:44:28 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
```

- In this example, CPU usage is high as indicated by the `0.00` (`%idle`) statistic.
- Press Ctrl + C after viewing a few intervals.

d. Use the `sar -u` command to display system-wide CPU usage. Only a partial output is shown as follows:

```
[root@host03 ~]# sar -u
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_ (
1 CPU)

12:00:01 AM     CPU     %user     %nice   %system   %iowait    %steal     %idle
12:10:01 AM     all      0.40      0.00      0.34      0.00      0.00     99.26
12:20:01 AM     all      0.23      0.00      0.11      0.00      0.00     99.66
12:30:01 AM     all      0.23      0.00      0.13      0.00      0.00     99.64
12:40:01 AM     all      0.31      0.00      0.21      0.00      0.00     99.48
12:50:01 AM     all      0.23      0.00      0.10      0.00      0.00     99.68
01:00:01 AM     all      0.24      0.00      0.11      0.00      0.00     99.64
01:10:02 AM     all      0.40      0.00      0.34      0.00      0.00     99.26
01:20:01 AM     all      0.23      0.00      0.09      0.00      0.00     99.67
01:30:01 AM     all      0.25      0.00      0.12      0.00      0.00     99.63
01:40:01 AM     all      0.31      0.00      0.20      0.00      0.00     99.49
01:50:01 AM     all      0.23      0.00      0.09      0.00      0.00     99.68
02:00:01 AM     all      0.23      0.00      0.09      0.00      0.00     99.68
02:10:01 AM     all      0.40      0.00      0.33      0.00      0.00     99.27
02:20:01 AM     all      0.23      0.00      0.10      0.00      0.00     99.67
02:30:01 AM     all      0.24      0.00      0.10      0.00      0.00     99.67
02:40:01 AM     all      0.31      0.00      0.20      0.00      0.00     99.50
02:50:01 AM     all      0.25      0.00      0.09      0.00      0.00     99.66
03:00:01 AM     all      0.24      0.00      0.11      0.00      0.00     99.65
03:10:01 AM     all      0.37      0.00      0.31      0.00      0.00     99.32
```

- In this example, CPU usage is low, with a few exceptions, as indicated by the `99.xx` (`%idle`) statistics.
- An "Average" line is not shown in this example.

e. Use the `sar -q` command to display run queue length and load averages. Only a partial output is shown as follows:

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
[root@host03 ~]# sar -q
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_
1 CPU)

12:00:01 AM   runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15   blocked
12:10:01 AM         3       296      0.00      0.01      0.05         0
12:20:01 AM         1       296      0.00      0.01      0.05         0
12:30:01 AM         3       299      0.00      0.01      0.05         0
12:40:01 AM         2       296      0.00      0.04      0.05         0
12:50:01 AM         2       297      0.00      0.01      0.05         0
01:00:01 AM         1       297      0.00      0.01      0.05         0
01:10:02 AM         3       296      0.00      0.01      0.05         0
01:20:01 AM         2       297      0.00      0.01      0.05         0
01:30:01 AM         3       299      0.00      0.01      0.05         0
01:40:01 AM         3       296      0.00      0.03      0.05         0
01:50:01 AM         2       297      0.00      0.01      0.05         0
02:00:01 AM         1       297      0.00      0.01      0.05         0
02:10:01 AM         2       296      0.00      0.01      0.05         0
02:20:01 AM         3       296      0.00      0.01      0.05         0
02:30:01 AM         2       299      0.00      0.01      0.05         0
02:40:01 AM         2       296      0.00      0.01      0.05         0
02:50:01 AM         4       297      0.00      0.01      0.05         0
03:00:01 AM         1       296      0.00      0.01      0.05         0
03:10:01 AM         2       297      0.00      0.01      0.05         0
```

- In this example, the CPU is not saturated as indicated by the low run queue length (`runq-sz`) and low load averages (last 1 minute: `ldavg-1`, last 5 minutes: `ldavg-5`, and last 15 minutes: `ldavg-15`).
- A run queue size greater than the number of CPUs on your system is usually indicative of a CPU bottleneck.
- An "Average" line is not shown in this example.

f. Use the `vmstat 1` command to display CPU statistics.

- This command is primarily used for monitoring virtual memory statistics.

```
[root@host03 ~]# vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0      0 124436 170520 970664    0    0     1     2   43   22  1  1 99  0  0
 1  0      0 124428 170520 970664    0    0     0     0 1052   21 19 81  0  0  0
 1  0      0 124428 170520 970664    0    0     0     0 1058   12 20 80  0  0  0
 1  0      0 124436 170520 970664    0    0     0     0 1060   27 21 79  0  0  0
 1  0      0 124436 170520 970664    0    0     0     0 1039   15 20 80  0  0  0
 1  0      0 124436 170520 970664    0    0     0     0 1049   25 19 81  0  0  0
 1  0      0 124436 170520 970664    0    0     0     0 1054   12 18 82  0  0  0
 1  0      0 124444 170520 970664    0    0     0     0 1040   19 20 80  0  0  0
 1  0      0 124444 170520 970664    0    0     0     0 1030   10 21 79  0  0  0
 1  0      0 124444 170520 970664    0    0     0     0 1037   26 19 81  0  0  0
 1  0      0 124444 170520 970664    0    0     0     0 1032   12 19 81  0  0  0
 1  0      0 124452 170520 970664    0    0     0     0 1041   19 19 81  0  0  0
 1  0      0 124452 170520 970664    0    0     0     0 1032   10 21 79  0  0  0
 1  0      0 124452 170520 970664    0    0     0     0 1043   17 19 81  0  0  0
 1  0      0 124452 170520 970664    0    0     0     0 1030   23 20 80  0  0  0
 1  0      0 124452 170520 970664    0    0     0     0 1041   19 21 79  0  0  0
 1  0      0 124452 170520 970664    0    0     0     0 1062   16 19 81  0  0  0
 1  0      0 124420 170520 970664    0    0     0     0 1048   17 19 81  0  0  0
 2  0      0 124228 170520 970664    0    0     0     0 1038   33 27 73  0  0  0
 1  0      0 124292 170520 970664    0    0     0     0 1051   60 20 80  0  0  0
 3  0      0 118236 170520 970664    0    0     0     0 1031  263 24 76  0  0  0
```

- The `1` argument provides statistics at one-second intervals.
- In this example, CPU usage is high as indicated by the low `id` (`%idle`) statistics.
- In this example, the high run queue length (`r`) statistics are also an indication of CPU saturation.
- Press Ctrl + C after viewing a few intervals.

g. Use the `iostat` command to display CPU usage.

- This command is primarily used for monitoring system I/O device loads.

```
[root@host03 ~]# iostat
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_(
1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.87    0.00    0.59    0.01    0.00   98.53

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
xvda              0.23         1.32         1.68     908131    1155340
scd0              0.00         0.04         0.00      29634          0
xvdb              0.00         0.01         0.00       5221         32
xvdd              0.00         0.01         0.00       5221         32

[root@host03 ~]# 
```

- In this example, the average `%idle` statistic for the CPU is 98.53%.

3. Observe the memory statistics.

a. Use the `top` command to display memory usage.

```
# top
```

A sample output of `top` is shown as follows:

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
top - 09:38:26 up 7 days, 23:12,  5 users,  load average: 1.03, 1.02, 0.96
Tasks: 172 total,   2 running, 170 sleeping,   0 stopped,   0 zombie
%Cpu(s): 18.9 us, 81.1 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   2050960 total,  1926632 used,   124328 free,   170540 buffers
KiB Swap:  3876860 total,        0 used,  3876860 free.   970668 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1540 root      20   0  107920    624    532 R 99.7  0.0  57:48.64 dd
 2550 root      20   0  123648   1620   1132 R  0.3  0.1   0:00.04 top
    1 root      20   0   53108   7064   3796 S  0.0  0.3   0:23.35 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.13 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:01.25 ksoftirqd/0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S  0.0  0.0   0:00.54 kworker/u:0
    7 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/u:0H
    8 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
   10 root      20   0       0      0      0 S  0.0  0.0   0:02.80 rcu_sched
   11 root      rt   0       0      0      0 S  0.0  0.0   0:08.48 watchdog/0
   12 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 cpuset
   13 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 khelper
   14 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   15 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
   16 root      20   0       0      0      0 S  0.0  0.0   0:00.00 xenwatch
```

- The upper section of `top` displays memory statistics:
  - The `KiB Mem:` line reflects how much physical memory your system has, how much is used, and how much is free.
  - The `KiB Swap:` line reflects how much swap memory your system has, how much is used, and how much is free.
  - When a computer runs out of physical memory and starts using swap space, its performance deteriorates dramatically.
  - If you run out of swap, you will most likely crash your programs or the OS.
- The lower section of `top` displays a list of processes sorted by CPU usage, with the top consumer of CPU listed first.

b. Sort by memory usage, press F or f to change the sort field, and then use the Up/Down arrow key to highlight either of the following:

   - `%MEM` = Memory usage (RES)
   - `RES` = Resident size (kb)

- Press the **s** key to set the selected sort field.
- Press **q** or **Esc** to exit the field management window and return to the `top` display.

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
top - 09:53:38 up 7 days, 23:27,  5 users,  load average: 1.00, 1.03, 1.01
Tasks: 172 total,   2 running, 170 sleeping,   0 stopped,   0 zombie
%Cpu(s): 19.3 us, 80.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   2050960 total,  1927160 used,   123800 free,   170556 buffers
KiB Swap:  3876860 total,        0 used,  3876860 free.   970676 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1751 oracle    20   0 1623828 362420  36980 S  0.0 17.7  74:36.75 gnome-shell
  713 root      20   0  191696  34652   7804 S  0.0  1.7   4:28.97 Xorg
 1746 oracle    20   0  551340  25976  12400 S  0.0  1.3   0:33.53 goa-daemon
 1674 oracle    20   0  924216  23572  14416 S  0.0  1.1   0:17.38 gnome-settin+
 1830 oracle    20   0 1047936  22816  15816 S  0.0  1.1   0:01.63 nautilus
 2297 oracle    20   0  619720  19332  12256 S  0.0  0.9   0:30.27 gnome-termin+
  532 root      20   0  549984  16108   5604 S  0.0  0.8   1:25.31 tuned
20856 polkitd   20   0  522724  16012   4876 S  0.0  0.8   0:21.75 polkitd
 1265 root      20   0  478596  14016   8804 S  0.0  0.7   0:00.74 libvirtd
 1953 oracle    20   0  898744  13960   8404 S  0.0  0.7   0:00.32 evolution-ca+
 1881 oracle    20   0  448152  13852   9392 S  0.0  0.7   0:00.10 abrt-applet
 1476 oracle    20   0  850872  12444   8244 S  0.0  0.6   0:02.26 gnome-session
 1855 oracle    20   0  424768  11336   4564 S  0.0  0.6   0:00.19 tracker-store
 1798 oracle    20   0  325428  11288   5536 S  0.0  0.6   0:03.02 mission-cont+
 1806 oracle    20   0  668216  10360   7624 S  0.0  0.5   0:00.09 evolution-so+
 1852 oracle    39  19  603232  10260   7220 S  0.0  0.5   0:00.12 tracker-mine+
 1890 oracle    20   0  870352   9760   7336 S  0.0  0.5   0:00.05 evolution-ad+
```

- Press **q** to quit the `top` command.

c. Use the `vmstat` command to display memory statistics.

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
[root@host03 ~]# vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0      0 124600 170732 970704    0    0     1     2   48   22  1  1 98  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1055   22 23 77  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1063   14 19 81  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1038   24 20 80  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1033   10 18 82  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1040   21 20 80  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1037   16 23 77  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1058   23 21 79  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1049   12 21 79  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1041   17 20 80  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1055   14 21 79  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1037   23 18 82  0  0  0
 1  0      0 124636 170732 970704    0    0     0     0 1048   12 20 80  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1069   26 20 80  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1035   12 19 81  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1074   21 21 79  0  0  0
 1  0      0 124652 170732 970704    0    0     0     0 1058   10 21 79  0  0  0
 1  0      0 124668 170732 970704    0    0     0     0 1073   19 19 81  0  0  0
 1  0      0 124668 170732 970704    0    0     0     0 1068   16 19 81  0  0  0
 1  0      0 124668 170732 970704    0    0     0     0 1037   10 20 80  0  0  0
^C
```

- The `1` argument provides statistics at one-second intervals.
- The important memory statistics are:
  - `swpd` – The amount of virtual memory used
  - `free` – The amount of idle memory
  - `si` – The amount of memory swapped in from disk (per second)
  - `so` – The amount of memory swapped out to disk (per second)
- In this example, the system has sufficient free memory and is not swapping.
- Press Ctrl + C after viewing a few intervals.

d. Use the `free` command to display memory statistics.

```
[root@host03 ~]# free -m
             total       used       free     shared    buffers     cached
Mem:          2002       1879        122         28        166        947
-/+ buffers/cache:        765       1237
Swap:         3785          0       3785
[root@host03 ~]#
```

- This example uses the `-m` option to display amounts in megabytes.
- This command displays the total amount of free and used physical and swap memory in your system.

e. Use the `sar -r` command to display memory usage statistics. Only a partial output is shown as follows:

```
[root@host03 ~]# sar -r
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_        (1 CPU)

12:00:01 AM kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit   %commit  kbactive   kbinact   kbdirty
12:10:01 AM     71080   1979880     96.53    169820   1037016   2237292     37.74    978628    783764         4
12:20:01 AM     70956   1980004     96.54    169860   1037024   2237292     37.74    978708    783752         4
12:30:01 AM     69140   1981820     96.63    169900   1037028   2279976     38.46    980236    783768         0
12:40:01 AM     70952   1980008     96.54    169944   1037036   2237992     37.75    978788    783792         0
12:50:01 AM     70696   1980264     96.55    169956   1037040   2299992     38.80    978880    783780         0
01:00:01 AM     70728   1980232     96.55    169968   1037048   2300116     38.80    978916    783780         0
01:10:02 AM     70668   1980292     96.55    170020   1037060   2237416     37.74    979000    783772         0
01:20:01 AM     70700   1980260     96.55    170040   1037064   2300116     38.80    979120    783732         0
01:30:01 AM     68760   1982200     96.65    170060   1037068   2280104     38.46    980612    783728         4
01:40:01 AM     70456   1980504     96.56    170096   1037080   2237420     37.74    979220    783724         4
01:50:01 AM     70412   1980548     96.57    170112   1037084   2238128     37.76    979296    783712         0
02:00:01 AM     70348   1980612     96.57    170144   1037148   2300148     38.80    979380    783724         0
02:10:01 AM     70332   1980628     96.57    170192   1037104   2237420     37.74    979388    783720         0
02:20:01 AM     70332   1980628     96.57    170228   1037104   2237420     37.74    979456    783700         0
02:30:01 AM     68424   1982536     96.66    170264   1037116   2280108     38.46    981036    783720         0
02:40:01 AM     70080   1980880     96.58    170304   1037120   2237416     37.74    979608    783728         0
02:50:01 AM     69960   1981000     96.59    170336   1037116   2300148     38.80    979680    783740         4
03:00:01 AM     69836   1981124     96.59    170360   1037128   2238120     37.76    980072    783760         4
03:10:01 AM     69448   1981512     96.61    170404   1037172   2237928     37.75    980028    783756         0
03:20:01 AM     69192   1981768     96.63    170424   1037176   2238640     37.76    980564    783712         0
03:30:01 AM     67512   1983448     96.71    170444   1037184   2280612     38.47    981704    783708         0
03:40:01 AM     68856   1982104     96.64    170608   1037816   2237448     37.74    981152    783404         0
```

- Observe the `%memused` value, which is the percentage of used memory.
- An "Average" line is not shown in this example.

f. Use the `sar -B` command to display memory paging statistics. Only a partial output is shown as follows:

```
[root@host03 ~]# sar -B
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_        (1 CPU)

12:00:01 AM  pgpgin/s pgpgout/s   fault/s  majflt/s  pgfree/s pgscank/s pgscand/s pgsteal/s    %vmeff
12:10:01 AM      0.00      1.38    406.19      0.00    111.26      0.00      0.00      0.00      0.00
12:20:01 AM      0.00      0.69    109.51      0.00     37.20      0.00      0.00      0.00      0.00
12:30:01 AM      0.00      0.47    114.64      0.00     38.99      0.00      0.00      0.00      0.00
12:40:01 AM      0.00      0.61    247.84      0.00     74.72      0.00      0.00      0.00      0.00
12:50:01 AM      0.00      0.21    109.61      0.00     37.87      0.00      0.00      0.00      0.00
01:00:01 AM      0.00      0.29    133.51      0.00     43.21      0.00      0.00      0.00      0.00
01:10:02 AM      0.00      1.32    418.44      0.00    115.87      0.00      0.00      0.00      0.00
01:20:01 AM      0.00      0.60    108.37      0.00     38.03      0.00      0.00      0.00      0.00
01:30:01 AM      0.00      0.45    134.78      0.00     44.17      0.00      0.00      0.00      0.00
01:40:01 AM      0.00      0.70    240.48      0.00     74.48      0.00      0.00      0.00      0.00
01:50:01 AM      0.00      0.23    107.17      0.00     37.92      0.00      0.00      0.00      0.00
02:00:01 AM      0.00      0.28    103.71      0.00     37.09      0.00      0.00      0.00      0.00
02:10:01 AM      0.00      1.23    413.13      0.00    116.42      0.00      0.00      0.00      0.00
02:20:01 AM      0.00      0.55    104.39      0.00     37.47      0.00      0.00      0.00      0.00
02:30:01 AM      0.00      0.52    108.29      0.00     38.15      0.00      0.00      0.00      0.00
02:40:01 AM      0.00      0.56    237.54      0.00     74.56      0.00      0.00      0.00      0.00
02:50:01 AM      0.00      0.25    103.81      0.00     37.67      0.00      0.00      0.00      0.00
03:00:01 AM      0.00      0.22    130.22      0.00     43.45      0.00      0.00      0.00      0.00
03:10:01 AM      0.00      1.25    366.53      0.00    108.44      0.00      0.00      0.00      0.00
03:20:01 AM      0.00      0.62    101.17      0.00     37.35      0.00      0.00      0.00      0.00
03:30:01 AM      0.00      0.53    104.90      0.00     38.07      0.00      0.00      0.00      0.00
03:40:01 AM      1.07     16.25    401.07      0.06    136.57      0.00      0.00      0.00      0.00
```

- Observe `pgscank/s`, which is the number of pages scanned by the `kswapd` daemon per second, and `pgscand/s`, which is the number of pages scanned directly per second.
- An "Average" line is not shown in this example.

g. Use the `sar -W` command to display memory swapping statistics. Only a partial output is shown as follows:

```
[root@host03 ~]# sar -W
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_    (
1 CPU)

12:00:01 AM  pswpin/s pswpout/s
12:10:01 AM      0.00      0.00
12:20:01 AM      0.00      0.00
12:30:01 AM      0.00      0.00
12:40:01 AM      0.00      0.00
12:50:01 AM      0.00      0.00
01:00:01 AM      0.00      0.00
01:10:02 AM      0.00      0.00
01:20:01 AM      0.00      0.00
01:30:01 AM      0.00      0.00
01:40:01 AM      0.00      0.00
01:50:01 AM      0.00      0.00
02:00:01 AM      0.00      0.00
02:10:01 AM      0.00      0.00
02:20:01 AM      0.00      0.00
02:30:01 AM      0.00      0.00
02:40:01 AM      0.00      0.00
02:50:01 AM      0.00      0.00
03:00:01 AM      0.00      0.00
03:10:01 AM      0.00      0.00
03:20:01 AM      0.00      0.00
03:30:01 AM      0.00      0.00
```

- The `pswpin/s` value is the number of swap pages the system swapped in per second, and `pswpout/s` is the number of swap pages the system swapped out per second.
- An "Average" line is not shown in this example.

h. Use the `cat` command to view the contents of `/proc/meminfo`. Only a partial output is shown as follows:

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
[root@host03 ~]# cat /proc/meminfo
MemTotal:        1786200 kB
MemFree:          332792 kB
Buffers:           92388 kB
Cached:           882680 kB
SwapCached:            0 kB
Active:           599772 kB
Inactive:         694680 kB
Active(anon):     322740 kB
Inactive(anon):     8708 kB
Active(file):     277032 kB
Inactive(file):   685972 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:       3876860 kB
SwapFree:        3876860 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:        319396 kB
Mapped:            86848 kB
Shmem:             12072 kB
Slab:             109564 kB
SReclaimable:      88524 kB
SUnreclaim:        21040 kB
```

4. Observe the disk I/O statistics.

a. Use the `iostat -xz` command to display I/O statistics for devices.

```
root@host03:~
File  Edit  View  Terminal  Tabs  Help
[root@host03 ~]# iostat -xz
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_        (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.98    0.00    1.04    0.01    0.00   97.96

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
xvda              0.01     0.14    0.08    0.15     1.31     1.68    26.53     0.00   12.15    7.14   14.77   0.76   0.02
scd0              0.00     0.00    0.00    0.00     0.04     0.00    94.53     0.00    2.64    2.64    0.00   2.21   0.00
xvdb              0.00     0.00    0.00    0.00     0.01     0.00     8.27     0.00    2.14    2.08   13.86   1.67   0.00
xvdd              0.00     0.00    0.00    0.00     0.01     0.00     8.27     0.00    2.29    2.24   11.57   1.83   0.00

[root@host03 ~]#
```

- Observe the `%util` value, which is the percentage of CPU time during which I/O requests were issued to the device.
- Device saturation occurs when this value is close to `100%`.
- Observe the `avgqu-sz` value, which is the average queue length of the requests that were issued to the device.
- If average queue length is greater than `1`, it is an indication of device I/O saturation.

b. Use the `sar -d` command to display I/O statistics for devices. Only a partial output is shown as follows:

```
root@host03:~
File Edit View Terminal Tabs Help
[root@host03 ~]# sar -d
Linux 3.8.13-44.1.1.el7uek.x86_64 (host03.example.com)  11/18/2014      _x86_64_        (1 CPU)

12:00:01 AM       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
12:10:01 AM  dev202-0      0.18      0.00      2.76     15.62      0.00      0.65      0.19      0.00
12:10:01 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:10:01 AM dev202-16      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:10:01 AM dev202-48      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:20:01 AM  dev202-0      0.10      0.00      1.39     13.64      0.00      0.28      0.11      0.00
12:20:01 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:20:01 AM dev202-16      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:20:01 AM dev202-48      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:30:01 AM  dev202-0      0.07      0.00      0.93     13.66      0.00      0.22      0.12      0.00
12:30:01 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:30:01 AM dev202-16      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:30:01 AM dev202-48      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:40:01 AM  dev202-0      0.09      0.00      1.22     13.24      0.00      0.18      0.11      0.00
12:40:01 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:40:01 AM dev202-16      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:40:01 AM dev202-48      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00

12:40:01 AM       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
12:50:01 AM  dev202-0      0.04      0.00      0.43     11.64      0.00      0.05      0.05      0.00
12:50:01 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:50:01 AM dev202-16      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:50:01 AM dev202-48      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
01:00:01 AM  dev202-0      0.05      0.00      0.57     12.74      0.00      0.26      0.11      0.00
01:00:01 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

- This command also provides `%util` and `avgqu-sz` statistics.
- An "Average" line is not shown in this example.

5. Observe network statistics.

a. Use the `ip -s link` command to observe network statistics.

```
root@host03:~
File Edit View Terminal Tabs Help
[root@host03 ~]# ip -s link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX: bytes  packets  errors  dropped overrun mcast
    32072      523      0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    32072      523      0       0       0       0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    14226688   361449   0       5765    0       0
    TX: bytes  packets  errors  dropped carrier collsns
    4325189    33737    0       0       0       0
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    13174904   346708   0       5765    0       0
    TX: bytes  packets  errors  dropped carrier collsns
    8667       62       0       0       0       0
[root@host03 ~]#
```

- The number of bytes received and bytes transmitted on each interface is provided by `RX: bytes` and `TX: bytes` values.
- This command also provides the number of packets transmitted and received, the number of errors, dropped packets, overruns, and collisions.

- Frames are dropped and the overrun counter is incremented when the capacity of the interface is exceeded.

b. Use the `netstat -s` command to observe network statistics. Only a partial output is shown as follows:

- If the `ip` command shows an excessive amount of errors, more information can be found by examining the `netstat -s` output.

```
[root@host03 ~]# netstat -s
Ip:
    10460 total packets received
    0 forwarded
    0 incoming packets discarded
    10326 incoming packets delivered
    29682 requests sent out
    125 dropped because of missing route
Icmp:
    234 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 216
        echo requests: 18
    349 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 331
        echo replies: 18
IcmpMsg:
        InType3: 216
        InType8: 18
        OutType0: 18
        OutType3: 331
Tcp:
    15 active connections openings
```

- This command displays summary statistics for each protocol.
- Observe the number of segments retransmitted as an indicator of network interface saturation.
- Many performance problems associated with the network involve retransmission of TCP packets.

c. Use the `netstat -i` command to observe a table listing of network interfaces.

```
[root@host03 ~]# netstat -i
Kernel Interface table
Iface       MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500   361726      0   5765 0        33900      0      0      0 BMRU
eth1       1500   346819      0   5765 0           62      0      0      0 BMRU
lo        65536      523      0      0 0          523      0      0      0 LRU
[root@host03 ~]#
```

- Observe the `RX-ERR` and `TX-ERR` values for any receive and transmit errors.

6. Use the System Monitor GUI to display system resource usage.

This application requires that you access the GNOME desktop.

a. Use the `exit` command to close the `ssh` connection from **dom0** to **host03**.

- Do not close the connection in the window that is running the "`dd`" command from step 2b.
- Close the connection in the window in which you ran `top`, `vmstat`, `iostat`, and `netstat` commands.

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

b. From **dom0**, run the `xm list -l host03 | grep location` command to determine the VNC port number for **host03**.

```
[dom0]# xm list –l host03 | grep location
                (location 0.0.0.0:5904)
                (location 3)
```

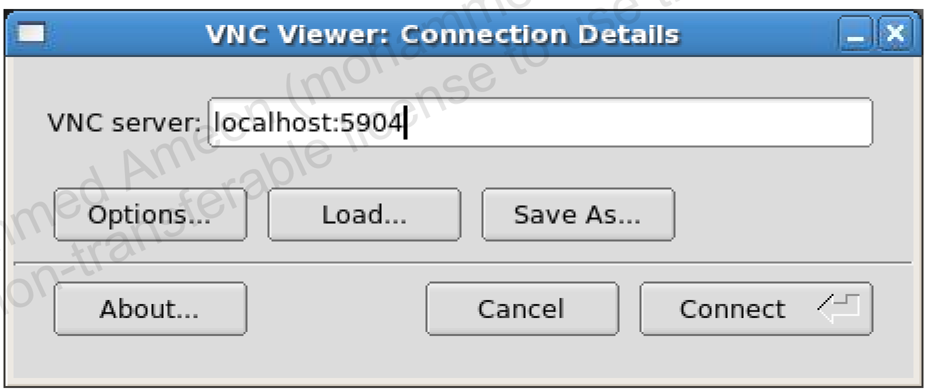
- In this example, the VNC port number is `5904`. This might not be true in your case.

c. Run the `vncviewer&` command:

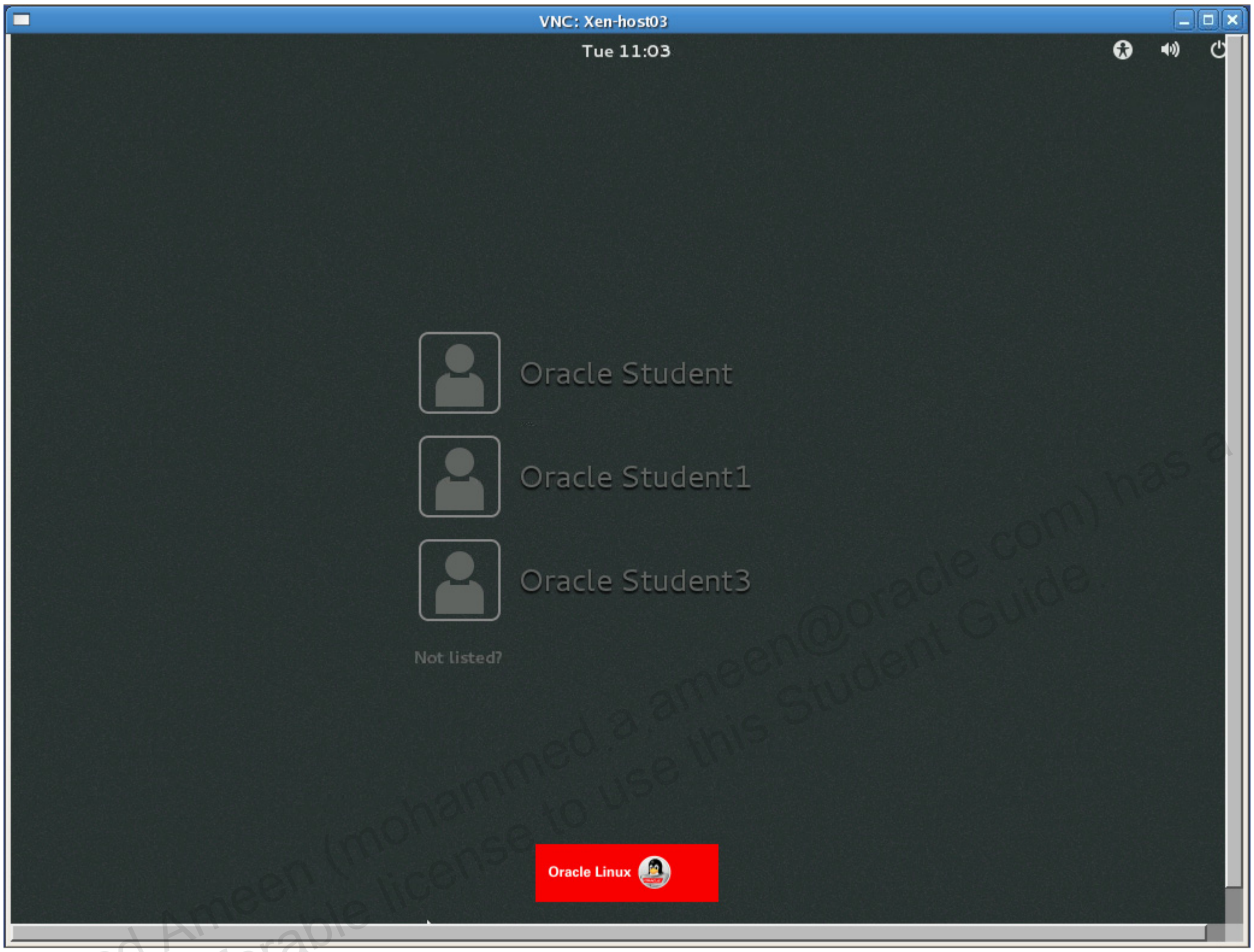
```
[dom0]# vncviewer&
```

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

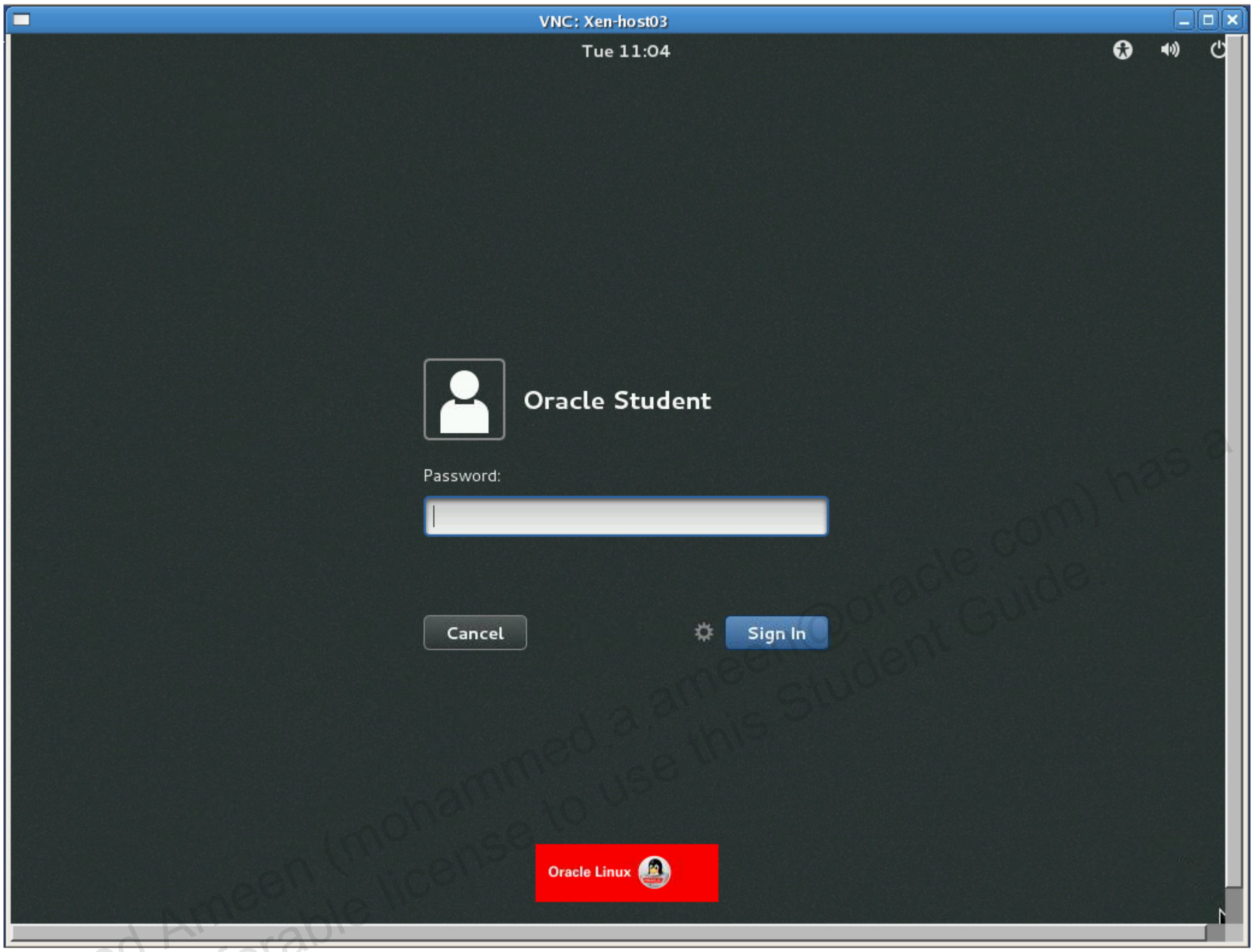
d. Enter the `localhost:<port_number>` command, substituting the VNC port number displayed from the previous `xm list` command. For example, if the port number is `5904`, enter `localhost:5904` as shown and click **Connect**.



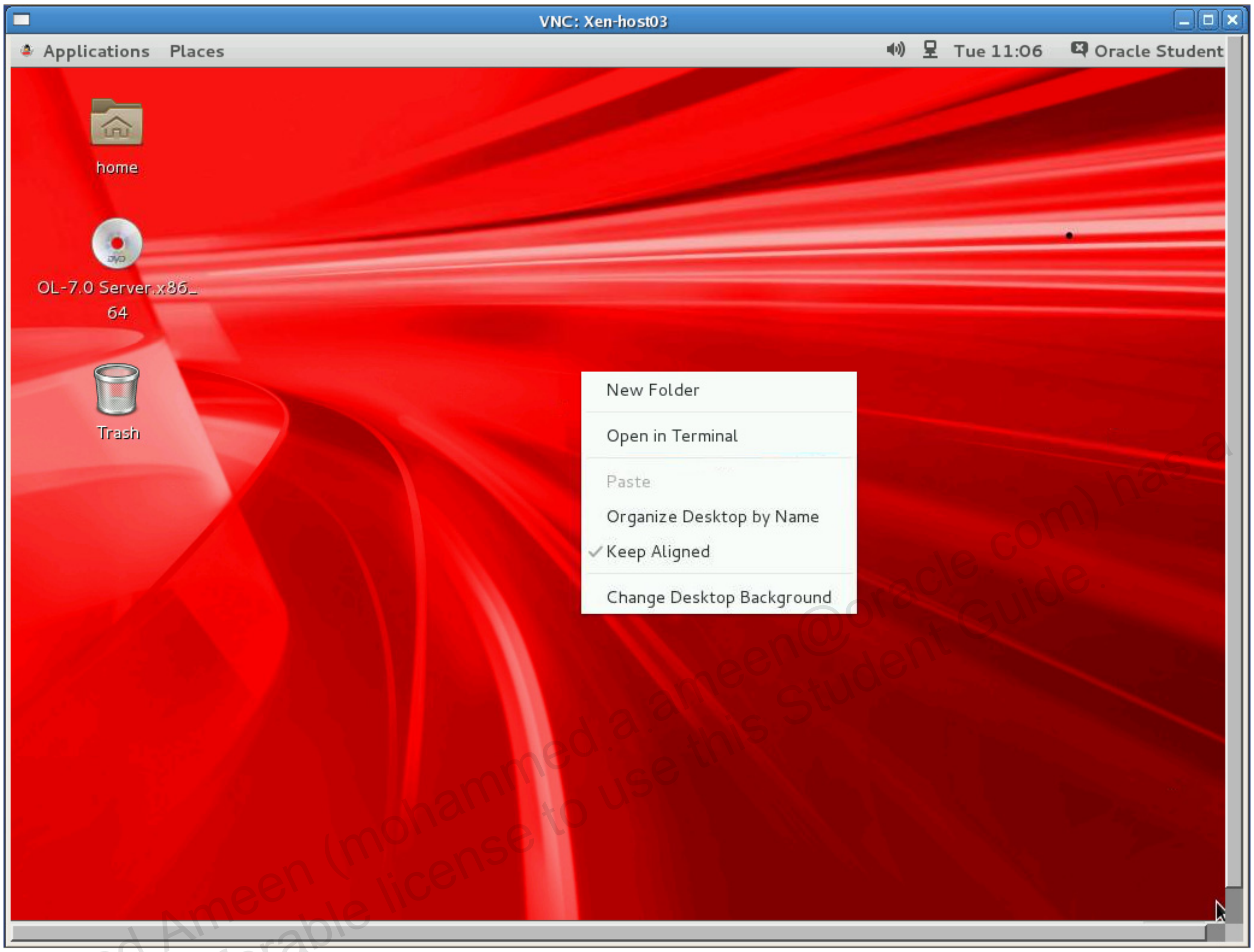
e. The GNOME desktop login window appears as shown in the following screenshot. Select **Oracle Student** from the list of users.



f. You are prompted for the **Oracle Student** password as shown in the following screenshot. Enter the password `oracle` and click **Sign In**.



g. The GNOME desktop appears. Right-click the desktop to display the pop-up menu as shown in the following screenshot. Click **Open in Terminal** to open a terminal window.



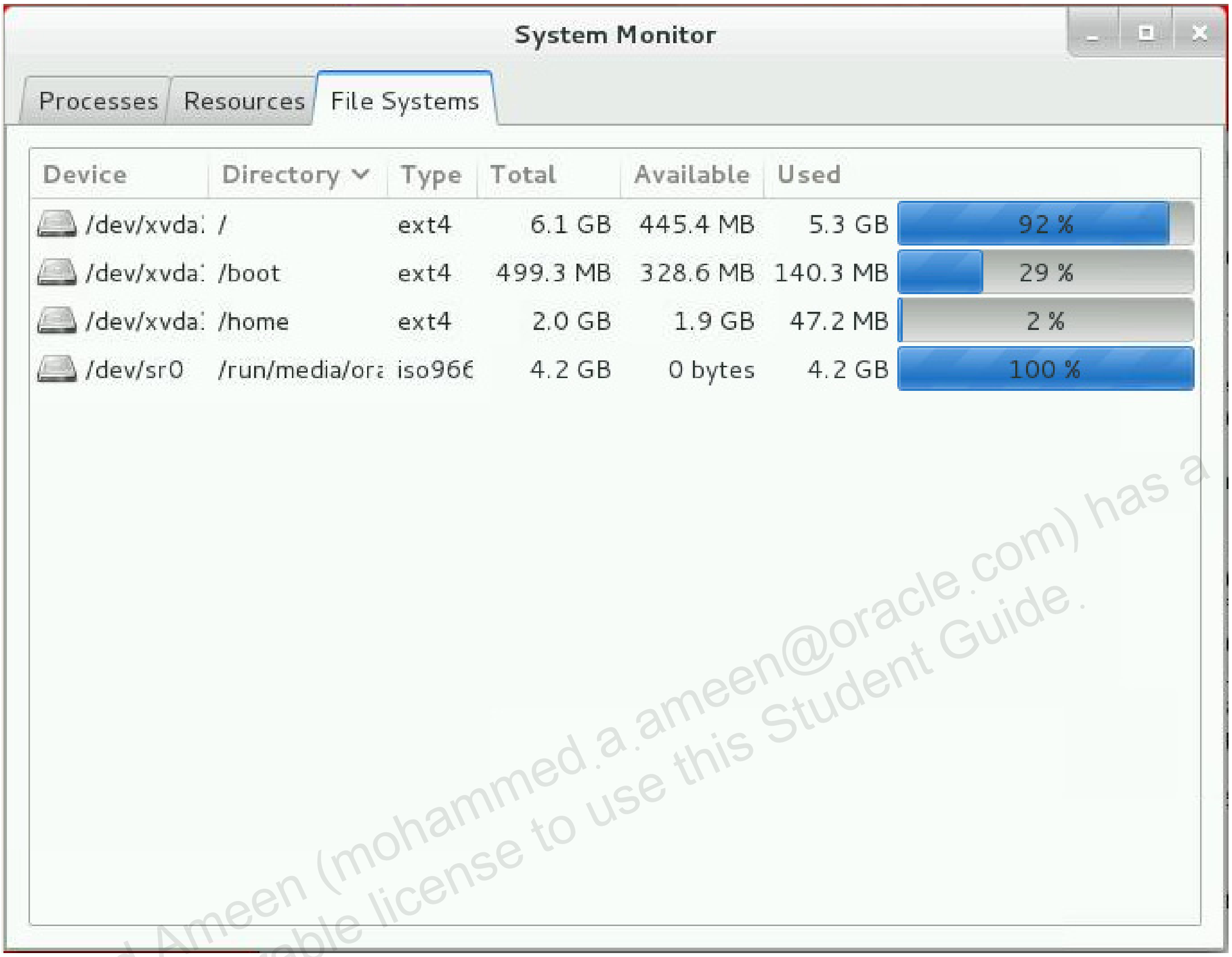
h. From the terminal window, use the `su -` command to become the `root` user. The `root` user's password is `oracle`.

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

i. From the terminal window, enter the `gnome-system-monitor` command to display the System Monitor GUI.

```
# gnome-system-monitor
```

- The System Monitor is displayed as shown in the following screenshot:



j. Click the other tabs and observe the information displayed on each tab:
   1) Processes tab
      - Click any column header to sort on that column.
      - Note that there is an "**End Process**" button you can use to kill a selected process.
   2) Resources tab
      - Note that there are graphs showing CPU, Memory, and Network history in real-time.

k. Click the **X** character in the top right of the window to exit the System Monitor GUI.
   - Do not exit the GNOME desktop.

Practice 20-3: Installing and Using OSWatcher

Overview

In this practice, you install and run the OSWatcher Black Box (OSWbb) product and view the collected data. Note that MOS Doc ID 580513.1 describes “How To Start OSWatcher Black Box Every System Boot.”

Assumptions

- You are the `root` user on the **host03** VM.
- You are the `root` user on **dom0**.

Tasks

1. Use the `sftp` command to copy the `oswbb732.tar` file from **dom0** to **host03**.
   a. From **host03**, use the `systemctl` command to verify that `sshd` is running.
      - Start the service using the “`systemctl start sshd`” command if necessary.

```
[host03]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service...
   Active: active (running) since ...
...
```

      - In this example, the service is running.

   b. From **dom0**, use the `cd` command to change to the `/OVS/seed_pool/oswbb` directory.

```
[dom0]# cd /OVS/seed_pool/oswbb
```

   c. Use the `ls` command to view the `oswbb732.tar` file.

```
[dom0]# ls
oswbb732.tar
```

   d. Use the `sftp` command to connect to **host03**.
      - Password is `oracle`.

```
[dom0]# sftp host03
Connecting to host03...
root@host03’s password: oracle
sftp>
```

   e. Use the `put` command to copy the `oswbb732.tar` file to **host03**.

```
sftp> put oswbb732.tar
Uploading oswbb732.tar to /root/oswbb732.tar
oswbb732.tar ...
```

   f. Use the `quit` command to exit `sftp`.

```
sftp> quit
```

Perform all remaining steps in this practice from **host03**.

2. From **host03**, install, start, and stop OSWbb.

   a. Use the `cd` command to change to the `root` user's home directory.

```
# cd
```

   b. Use the `tar` command to extract the `oswbb732.tar` file.

```
# tar xvf oswbb732.tar
oswbb/
...
```

   c. Use the `cd` command to change to the `oswbb` directory, and then use `ls` to view the contents of the directory.

```
# cd oswbb
# ls
analysis/    ifconfigusub.sh   oswib.sh   tarupfiles.sh
call_du.sh   iosub.sh          oswnet.sh  tar_up_partial_arch...
call_sar.sh  locks/            oswrds.sh  tmp/
...
```

   - Note the `startOSWbb.sh` file, which is the script used to start OSWbb.

   d. Use the `startOSWbb.sh` command to start OSWbb.

```
# ./startOSWbb.sh
Info...You did not enter a value for snapshotInterval.
Info...Using default value = 30
Info...You did not enter a value for archiveInterval.
Info...Using default value = 48
Setting the archive log directory to /root/oswbb/archive

Testing for discover of OS Utilities
VMSTAT found on your system
IOSTAT found on your system
MPSTAT found on your system
NETSTAT found on your system
TOP found on your system

Testing for discover of OS CPU COUNT
...
CPU COUNT = 1
Discovery completed.

Starting OSWatcher Black Box v7.3.2 on ...
With SnapshotInterval = 30
With ArchiveInterval = 48
...
Starting Data Collection...
```

```
oswbb heartbeat:...
oswbb heartbeat:...
oswbb heartbeat:...
...
```

- OSWbb started successfully, the discovery process completed, and data collection begins.
- The default intervals (`snapshotInterval = 30` and `archiveInterval = 48`) are used.

e. After a few data collection events (`oswbb heartbeat`) have completed, use the `stopOSWbb.sh` command to stop OSWbb.

```
# ./stopOSWbb.sh
```

3. View the data collection directories.

   Use the `cd` command to change to the `archive` directory, and then use `ls` to view the contents of the directory.

```
# cd archive
# ls
oswifconfig/  oswmeminfo/  oswnetstat/  oswps/  oswtop/
oswiostat/    oswmpstat/   oswprvtnet/  oswslabinfo/  oswvmstat/
```

- The `archive` directory is created when OSWbb is started for the first time.
- The directory contains 10 subdirectories, one for each data collector.

4. View the `oswiostat` directory.

a. Use the `cd` command to change to the `oswiostat` directory, and then use `ls` to view the contents of the directory.

```
# cd oswiostat
# ls
host03.example.com_iostat...
```

b. Use the `less` command to view the file.

```
# less host03...
Linux OSW v7.3.2
zzz ***...
...
```

- Note that this file contains the output of the `iostat -x` command.
- The `iostat` command ran at 30-second intervals (the value of `snapshotInterval`).
- Each interval begins with `zzz ***` characters followed by a time stamp.

5. View the `oswmpstat` directory.

a. Use the `cd` command to change to the `oswmpstat` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswmpstat
# pwd
/root/oswbb/archive/oswmpstat
```

```
# ls
host03.example.com_mpstat...
```

b. Use the `less` command to view the file.

```
# less host03...
Linux OSW v7.3.2
zzz ***...
...
```

- Note that this file contains the output of the `mpstat` command.
- The `mpstat` command ran at 30-second intervals (the value of `snapshotInterval`).
- Each interval begins with `zzz ***` characters followed by a time stamp.

6. View the `oswprvtnet` directory.

   Use the `cd` command to change to the `oswprvtnet` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswprvtnet
# pwd
/root/oswbb/archive/oswprvtnet
# ls
```

- Note that this directory is empty.
- This directory contains the status of RAC private networks.
- You need to manually create the `private.net` file to run `traceroute` commands.

7. View the `oswslabinfo` directory.

a. Use the `cd` command to change to the `oswslabinfo` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswslabinfo
# pwd
/root/oswbb/archive/oswslabinfo
# ls
host03.example.com_slabinfo...
```

b. Use the `less` command to view the file.

```
# less host03...
zzz ***...
slabinfo - version: 2.1
...
```

- Note that this file contains the contents of the `/proc/slabinfo` file.
- The `/proc/slabinfo` file is read at 30-second intervals (the value of `snapshotInterval`).
- Each interval begins with `zzz ***` characters followed by a time stamp.

8. View the `oswvmstat` directory.
   a. Use the `cd` command to change to the `oswvmstat` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswvmstat
# pwd
/root/oswbb/archive/oswvmstat
# ls
host03.example.com_vmstat...
```

   b. Use the `less` command to view the file.

```
# less host03...
Linux OSW v7.3.2 ...
SNAP_INTERVAL 30
CPU_COUNT 1
OSWBB_ARCHIVE_DEST /root/oswbb/archive
zzz ***...
...
```

   - Note that this file contains the output of the `vmstat` command.
   - The `vmstat` command ran at 30-second intervals (the value of `snapshotInterval`).
   - Each interval begins with `zzz ***` characters followed by a time stamp.

9. View the `oswmeminfo` directory.
   a. Use the `cd` command to change to the `oswmeminfo` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswmeminfo
# pwd
/root/oswbb/archive/oswmeminfo
# ls
host03.example.com_meminfo...
```

   b. Use the `less` command to view the file.

```
# less host03...
zzz ***...
MemTotal: ...
MemFree: ...
...
```

   - Note that this file contains the contents of the `/proc/meminfo` file.
   - The `/proc/meminfo` file is read at 30-second intervals (the value of `snapshotInterval`).
   - Each interval begins with `zzz ***` characters followed by a time stamp.

10. View the `oswnetstat` directory.
   a. Use the `cd` command to change to the `oswnetstat` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswnetstat
# pwd
/root/oswbb/archive/oswnetstat
# ls
host03.example.com_netstat...
```

b. Use the `less` command to view the file.

```
# less host03...
Linux OSW v7.3.2
zzz ***...
...
```

- Note that this file contains the output of the `netstat` command.
- The `netstat` command ran at 30-second intervals (the value of `snapshotInterval`).
- Each interval begins with `zzz ***` characters followed by a time stamp.

11. View the `oswps` directory.
    a. Use the `cd` command to change to the `oswps` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswps
# pwd
/root/oswbb/archive/oswps
# ls
host03.example.com_ps...
```

b. Use the `less` command to view the file.

```
# less host03...
Linux OSW v7.3.2
zzz ***...
...
```

- Note that this file contains the output of the `ps` command.
- The `ps` command ran at 30-second intervals (the value of `snapshotInterval`).
- Each interval begins with `zzz ***` characters followed by a time stamp.

12. View the `oswtop` directory.
    a. Use the `cd` command to change to the `oswtop` directory, and then use `ls` to view the contents of the directory.

```
# cd ../oswtop
# pwd
/root/oswbb/archive/oswtop
# ls
host03.example.com_top...
```

b. Use the `less` command to view the file.

```
# less host03...
Linux OSW v7.3.2
```

```
zzz ***...
...
```

- Note that this file contains the output of the top command.
- The top command ran at 30-second intervals (the value of snapshotInterval).
- Each interval begins with zzz *** characters followed by a time stamp.

13. View the oswifconfig directory.
    a. Use the cd command to change to the oswifconfig directory, and then use ls to view the contents of the directory.

```
# cd ../oswifconfig
# pwd
/root/oswbb/archive/oswifconfig
# ls
host03.example.com_top...
```

    b. Use the less command to view the file.

```
# less host03...
Linux OSW v7.3.2
zzz ***...
...
```

- Note that this file contains the output of the ifconfig command.
- The top command ran at 30-second intervals (the value of snapshotInterval).
- Each interval begins with zzz *** characters followed by a time stamp.

Practice 20-4: Using OSWatcher Analyzer

Overview

In this practice, you perform the following:

- Start OSWatcher Analyzer (OSWbba) on **host03**.
- View CPU and Memory Graphs from OSWbba.
- Use OSWbba to analyze the data.
- View the analysis report.

Assumptions

- You are the `root` user on **host03**.
- You completed Practice 20-3: Installing and Using OSWatcher.

Tasks

1. Start OSWatcher Analyzer (OSWbba) on **host03**.
   a. Use the `java -version` command to display the Java version number.

```
# java –version
java version “1.7.0_51”
...
```

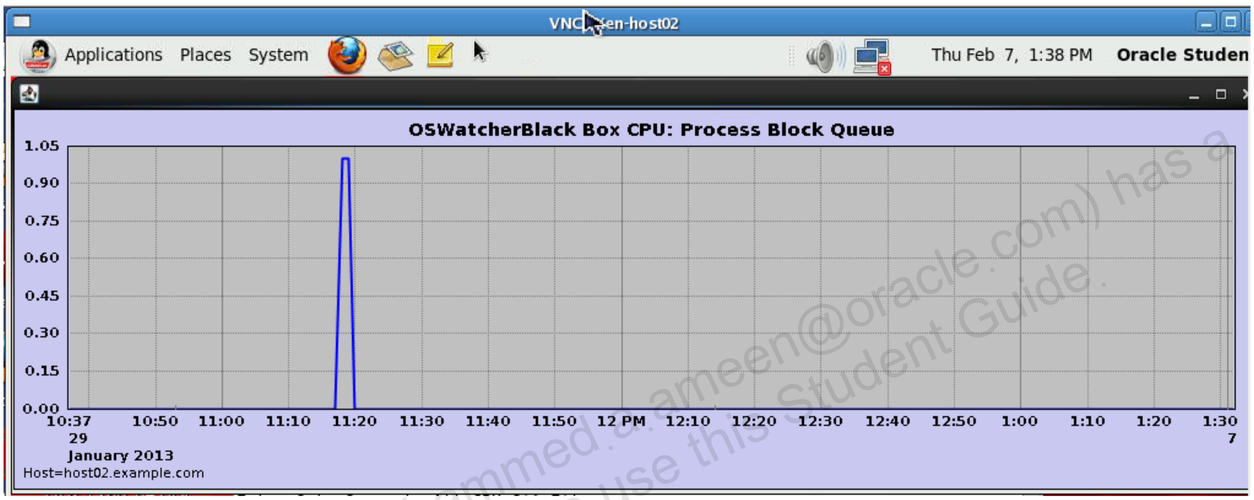
   - In this example, version `1.7.0_51` is installed.
   - The minimum version is `1.4.2`.

   b. Use the `java –jar` command to display the OSWbba menu. First, ensure that you are in the directory where OSWbba is installed (`~/oswbb`).

```
# cd ~/oswbb
# java –jar oswbba.jar –i ~/oswbb/archive
...
Enter 1 to Display CPU Process Queue Graphs
Enter 2 to Display CPU Utilization Graphs
Enter 3 to Display CPU Other Graphs
Enter 4 to Display Memory Graphs
Enter 5 to Display Disk IO Graphs

Enter 6 to Generate All CPU Gif Files
Enter 7 to Generate All Memory Gif Files
Enter 8 to Generate All Disk Gif Files

Enter L to Specify Alternate Location of Gif Directory
Enter T to Alter Graph Time Scale Only (Does not change ...)
Enter D to Return to Default Time scale
Enter R to Remove Currently Displayed Graphs

Enter A to Analyze Data
Enter S to Analyze Subset of Data(Changes analysis dataset ...
```

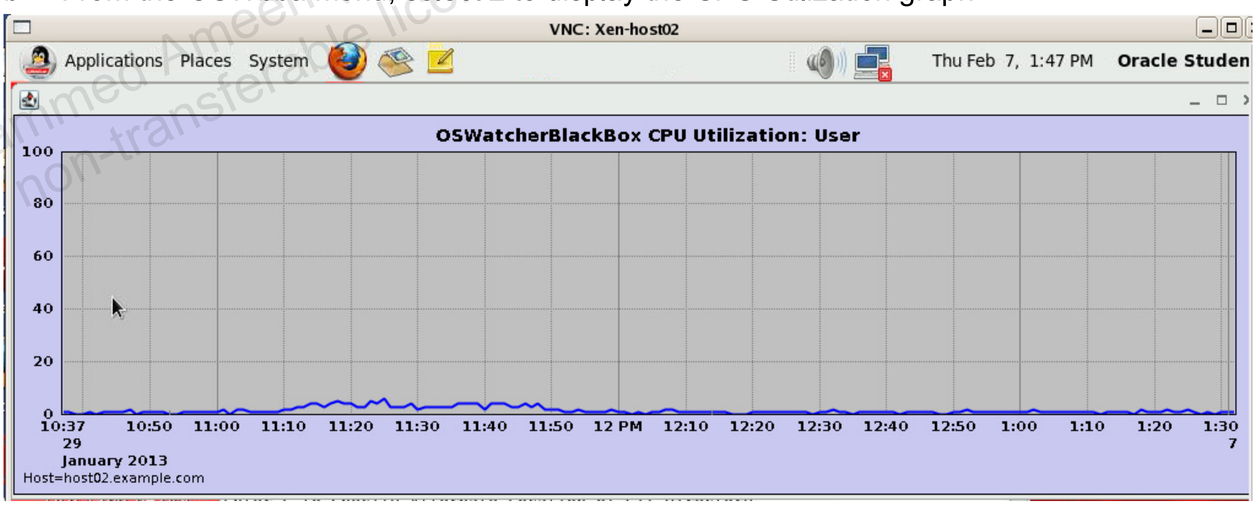
```
Enter P to Generate A Profile
Enter X to Export Parsed Data to File
Enter Q to Quit Program

Please Select an Option:
```

2. View CPU and Memory Graphs. A sample output is shown as follows. Your graphs will be different.
   a. From the OSWbba menu, select `1` to display the CPU Process Queue graph.



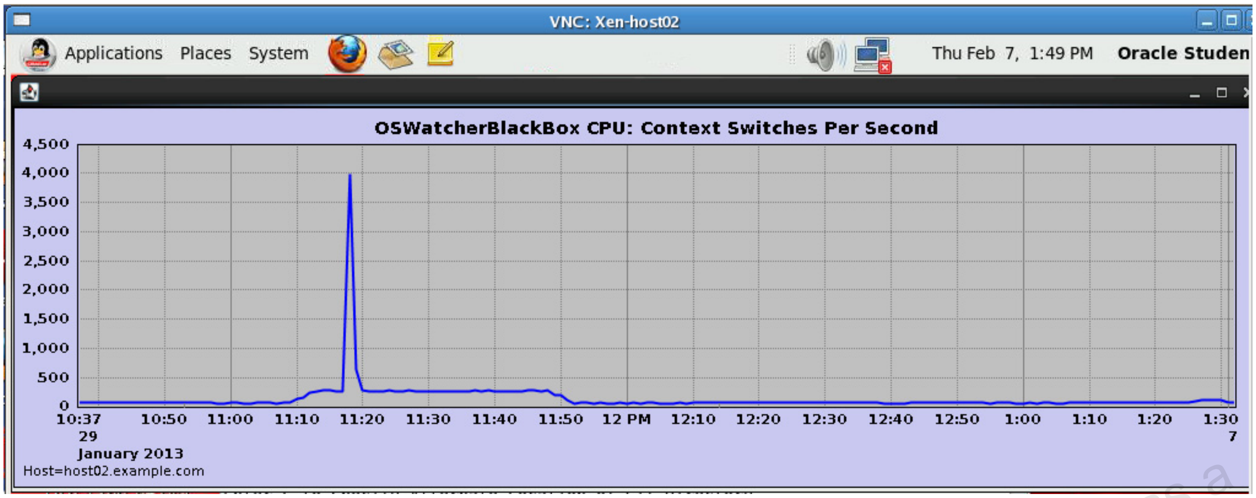
   - After viewing the graph, select R from the menu to remove the graph.

   b. From the OSWbba menu, select 2 to display the CPU Utilization graph.



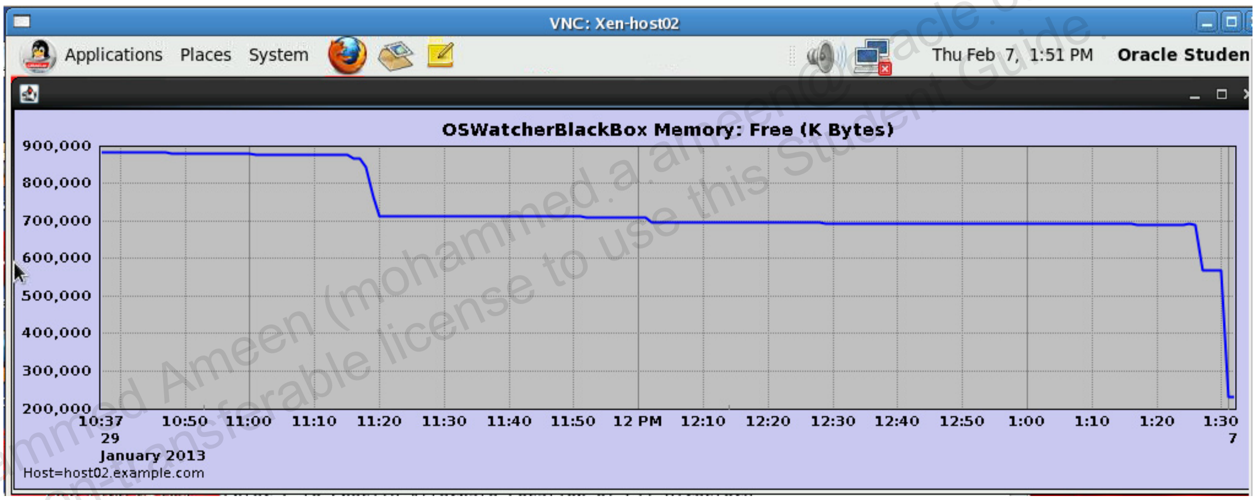
   - After viewing the graph, select R from the menu to remove the graph.

c. From the OSWbba menu, select 3 to display the CPU Other graph.



- After viewing the graph, select R from the menu to remove the graph.

d. From the OSWbba menu, select 4 to display the Memory Graph.



- After viewing the graph, select R from the menu to remove the graph.

3. Run the OSWbba analyzer and view the report.

a. From the OSWbba menu, select A to analyze the collected data and produce a report. Provide a file name or press Enter to accept the default name.

```
...
Enter A to Analyze Data
...
Please Select an Option:a
Enter a fully qualified analysis file name or enter <CR> to
accept default name: ENTER
A new analysis file analysis/host03...txt has been created.
```

- This message "A new analysis file analysis/host03...txt has been created" appears and the menu is then redisplayed.

- You could alternatively run the analyzer from the command line by using the following command (do not run this command, this is information only):

```
# java -jar oswbba.jar -i ~/oswbb/archive -A
```

b. Exit the OSWbba menu by selecting the Q option to quit.

c. From the **host03** command line, use the `ls` command to display the analysis file in the `~/oswbb/analysis` directory.

```
# ls ~/oswbb/analysis
host03.example.com...txt
```

- Your analysis file name is different from the sample file name shown.

d. Use the `less` command to view the analysis file.

- Use your analysis file name observed in the previous step.

```
# less ~/oswbb/analysis/host03*
...
```

- Note the file has system information at the top followed by sections such as:
  - Section 1: System Status
  - Section 2: System Slowdown
    - Section 2.1: System Slowdown RCA Process Level Ordered By Impact
  - Section 3: System General Findings
  - Section 4: CPU Detailed Findings
    - Section 4.1: CPU Run Queue:
    - Section 4.2: CPU Utilization: Percent Busy
    - Section 4.3: CPU Utilization: Percent Sys
  - Section 5: Memory Detailed Findings
    - Section 5.1: Memory: Process Swap Queue
    - Section 5.2: Memory: Scan Rate
    - Section 5.3: Memory: Page In:
    - Section 5.4: Memory: Page Tables (Linux only):
    - Section 5.5: Top 5 Memory Consuming Processes Beginning
    - Section 5.6: Top 5 Memory Consuming Processes Ending
  - Section 6: Disk Detailed Findings
    - Section 6.1: Disk Percent Utilization Findings
    - Section 6.2: Disk Service Times Findings
    - Section 6.3: Disk Wait Queue Times Findings
    - Section 6.4: Disk Throughput Findings
    - Section 6.5: Disk Reads Per Second
    - Section 6.6: Disk Writes Per Second
    - Section 6.7: Disk Percent CPU Waiting on I/O
  - Section 7: Network Detailed Findings
    - Section 7.1: Network Data Link Findings
    - Section 7.2: Network IP Findings

  - Section 7.3: Network UDP Findings
  - Section 7.4: Network TCP Findings
- Section 8: Process Detailed Findings
  - Section 8.1: PS Process Summary Ordered By Time
  - Section 8.2: PS for Processes With Status = D or T Ordered By Time
  - Section 8.3: PS for (Processes with CPU > 0) When System Idle CPU < 30% Ordered By Time
  - Section 8.4: Top VSZ Processes Increasing Memory Per Snapshot
  - Section 8.5: Top RSS Processes Increasing Memory Per Snapshot

e. After viewing the analysis file, press Q to exit the `less` command.

4. Terminate the `dd` command started in Practice 20-2 (2b).

   In the second terminal window on **host03** in which you initiated the following `dd` command, press Ctrl + C to terminate the command.

```
# dd if=/dev/zero of=/dev/null bs=1024
CTRL-C
```

Practices for Lesson 21: System Logging

Chapter 21

Practices for Lesson 21: Overview

Practices Overview

In these practices, you configure system logging, use `rsyslog` templates to format log messages, install and run `logwatch`., view the `journald` journal, and configure persistent `journald` storage.

Practice 21-1: Configuring System Logging

Overview

In this practice, you view the system logging configuration file, modify the file, and observe the impact of the modifications. You also configure log file rotation.

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Explore the main configuration file for system logging, `/etc/rsyslog.conf`.
   a. Use the `less` command to view the system logging configuration file.

```
# less /etc/rsyslog.conf
...
#### MODULES ####
...
$ModLoad imuxsock   # provides support for local system...
$ModLoad imjournal   # provides access to the systemd journal
...
#### GLOBAL DIRECTIVES ####

# Where to place auxiliary files
$WorkDirectory /var/lib/rsyslog

# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
...
# Include all config files in /etc/rsyslog.d/
$IncludeConfig /etc/rsyslog.d/*.conf

# Turn off message reception via local log socket;
# local messages are retrieved through imjournal now.
$OmitLocalLogging on

# File to store the position in the journal
$IMJournalStateFile imjournal.state

#### RULES ####
...
# Log all kernel messages to the console.
# Logging much else clutters up the screen
#kern.*                                    /dev/console

# Log anything (except mail) of level info or higher.
```

```
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none   /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                 /var/log/secure

# Log all the mail messages in one place.
mail.*                                    -/var/log/maillog

# Log cron stuff
cron.*                                     /var/log/cron
...
```

b. Use the up-arrow and down-arrow keys to view the various sections of the file.
   - The **MODULES** section uses the `$ModLoad` command to load the modules.
   - The **GLOBAL DIRECTIVES** section specifies configuration options.
   - The **RULES** section defines a selector (facility.priority) and an action.

c. Press the **q** key to exit the "`less`" command.

2. Change the action for `cron` logging.

a. Use the `ls` command to list the `cron*` files in the `/var/log` directory.
   - Note that the rotated log files have a date stamp.

```
# ls /var/log/cron*
/var/log/cron
/var/log/cron-2014...
...
```

b. Use the `vi` editor to modify the system logging configuration file. Change the action for `cron` logging to log to a different log file: `/var/log/cron_new`.

```
# vi /etc/rsyslog.conf
...
# Log cron stuff
cron.*            /var/log/cron              (old entry)
cron.*            /var/log/cron_new          (new entry)
```

c. Use the `systemctl` command to restart the `rsyslog` service.

```
# systemctl restart rsyslog
```

d. Use the `crontab -e` command to create a `cron` job that runs the `ls` command every minute.

```
# crontab -e
* * * * * ls
```

e. Use the `ls` command to list the `cron*` files in the `/var/log` directory.

```
# ls /var/log/cron*
/var/log/cron
/var/log/cron-2014...
```

```
...
/var/log/cron_new
```

- Note the new log file, cron_new.

f. Use the tail command to view the last two lines in the cron log file.

```
# tail -2 /var/log/cron
Dec 13 11:22:51 host03 CROND[979]: (root) CMD (/usr/lib64/...)
Dec 13 11:23:01 host03 CROND[1094]: (root) CMD (/usr/lib64/...)
```

- This is sample output only.

g. Use the head command to view the first entries in the cron_new log file.

```
# head /var/log/cron_new
Dec 13 11:24:01 host03 crontab[1198]: (root) BEGIN EDIT (root)
Dec 13 11:25:01 host03 crontab[1198]: (root) REPLACE )root)
...
```

- Note from the date_time stamps that the new log entries are being written to cron_new.

h. Use the vi editor to modify the system logging configuration file. Change the action for cron logging back to the original log file.

```
# vi /etc/rsyslog.conf
...
# Log cron stuff
cron.*            /var/log/cron_new            (old entry)
cron.*            /var/log/cron                (new entry)
```

i. Use the systemctl command to restart the rsyslog service.

```
# systemctl restart rsyslog
```

j. Use the tail command to ensure that cron is now logging to /var/log/cron.

- This is sample output only.

```
# tail -1 /var/log/cron
...
Dec 13 11:44:01 host03 CROND[1094]: ...

# tail -1 /var/log/cron_new
...
Dec 13 11:43:01 host03 CROND[8376]: ...
```

- Note that the entry in cron has a later time stamp than the entry in cron_new.
- You might need to wait a minute for the cron job to run.

k. Use the rm command to delete the cron_new log file.

```
# rm /var/log/cron_new
rm: remove regular file 'cron_new'? y
```

l. Use crontab -r to remove the crontab.

```
# crontab -r
```

3. Configure `rsyslog` to log debug messages.

   a. Use the `vi` editor to modify the system logging configuration file. Add an entry at the bottom of the file to log all debug messages to `/var/log/debug`.

```
# vi /etc/rsyslog.conf
...
*.debug                          /var/log/debug
```

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

```
# systemctl restart rsyslog
```

   c. Use the `logger` command to generate an **informational** log message.

   - Logger is an interface to the `syslog(3)` system log module.
   - Logger makes entries in the system log.

```
# logger -p info "This is an info-priority message"
```

   d. Use the `tail` command to view the log files.

```
# tail /var/log/messages
...
Dec 13 13:36:12 host03 oracle: This is an info-priority message
# tail /var/log/debug
...
Dec 13 13:36:12 host03 oracle: This is an info-priority message
```

   - Note that the message was written to both log files.

   e. Use the `logger` command to generate a **debug-level** log message.

```
# logger -p debug "This is a debug-priority message"
```

   f. Use the `tail` command to view the log files.

```
# tail /var/log/messages
...
Dec 13 13:36:12 host03 oracle: This is an info-priority message
# tail /var/log/debug
...
Dec 13 13:42:16 host03 oracle: This is a debug-priority message
```

   - Note that the debug-level message was written only to `/var/log/debug`.

   g. Use the `vi` editor to modify `/etc/rsyslog.conf` and remove the entry at the bottom of the file to log all debug messages.

```
# vi /etc/rsyslog.conf
...
*.debug                          /var/log/debug (delete this entry)
```

   h. Use the `systemctl` command to restart the `rsyslog` service.

```
# systemctl restart rsyslog
```

4. Configure log file rotation.
   a. Use the `ls` command to view the contents of the `/var/log` directory.
      - This is sample output only.

```
# ls /var/log/messages*
/var/log/messages
/var/log/messages-2014...
...

# ls /var/log/maillog*
/var/log/maillog
/var/log/maillog-2014...
...

# ls /var/log/cron*
/var/log/cron
/var/log/cron-2014...
...
```

      - Note that some files in `/var/log` have numbers at the end of the file name.
      - These numbers represent a rotated log with the time stamp added to the log file name.
      - You might not have log files with a time stamp appended to the file name. It depends on how long your system has been running.
   b. Use the `vi` editor to modify the `/etc/logrotate.conf` configuration file. Change the frequency of the default log file rotation from **weekly** to **daily**.

```
# vi /etc/logrotate.conf
...
# rotate log files weekly
weekly                                              (old entry)
daily                                               (new entry)
```

      - Your log files now rotate daily after making this change.

Practice 21-2: Using `rsyslog` Templates

Overview

In this practice, you use `rsyslog` templates to format `rsyslog` output.

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Define and use a template.
   a. Use the `vi` editor to modify `/etc/rsyslog.conf` and define a template.
      - Add the template definition line shown at the bottom of the file.
      - This entry creates a template named `class`.
      - Do not exit the `vi` editor.

```
# vi /etc/rsyslog.conf
...
$template class,"Message: %msg%\n"
```

   b. Continue editing `/etc/rsyslog.conf` and create a log file that uses the template.
      - Add the new line (shown in **bold**) after the entry that defined the template.
      - This entry writes all messages to the `/var/log/class.log` file and formats the entries by using the `class` template.
      - Exit the `vi` editor and save the file after adding the new line.

```
...
$template class,"Message: %msg%\n"
*.*    /var/log/class.log;class
```

   c. After saving the changes to `/etc/rsyslog.conf`, use the `systemctl` command to restart the `rsyslog` service.

```
# systemctl restart rsyslog
```

   d. Use the `cat` command to view the `/var/log/class.log` file.

```
# cat /var/log/class.log
Message: ...
...
```

      - Note that all entries are preceded by the text "`Message:`" followed by the actual message, as defined in the `class` template.
2. Modify the `class` template.
   a. Use the `vi` editor to edit `/etc/rsyslog.conf` and modify the `class` template.
      - Change the template definition as shown.

```
# vi /etc/rsyslog.conf
...
$template class,"Message: %msg%\n"        (old entry)
```

```
$template class,"Time: %timestamp%, Facility: %syslogfacility-text%, Priority: %syslogpriority-text%, Hostname: %hostname%, Message: %msg%\n"                          (new entry)
```

b. After saving the changes to /etc/rsyslog.conf, use the systemctl command to restart the rsyslog service.

```
# systemctl restart rsyslog
```

c. Use the cat command to view the /var/log/class.log file.

```
# cat /var/log/class.log
Message: ...
...
Time: ...
...
```

- Note that the newest entries now include the Time, Facility, Priority, Hostname, and Message properties, as defined in the class template.

3. Restore default configuration.

a. Use the vi editor to modify /etc/rsyslog.conf and:

- Delete the template definition entry
- Delete the rule that uses the template entry

```
# vi /etc/rsyslog.conf
...
$template class, "Time: %timestamp%, Facility: %syslogfacility-text%, Priority: %syslogpriority-text%, Hostname: %hostname%, Message: %msg%\n"
*.*    /var/log/class.log;class
```

b. After saving the changes to /etc/rsyslog.conf, use the systemctl command to restart the rsyslog service.

```
# systemctl restart rsyslog
```

c. Use the rm command to remove the /var/log/class.log file.

```
# rm /var/log/class.log
rm: remove regular file '/var/log/class.log'? y
```

Practice 21-3: Using `logwatch`

Overview

In this practice, you install the `logwatch` package, view the main configuration file (the `cron` file), and run the `logwatch` utility from the command line.

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Install the `logwatch` package.
   a. Use the `rpm` command to determine whether the `logwatch` package is already installed.

```
# rpm -q logwatch
```

   - In this example, the `logwatch` package is not installed.

   b. Use the `yum` command to install the `logwatch` package.
   - Answer **y** when prompted.

```
# yum install logwatch
...
Transaction Summary
===================================================
Install     1 Package (+3 Dependent packages)
Total download size: 1.6 M
Installed size: 14 M
Is this ok [y/d/N]: y
...
Complete!
```

2. View `logwatch` files.
   a. Use the `find` command to list all `logwatch` files.
   - Only partial output is displayed.

```
# find / -name "*logwatch*"
...
/usr/sbin/logwatch
/usr/share/doc/logwatch-7.4.0
...
/usr/share/logwatch/default.conf/logwatch.conf
...
/etc/logwatch/conf/logwatch.conf
/etc/cron.daily/0logwatch
...
```

   b. Use the `less` command to view the `logwatch` configuration file.

```
# less /usr/share/logwatch/default.conf/logwatch.conf
...
```

- Note various configurable items such as the following:
  - LogDir
  - TmpDir
  - MailTo
  - MailFrom
  - Range
  - Detail
  - Service

c. Use the `less` command to view the `logwatch cron` file.

```
# less /etc/cron.daily/0logwatch
...
```

3. Run `logwatch` from the command line.

a. Run the `logwatch --help` command to view the `logwatch` help.

```
# logwatch --help
Usage: /usr/sbin/logwatch [--detail <level] [--logfile <name>]
   [--output <output_type>] ...
...
```

b. Run `logwatch` and view the output.

- Sample output is displayed.

```
# logwatch

################# logwatch 7.4.0 (03/01/11) #################
        Proccessing Initiated: Fri Nov...
        Date Range Processed: yesterday
...
        Logfiles for Host: host03.exmaple.com
##############################################################
...
----------------------Kernel Begin --------------------------

 WARNING: ...
...
```

Practice 21-4: Using `journald`

Overview

In this practice, you use the `journalctl` command to query the `systemd` journal, view `journald` metadata, and enable persistent `journald` storage.

Assumptions

You are the `root` user on the **host03** VM.

Tasks

1. Use the `journalctl` command to query the `systemd` journal.
   a. Run `journalctl` with no options or arguments.

```
# journalctl
-- Logs begin at ..., end at ...
<date_time> host03.example.com systemd-journal[65]: ...
<date_time> host03.example.com kernel: Initializing cgroup ...
<date_time> host03.example.com kernel: Linux version 3.8.13-...
<date_time> host03.example.com kernel: Command line: BOOT_IM...
<date_time> host03.example.com kernel: e820: BIOS-provided ...
<date_time> host03.example.com kernel: BIOS-e820 [mem 0x...
...
<date_time> host03.example.com kernel: Cannot get hvm parame...
...
```

   - Note that all log data is displayed, including rotated logs.
   - Note that the beginning of the boot process is indicated with a special entry.
   - Note that entries with error priority and higher are in red.
   - Note that entries with notice and warning priority are in bold font.

   b. Run the `journalctl -h` command to display usage and query options.

```
# journalctl -h
journalctl [OPTIONS...] [MATCHES...]
Query the journal.
Flags:
     --system         Show only the system journal
     --user           Show only the user journal for the curr...
     --since=DATE     Start showing entries on or newer than ...
     --until=DATE     Stop showing entries on or older than ...
...
  -k --dmesg          Show kernel message log from the curren...
  -u --unit=UNIT      Show data only from the specified unit
...
  -p -priority=RANGE Show only messages within the specified ...
...
```

c. Run the `journalctl -r` command to display the newest log entries first.

```
# journalctl -r
-- Logs begin at ..., end at ...
<date_time> host03.example.com CROND[27325]: (root) CMD (/usr...
<date_time> host03.example.com systemd[1]: Started session ...
<date_time> host03.example.com fprint[27149]: ** Message: No ...
<date_time> host03.example.com dbus[557]: [system] Successful...
<date_time> host03.example.com dbus-daemon[557]: dbus[557] ...
...
```

d. Run the `journalctl -n 3` command to display the three newest log entries.

- Using the `-n <number>` option displays a specific `<number>` of the most recent log entries.

```
# journalctl -n 3
-- Logs begin at ..., end at ...
<date_time> host03.example.com systemd[1]: Started session ...
<date_time> host03.example.com systemd[1]: Started session ...
<date_time> host03.example.com CROND[27452]: (root) CMD (/usr...
```

e. Run the `journalctl -p crit` command to display log entries with a priority of `crit`.

- You can use the `-p` option to display log entries of any priority. Valid priorities are `debug`, `info`, `notice`, `warning`, `err`, `crit`, `alert`, and `emerg`.

```
# journalctl -p crit
-- Logs begin at ..., end at ...
<date_time> host03.example.com smartd[512]: Problem creating ...
<date_time> host03.example.com smartd[512]: In the system's ...
<date_time> host03.example.com firewalld[506]: 2014-11-10 ...
...
```

f. Run the `journalctl -u crond` command to display log entries associated with the `crond` unit.

- You can use the `-u` option to display log entries for any `systemd` unit.

```
# journalctl -u crond
-- Logs begin at ..., end at ...
<date_time> host03.example.com systemd[1]: Starting Command ...
<date_time> host03.example.com systemd[1]: Started Command ...
<date_time> host03.example.com crond[578]: (CRON) INFO (RAN...
<date_time> host03.example.com crond[578]: (CRON) INFO (runn...
```

2. View `journald` metadata.

- `Journald` adds structured metadata to the messages that assists in troubleshooting.

a. Run the `journalctl` command and then press the Tab key twice to display the metadata fields.

```
# journalctl <TAB <TAB>
```

```
_AUDIT_LOGINUID=           _MONOTONIC_TIMESTAMP=
_AUDIT_SESSION=            _PID
_BOOT_ID                   _PRIORITY=
_CMDLINE=                  _REALTIME_TIMESTAMP=
CODE_FILE=                 _SELINUX_CONTEXT=
...
_MACHINE_ID=               _UDEV_SYSNAME=
MESSAGE=                   _UID=
MESSAGE_ID=
```

b. Run the `journalctl -o verbose` command to display log entries in `verbose` format.

- `Verbose` format shows the metadata fields and values for all journal entries.
- You can use the `-o` option to display log entries in any supported format. Supported format options are `short`, `short-iso`, `short-precise`, `short-monotonic`, `verbose`, `export`, `json`, `json-pretty`, `json-see`, and `cat`.

```
# journalctl -o verbose
-- Logs begin at ..., end at ...
<date_time>
    PRIORITY=6
    _TRANSPORT=driver
    MESSAGE=Runtime journal is using 8.0M (max 100.1M, leav...
    MESSAGE_ID=...
    _PID=65
    _UID=0
    _GID=0
    _COMM=systemd-journal
    _EXE=/usr/lib/systemd/systemd-journald
...
```

c. Run the `journalctl -F _UID` command to display unique values for the `_UID` metadata field.

- Sample output is shown.

```
# journalctl -F _UID
172
42
999
997
1000
89
81
0
994
70
```

d. Run the `journalctl _UID=<value>` command to show only log entries that match the condition.

- This example uses `172` as `<value>`. Substitute a valid `UID` value output from the previous command.

```
# journalctl _UID=172
-- Logs begin at ..., end at ...
-- Logs begin at ..., end at ...
<date_time> host03.example.com rtkit-daemon[554]: Succesfu ...
<date_time> host03.example.com rtkit-daemon[554]: Succesfu ...
<date_time> host03.example.com rtkit-daemon[554]: Running.
<date_time> host03.example.com rtkit-daemon[554]: Watchdog ...
<date_time> host03.example.com rtkit-daemon[554]: Canary ...
...
```

3. Enable persistent `journald` storage.

a. Use the `mount` command, pipe the output to `grep`, and search for the string "`run`".

```
# mount | grep run
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
...
```

- Note that the file system type mounted on `/run` is `tmpfs`.

b. Use the `ls -l` command to view the `/run/log/journal/` directory.

```
# ls -l /run/log/journal
drwxr-sr-x ... <date_time> ...
```

- By default, journal data is stored in this directory.
- Note the date_time stamp on this directory.

c. Use the `mkdir -p` command to make the `/var/log/journal/` directory.

```
# mkdir -p /var/log/journal
```

d. Use the `systemctl` command to restart the `systemd-journald` service.

```
# systemctl restart systemd-journald
```

e. Use the `ls -l` command to view the `/var/log/journal/` directory.

```
# ls -l /var/log/journal
drwxr-sr-x ... <date_time> ...
```

- Journal data is now stored in this directory.
- Note the date_time stamp on this directory is more recent than the date_time stamp on `/run/log/journal/`.

Practices for Lesson 22: Troubleshooting

Chapter 22

Practices for Lesson 22: Overview

Practices Overview

In these practices, you perform a variety of troubleshooting exercises and fix some common problems. Specific problems you encounter include the following:

- System boots into single-user mode by default.
- Status utilities are not producing expected output.
- A `cron` job fails to run.
- A user cannot log in.
- File system does not mount.
- Logical volume space is exhausted.
- You cannot `ping` remote hosts.
- There are NFS permission problems.
- You cannot log in to remote hosts using `ssh`.
- Log file is not getting updated.

Each practice:

- Tells you what problem the activity is designed to simulate
- Tells you how to set up the problem
- Gives you some hints or things to check
- Has an associated “Solution” that provides steps to diagnose and resolve the problem

Each solution:

- Includes only the steps to resolve the problem
- Does not include steps to set up the problem

Appendix:

- An appendix exists, which provides the source code for the executables that cause the problems that you need to troubleshoot and fix.

Practice 22-1: Transferring Utilities from dom0

Overview

In this practice, you transfer files hosted on **dom0** to **host01**. In subsequent practices, you are instructed to run one of these executables that causes the problem that you need to troubleshoot and fix.

Assumptions

The host01 VM is running.

Tasks

1. Log on to **host01**.

   From **dom0**, use the `ssh` command to connect to **host01**.

   - The `root` password is `oracle`.

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

2. Use the `sftp` command to transfer files from **dom0** to **host01**.

   a. From **host01**, use the `sftp` command to connect to **dom0**.

   - Answer "`yes`" when prompted, "Are you sure…"
   - Provide `root` password of `oracle` when prompted.

   ```
   [host01]# sftp dom0
   The authenticity of host 'dom0 (192.0.2.1)' can't be
   established. RSA key fingerprint is ...
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added '192.0.2.1' (RSA) to the list of
   known hosts.
   root@dom0's password: oracle
   sftp>
   ```

   b. From the `sftp` prompt, use the `cd` command to change to the `/OVS/seed_pool/ts_scripts` directory on **dom0**.

   - Use the `ls` command to list the directory contents.

   ```
   sftp> cd /OVS/seed_pool/ts_scripts
   sftp> ls
   two.x
   three.x
   four.x
   five.x
   six.x
   eight.x
   nine.x
   ten.x
   eleven.x
   sftp>
   ```

c. From the `sftp` prompt, use the `mget *` command to upload all the files in the `/OVS/seed_pool/ts_scripts` directory from **dom0** to **host01**.

```
sftp> mget *
Fetching .../two.x to two.x
...
sftp>
```

d. Use the `quit` command to exit `sftp`.

```
sftp> quit
```

Practice 22-2: System Boots into Single-User Mode

Overview

In this practice, your system boots into single-user mode by default, which is not what you want.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. Display the default system-state target.
   a. Use the `systemctl get-default` command to view the default system-state target.

```
# systemctl get-default
multi-user.target
```

   - The `multi-user.target` unit corresponds to run level 3 on a SysV init system.

   b. Use the `runlevel` command to display the current run level.

```
# runlevel
N 3
```

   - The `runlevel` command still exists in Oracle Linux 7 but is included only for compatibility reasons.

2. Execute the `two.x` program from the `root` user's home directory.

```
# cd
# pwd
/root
# ./two.x
```

3. Reboot the system.
   a. Use the `systemctl reboot` command to reboot your system.
   - It might take a couple minutes for the reboot to complete.

```
# systemctl reboot
...
```

   - After you reboot your system, your `ssh` session closes.

   b. Connect to **host01** by using VNC.
      1) Run the `vncviewer&` command.

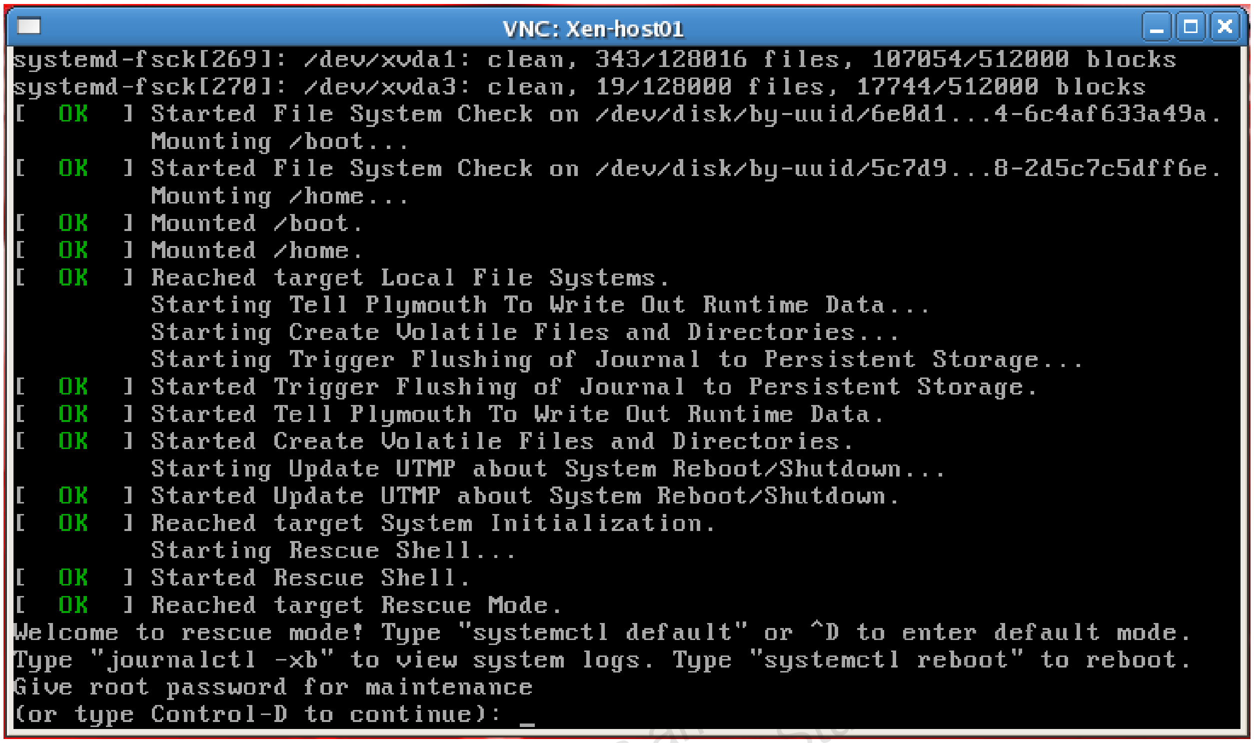
```
# vncviewer&
```

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

      2) Enter the command `localhost:<port_number>` command, substituting the correct port number for the **host01** guest. For example, if the port number is 5902, enter the following and click **Connect**:

```
localhost:5902
```

- The following window appears:



```
VNC: Xen-host01
systemd-fsck[269]: /dev/xvda1: clean, 343/128016 files, 107054/512000 blocks
systemd-fsck[270]: /dev/xvda3: clean, 19/128000 files, 17744/512000 blocks
[  OK  ] Started File System Check on /dev/disk/by-uuid/6e0d1...4-6c4af633a49a.
         Mounting /boot...
[  OK  ] Started File System Check on /dev/disk/by-uuid/5c7d9...8-2d5c7c5dff6e.
         Mounting /home...
[  OK  ] Mounted /boot.
[  OK  ] Mounted /home.
[  OK  ] Reached target Local File Systems.
         Starting Tell Plymouth To Write Out Runtime Data...
         Starting Create Volatile Files and Directories...
         Starting Trigger Flushing of Journal to Persistent Storage...
[  OK  ] Started Trigger Flushing of Journal to Persistent Storage.
[  OK  ] Started Tell Plymouth To Write Out Runtime Data.
[  OK  ] Started Create Volatile Files and Directories.
         Starting Update UTMP about System Reboot/Shutdown...
[  OK  ] Started Update UTMP about System Reboot/Shutdown.
[  OK  ] Reached target System Initialization.
         Starting Rescue Shell...
[  OK  ] Started Rescue Shell.
[  OK  ] Reached target Rescue Mode.
Welcome to rescue mode! Type "systemctl default" or ^D to enter default mode.
Type "journalctl -xb" to view system logs. Type "systemctl reboot" to reboot.
Give root password for maintenance
(or type Control-D to continue): _
```

- Note that you booted into rescue mode. In Oracle Linux 7, rescue mode is the same as single-user mode.

c. Enter the `root` password for maintenance.

- The `root` password is `oracle`.

```
Give root password for maintenance
(or type Control-D to continue): oracle
```

d. Use the `runlevel` command to display the current run level.

```
# runlevel
N 1
```

- Note that you are in single-user mode (run level is `1`).

4. Diagnose and fix the problem so that the system does not boot into single-user mode.
   - Review Lesson 4, “Oracle Linux 7 Boot Process”.
     – Which file allows you to view kernel boot parameters? Is `single` a kernel boot parameter?
     – Which file allows you to specify kernel boot parameters? Is `single` specified as a kernel boot parameter?
     – How do you change kernel boot parameters at boot time?
     – How do you permanently change kernel boot parameters?
5. When the problem is fixed, reboot the system to confirm it does not boot into single-user mode.

Solution 22-2: System Boots into Single-User Mode

Steps

1. Determine why the system boots into single-user mode.
   a. Use the `cat` command to view the kernel boot parameters in the `/proc/cmdline` file.

```
# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.8.13-35.3.1.el7urk.x86_64 ro root=UUID=...
ro vconsole.keymap=us crashkernel=257M@0M
vconsole.font=latarcyrheb-sun16 rhgb quiet single
```

   - Note the word `single` is a kernel boot parameter that is causing your system to boot into single-user mode.

   b. Use the `grep` command and search for the word "`single`" in the `/boot/grub2/grub.cfg` file.

```
# grep single /boot/grub2/grub.cfg
linux16 /vmlinuz-3.8.13-35... single
```

   - Note the word `single` appears on the kernel line for the `3.8.13-35` kernel.
   - This would cause the `3.8.13-35` kernel to boot into single-user mode by default.

   c. Use the `uname -r` command to determine which kernel is running.

```
# uname -r
3.8.13-35.3.1.el7uek.x86_64
```

   - The `3.8.13-35` kernel is running. The word `single` needs to be removed from the associated kernel line in the `/boot/grub2/grub.cfg` file.

2. Fix the system so it does not boot into single-user mode.
   a. Use the `vi` editor to modify `/boot/grub2/grub.cfg` and remove the word `single` at the end of the `linux16 /vmlinuz-3.8.13-35` kernel line.

```
# vi /boot/grub2/grub.cfg
...
linux16 /vmlinuz-3.8.13-35... single
...
```

   b. Press Ctrl + D to continue the boot process.
   - The login prompt appears after the boot process completes.

```
# CTRL-D
...
host01 login:
```

   c. Log in as `root`. Password is `oracle`.

```
host01 login: root
Password: oracle
```

   d. Use the `runlevel` command to display the current run level.

```
# runlevel
1 3
```

   - Note the run level is `3`. The previous run level was `1`.

3. Confirm the system does not boot into single-user mode.

   a. Use the `systemctl reboot` command to reboot your system.

      - It might take a couple minutes for the reboot to complete.

```
# systemctl reboot
...
```

      - After you reboot your system, your VNC session closes.

   b. You can connect to **host01** by using either VNC or `ssh`.

      - You might need to wait a few seconds for the reboot to complete
      - The `root` password is `oracle`.

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

   c. Use the `runlevel` command to display the current run level.

```
# runlevel
N 3
```

      - Note that the run level is now `3` and not single-user.

Practice 22-3: Status Commands Fail

Overview

In this practice, you note that some of the "status" utilities are not producing the expected output. You diagnose and fix this problem.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. Execute the `three.x` program from the `root` user's home directory.

```
# cd
# pwd
/root
# ./three.x
```

2. Run some "status" tools and note the errors.
   a. Run the `mpstat` command.

```
# mpstat
Linux 3.8.13-35...
Cannot open /proc/stat: No such file or directory
```

   b. Run the `iostat` command.

```
# iostat
Linux 3.8.13-35...
Cannot open /proc/stat: No such file or directory
```

   c. Run the `netstat` command.

```
# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags     Type     State   I-Node Path
```

   d. Run the `route` command.

```
# route
/proc/net/route: No such file or directory
INET (IPv4) not configured in this system.
```

   e. Run the `ifconfig` command.

```
# ifconfig
Warning: cannot open /proc/net/dev (No such file or directory).
Limited output.
...
```

- Note that none of these tools produce the expected output.

3. Diagnose and fix the problem with the “status” tools.
   - Review Lesson 5, “System Configuration”.
     - Commands that are failing are commands that provide information about the current state of the kernel.
     - Which file system contains a hierarchy of special files that represent the current state of the kernel?
     - How do you check if this file system is mounted?
     - If this file system is not mounted, how do you mount it?
4. Re-run some of the previous commands that failed to ensure that the problem was fixed.
   a. Run the `mpstat` command.
   b. Run the `iostat` command.
   c. Run the `netstat` command.
   d. Run the `route` command.
   e. Run the `ifconfig` command.

Solution 22-3: Status Commands Fail

Steps

1. Some of the output indicates a problem with the `proc` file system. Diagnose the problem.
   a. Use the `ls` command to display the contents of `/proc`.

```
# ls /proc
```

   – No output from this command suggests the `proc` file system is not mounted.

   b. Run the `mount` command to display mounted file systems.

```
# mount
mount: failed to read mtab: No such file or directory
```

   c. Run the `df -h` command to display mounted file systems.

```
# df -h
df: cannot read table of mounted file systems: No such file or
directory
```

2. Fix the problem.
   - It appears that the `/proc` file system is not mounted.

   a. Use the `systemctl reboot` command to reboot your system.
      – It might take a couple minutes for the reboot to complete.

```
# systemctl reboot
...
```

   – After you reboot your system, the `ssh` connection closes.

   b. Connect to **host01** by using `ssh`.
      – The `root` password is `oracle`.

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

3. Re-run some of the previous "status" commands to ensure that the problem is fixed.
   a. Run the `mpstat` command.
   b. Run the `iostat` command.
   c. Run the `netstat` command.
   d. Run the `route` command.
   e. Run the `ifconfig` command.

Practice 22-4: `cron` Job Fails to Run

Overview

In this practice, you diagnose and fix a problem that is preventing a `cron` job from running.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. Create a `crontab` for the `root` user.
   a. Use the `crontab -e` command to create a `crontab` job that runs the `vmstat` command every minute.

```
# crontab -e
* * * * * vmstat
```

   - The `crontab -e` command uses the `vi` editor. Save your changes and exit `vi`.

   b. After one minute, use the `tail` command to view the last few lines in the `/var/log/cron` log file.

```
# tail /var/log/cron
...
<date_time> host01 CROND[...] (root) CMD (vmstat)
```

   - Note there is an entry in this file that states the date and time that the `cron` daemon ran the `vmstat` command.
   - You also have mail, because the output from `cron` jobs is sent to the user's mailbox.

   c. Use the `mail` command to view the results of your job.

```
# mail
...
N # Cron Daemon  <date_time> ... "Cron <root@host01> vm"
&
```

   - Note you have mail that contains the output of the `vmstat` command.

   d. View the details of mailbox entries by entering the associated number and then press **Enter**.
   - To re-display the header, press **h** and then press **Enter**.
   - To quit the mail program, press **q** and then press **Enter**.

   e. Press **q** and then **Enter** to quit the mail program.

```
& q
```

2. Execute the `four.x` program from the `root` user's home directory.

```
# cd
# pwd
/root
# ./four.x
```

3. Edit the crontab for the root user.
   a. Use the crontab -e command to edit the crontab job and replace vmstat with iostat.

```
# crontab -e
* * * * * vmstat                              (old entry)
* * * * * iostat                              (new entry)
```

   b. After one minute, use the tail command to view the last few lines in the /var/log/cron log file.

```
# tail /var/log/cron
...
```

   - Note that there is not an entry in this file stating that the iostat command ran.

   c. Use the mail command to view your incoming mailbox.

```
# mail
...
&
```

   - Note that there is NO mail entry containing the output of the iostat command.

   d. Press **q** and then **Enter** to quit the mail program.

```
& q
```

4. Diagnose and fix the problem of the cron job failing to run.
   - Review Lesson 8, “Automating Tasks”.
     – Does the root user have a crontab entry?
     – Is the root user’s crontab entry valid?
     – Are the permissions correct on the root user’s crontab file?
     – Is the cron daemon running?
5. Verify that the cron job is running every minute.
   a. Run the mail command to view the output of the iostat command.
   b. Use the tail command to view /var/log/cron and ensure the cron job is running every minute.
6. Remove the root user’s crontab before continuing to the next practice.

Solution 22-4: `cron` Job Fails to Run

Steps

1. Diagnose the problem of the `cron` job failing to run.
   a. Use the `crontab -l` command to list the `crontab` entry.

```
# crontab -l
* * * * * iostat
```

   - Note that the `crontab` entry exists and the format is valid.

   b. Use the `ls -l` command to view the root user's `crontab` entry in the `/var/spool/cron` directory.

```
# ls -l /var/spool/cron
-rw-------. root root ... root
```

   - The `root` file exists with the proper permissions (read-write for the owner).

   c. Use the `cat` command to view the `/var/spool/cron/root` file.

```
# cat /var/spool/cron/root
* * * * * iostat
```

   - This confirms the `crontab` entry is valid.

   d. Use the `systemctl` command to view the status of the `crond` daemon.

```
# systemctl status crond
crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service: ...
   Active: inactive (dead) since ...
...
```

   - Note that the `crond` service is not running. This is the cause of the problem.

2. Fix the problem of the `cron` job failing to run.
   Use the `systemctl` command to start the `crond` daemon.

```
# systemctl start crond
```

3. Verify that the `cron` job is running every minute.
   a. After one minute, use the `tail` command to view the last few lines in the `/var/log/cron` log file.

```
# tail /var/log/cron
...
<date_time> host01 CROND[...] (root) CMD (iostat)
```

   - Note that there is now an entry in this file that states the date and time that the `cron` daemon ran the `iostat` command.

   b. Use the `mail` command to view the results of your job.

```
# mail
...
N # Cron Daemon  <date_time> ... "Cron <root@host01> io"
```

   - Note that you have mail that contains the output of the `iostat` command.

c. Press **q** and then **Enter** to quit the mail program.

```
& q
```

4. Remove the `root` user's `crontab` before continuing to the next practice.

```
# crontab -r
```

Practice 22-5: User Cannot Log In

Overview

In this practice, you diagnose and fix a problem that is preventing a user from being able to log in.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. Add a new user.
   a. User the `useradd john` command to add user `john`.

```
# useradd john
```

   b. Use the `passwd john` command to assign a password (`password`) to user `john`.
   - Disregard the “BAD PASSWORD” warning.

```
# passwd john
Changing password for user john.
New password: password
BAD PASSWORD: The password fails the dictionary check - it is
based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
```

2. Verify you can log in as user `john`.
   a. Use the `exit` command to log out as user `root`.

```
# exit
logout
```

   - The `ssh` connection closes.

   b. From **dom0**, use the `ssh` command to log in to **host01** as user `john`.
   - Password is `password`. Verify you can successfully log in as user `john`.

```
[dom0]# ssh john@host01
john@host01’s password: password
$ pwd
/home/john
$ whoami
john
```

   - You can successfully log in as user `john`.

   c. Use the `exit` command to log out as user `john`.

```
$ exit
logout
```

   - The `ssh` connection closes.

3. From **dom0**, log in to **host01** as `root`.

   a. Use the `ssh` command to log in to **host01**.

      - Password is `oracle`.

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

   b. Run the `five.x` program from the `root` user's home directory.

```
# cd
# pwd
/root
# ./five.x
```

   c. Use the `exit` command to log out.

```
# exit
logout
```

      - The `ssh` connection closes.

4. From **dom0**, log in to **host01** as user `john`.

   Use the `ssh` command to log in to **host01**.

      - Password is `password`.

```
[dom0]# ssh john@host01
root@host01's password: password
Permission denied, please try again.
root@host01's password: CTRL-C
```

      - Note you cannot log in as user `john`.
      - When prompted a second time for a password, press Ctrl + C.

5. Diagnose and fix the problem of user `john` not being able to log in.

      - Review Lesson 10, "User and Group Administration".
        - Which configuration file contains usernames? Is the user `john` present in this file?
        - Which configuration file contains user passwords? Is the user `john` present in this file?
        - If entries are missing in these files, how do you re-create the entries?
        - If passwords are corrupted or forgotten, how do you re-create passwords?

6. Verify you can successfully log in as user `john`.
7. Log in to **host01** as `root` in preparation for the next practice.

Solution 22-5: User Cannot Log In

Steps

1. Diagnose and fix the problem of user john not being able to log in.
   a. Log in as the root user. Password is oracle.

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

   b. Use the grep command to search for john in the /etc/passwd file.

```
# grep john /etc/passwd
john:x:1001:1001::/home/john:/bin/bash
```

   - Note the john entry is present.

   c. Use the grep command to search for john in the /etc/shadow file.

```
# grep john /etc/shadow
#john:$6$...:0:99999:7:::
```

   - Note the john entry is present but is commented out (# sign).

   d. Use the vi editor to modify /etc/shadow. Remove the # character from the beginning of the john line.

   - Because /etc/shadow is a read-only file, use :wq! to write, save, and exit vi after removing the # character.

```
# vi /etc/shadow
...
#john:$6$...:0:99999:7:::                    (old entry)
john:$6$...:0:99999:7:::                     (new entry)
```

   - Alternatively, you could run the following command to re-create a password entry in the /etc/shadow file for user john.

```
# passwd john
Changing password for user john.
...
```

   e. Use the exit command to log out as user root.

```
# exit
logout
```

   - The ssh connection closes.

2. Verify you can log in as user john.
   a. From **dom0**, use the ssh command to log in to **host01** as user john.

   - Password is password. Verify you can successfully log in as user john.

```
[dom0]# ssh john@host01
john@host01's password: password
$ pwd
/home/john
$ whoami
```

```
john
```

- Note that you can now log in as user `john`.

b. Use the `exit` command to log out as user `john`.

```
$ exit
Logout
```

c. Use the `ssh` command to log in to **host01** as `root` in preparation for the next practice.

- Password is `oracle`.

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

Practice 22-6: File System Troubleshooting

Overview

In this practice, you diagnose and fix an unmountable file system.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. Create a partition on `/dev/xvdb`.
   a. Use the `fdisk` command to partition `/dev/xvdb`.

```
# fdisk /dev/xvdb
...
Command (m for help):
```

   b. Add a new **primary** partition, giving the partition number **1**.

```
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
```

   c. Continue adding the new partition, using the entire disk as follows:

```
First sector (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): ENTER
Using default value 10485759
Partition 1 of type Linux and of size 5 GiB is set
```

   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.
```

2. Make a file system on `/dev/xvdb1`.

   Use the `mkfs` command to make an **ext4** file system on `/dev/xvdb1`.

```
# mkfs -t ext4 /dev/xvdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
...
```

```
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. Mount the file system.
   a. Use the mkdir command to create /Test.

```
# mkdir /Test
```

   b. Use the mount command to mount /dev/xvdb1 on /Test.

```
# mount /dev/xvdb1 /Test
```

4. Copy files to /Test.

   Use the cp command to copy /boot/init* to /Test. Use the ls command to display the contents of the /Test directory.

```
# cp /boot/init* /Test
# ls /Test
initramfs-0-rescue-...img
initramfs-3.10.0-123.el7.x86_64.img
initramfs-3.8.13-35.3.1.el7uek.x86_64.img
...
```

5. Unmount the file system on /dev/xvdb1.

   Use the umount command to unmount /Test.

```
# umount /Test
```

6. Execute the six.x program from the root user's home directory.

```
# cd
# pwd
/root
# ./six.x
```

7. Attempt to mount the file system on /dev/xvdb1 on /Test.

   Use the mount command as shown in Task 3b.

```
# mount /dev/xvdb1 /Test
mount: unknown filesystem type '(null)'
```

   - Note that the mount command fails.

8. Diagnose and fix the problem with the file system.
   - Review Lesson 11, "File Systems".
     – How do you specify the file system type with the mount command?
     – What commands are available to check and repair file systems?

9. Ensure you can mount the file system on /Test.
   - Ensure the files (initramfs*) exist on /Test.

Solution 22-6: File System Troubleshooting

Steps

1. Diagnose the problem.
   a. Run the `mount` command but include the `-t ext4` option to specify the file system type.

```
# mount -t ext4 /dev/xvdb1 /Test
mount: wrong fs type, bad option, bad superblock on /dev/xvdb1,
       missing codepage or helper program, or other error

       In some cases useful info is found in syslog - try
       dmesg | tail or so
```

   b. Per the preceding suggestion, run the `dmesg | tail` command.

```
# dmesg | tail
...
[ ...] EXT4-fs (xvdb1): VFS: Can't find ext4 filesystem
```

   - Note the VFS error: "Can't find ext4 filesystem" on `xvdb1`.

   c. Use the `tail` command to view the last lines in the `/var/log/messages` file.

```
# tail /var/log/messages
...
<date_time> host01 kernel: EXT4-fs (xvdb1): VFS: Can't find ext4
filesystem
```

   - Note the error message written to the `/var/log/messages` file is similar to the error message given by the `dmesg` command.

   d. Use the `tune2fs -l /dev/xvdb1` command to list the contents of the file system superblock.

```
# tune2fs -l /dev/xvdb1
tune2fs 1.42.0 (28-Dec-2013)
tune2fs: Bad magic number in super-block while trying to open
/dev/xvdb1
Couldn't find valid filesystem superblock
```

   - Note the "superblock" error.

   e. Use the `dumpe2fs /dev/xvdb1` command to dump file system information.

```
# dumpe2fs /dev/xvdb1
dumpe2fs 1.42.0 (28-Dec-2013)
dumpe2fs: Bad magic number in super-block while trying to open
/dev/xvdb1
Couldn't find valid filesystem superblock
```

   - Note this command reports the same "superblock" error on `/dev/xvdb1`.

2. Fix the file system.
   a. Use the `fsck /dev/xvdb1` command to check (and repair) the file system.
      - Press **Enter** to accept the default **y** (yes) answer.

```
# fsck /dev/xvdb1
fsck from util-linux 2.23.2
e2fsck 1.42.9 (28-Dec-2013)
ext2fs_open2: Bad magic number in super-block
fsck.ext2: Superblock invalid, trying backup blocks...
/dev/xvdb1 was not cleanly unmounted, check forced.
Resize inode not valid. Recreate<y>? ENTER
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free blocks count wrong for group #0 ...
Fix<y>? ENTER
Free blocks count wrong for group #1 ...
Fix<y>? ENTER
Free blocks count wrong ...
Fix<y>? ENTER
Free inodes count wrong for group #0 ...
Fix<y>? ENTER
Free inodes count wrong ...
Fix<y>? ENTER

/dev/xvdb1: *****    FILE SYSTEM WAS MODIFIED *****
/dev/xvdb1: ...
```

   b. Run the `fsck /dev/xvdb1` command a second time to check the file system.

```
# fsck /dev/xvdb1
fsck from util-linux 2.23.2
e2fsck 1.42.9 (28-Dec-2013)
/dev/xvdb1: clean ...
```

      - Note that the file system is fixed (clean).

3. Mount the file system.
   a. Use the `mount` command to mount `/dev/xvdb1` on `/Test`.

```
# mount /dev/xvdb1 /Test
```

      - The `mount` command is successful. The `fsck` command fixed the corrupt superblock.

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

```
# ls /Test
# ls /Test
initramfs-0-rescue-...img
initramfs-3.10.0-123.el7.x86_64.img
initramfs-3.8.13-35.3.1.el7uek.x86_64.img
...
```

- Note that all of the original files are present in the directory.

Practice 22-7: Logical Volume Space Is Exhausted

Overview

In this practice, you exhaust space in a logical volume and expand the file system to fix the problem.

Assumptions

You are the `root` user on the **host01** VM.

Tasks

1. Create three new partitions on `/dev/xvdd`.
   - Change the system ID on each new partition to `8e` (Linux LVM).

   a. Use the `fdisk` command to create a 1 GB **primary** partition on `/dev/xvdd` by using the following parameters:

```
# fdisk /dev/xvdd
...
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 (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): +1G
Partition 1 of type Linux and of size 1 GiB is set
```

   b. Use the "`t`" command to change the system ID on the new partition.

```
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'
```

   c. Continue using `fdisk` to create a second 1 GB **primary** partition on `/dev/xvdd` by using the following parameters:

```
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 (2099200-10485759, default 2099200): ENTER
```

```
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-10485759, default
10485759): +1G
Partition 2 of type Linux and of size 1 GiB is set
```

d. Use the "**t**" command to change the system ID on the new partition.

```
Command (m for help): t
Partition number (1-2, default 2): ENTER
Hex code (type L to list codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'
```

e. Continue using `fdisk` to create a third 1 GB **primary** partition on `/dev/xvdd` by using the following parameters:

```
Command (m for help): n
Partition type:
    p    primary partition (2 primary, 0 extended, 2 free)
    e    extended
Select (default p): ENTER
Using default response p
Partition number (3-4, default 3): ENTER
First sector (4196352-10485759, default 4196352): ENTER
Using default value 4196352
Last sector, +sectors or +size{K,M,G} (4196352-10485759, default
10485759): +1G
Partition 3 of type Linux and of size 1 GiB is set
```

f. Use the "**t**" command to change the system ID on the new partition.

```
Command (m for help): t
Partition number (1-3, default 3): ENTER
Hex code (type L to list codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'
```

g. Print the new partition table.

```
Command (m for help): p
Disk /dev/xvdb: 5368 MB, 5368709120 bytes, 10485760 sectors
...
    Device  Boot  Start        End    Blocks    Id    System
/dev/xvdd1         2048    2099199   1048576    8e    Linux LVM
/dev/xvdd2      2099200    4196351   1048576    8e    Linux LVM
/dev/xvdd3      4196352    6293503   1048576    8e    Linux LVM
```

h. 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.
```

2. 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/xvdd1         2048    2099199   1048576    8e    Linux LVM
/dev/xvdd2      2099200    4196351   1048576    8e    Linux LVM
/dev/xvdd3      4196352    6293503   1048576    8e    Linux LVM
```

   b. Use the `pvcreate` command to create physical volumes on all three partitions.

```
# pvcreate /dev/xvdd[123]
Physical volume "/dev/xvdd1" successfully created
Physical volume "/dev/xvdd2" successfully created
Physical volume "/dev/xvdd3" successfully created
```

3. Create a volume group.

   a. Use the `vgcreate` command to create a volume group named **myvolg** from the three physical volumes.

```
# vgcreate myvolg /dev/xvdd[123]
  Volume group "myvolg" successfully created
```

   b. Use the `vgs` command to display attributes of the volume group.

```
# vgs
 VG       #PV #LV #SN Attr   VSize  VFree
 myvolg     3   0   0 wz--n- 2.99g  2.99g
```

   c. Use the `pvs` command to display information about the physical volumes.

```
# pvs
 PV          VG      Fmt   Attr PSize     PFree
 /dev/xvdd1  myvolg  lvm2  a--  1020.00m  1020.00m
 /dev/xvdd2  myvolg  lvm2  a--  1020.00m  1020.00m
 /dev/xvdd3  myvolg  lvm2  a--  1020.00m  1020.00m
```

4. Create a logical volume.

   a. Use the `lvcreate` command to create a 4 MB logical volume named **myvol** from the **myvolg** volume group.

```
# lvcreate -L 4m -n myvol myvolg
  Logical volume "myvol" created
```

   b. Use the `lvs` command to display attributes of the logical volume.

```
# lvs
 LV    VG     Attr   LSize Pool Origin Data% Move Log Cpy%...
 myvol myvolg -wi-a- 4.00m
```

5. Create and mount a file system on the logical volume.

   a. List the `/dev` entries for the **myvol** logical volume.

```
# ls -l /dev/myvolg/myvol
lrwxrwxrwx.  /dev/myvolg/myvol -> ../dm-0
# ls -l /dev/mapper/myvolg-myvol
```

```
lrwxrwxrwx.  /dev/mapper/myvolg-myvol -> ../dm-0
```

b. 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
...
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
```

c. Create a `/myvol` mount point.

```
# mkdir /myvol
```

d. Mount the file system.

```
# mount /dev/mapper/myvolg-myvol /myvol
```

e. Run the `df -h` command to display mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/myvolg-myvol
              2.9M   45K   2.6M    2%  /myvol
```

f. Use the `vi` editor to edit the `/etc/fstab` file.

- Add the following entry to the end of the `/etc/fstab` file.

```
# vi /etc/fstab
/dev/mapper/myvolg-myvol /myvol  ext4  defaults  0  0
```

6. Exhaust space on logical volume.

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

```
# cd /myvol
```

b. Use the `dd if=/dev/zero of=bigfile bs=1024M count=5120` command to attempt to create a 5 MB file on `/myvol`.

```
# dd if=/dev/zero of=bigfile bs=1024 count=5120
dd: writing `bigfile': No space left on device
2797+0 records in
2796+0 records in
2863104 bytes (2.9 MB) copied...
```

- Note there is no space left on the device.

c. Use the `ls -l` command to display the contents of the `/myvol` directory.

```
# ls -l /myvol
-rw-r—r--. 1 root root 2879488 <date_time> bigfile
...
```

d. Run the `df -h` command to display mounted file systems.

```
# df -h
Filesystem     Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/myvolg-myvol
               2.9M   2.8M    0M  100%  /myvol
```

- Note the logical volume mounted on `/myvol` is full.

7. Determine what can be done to allocate more space to `/myvol`.
   - Review Lesson 14, "Storage Administration".
     - What is the total size of the logical volume?
     - What volume group does the logical volume belong to?
     - What is the total size of the volume group?
     - Can you expand the logical volume by using available space in the volume group?
     - Do you need to add more space to the volume group before you can expand the logical volume?
     - What is the command that allows you to allocate more space from a volume group to a logical volume?
8. Allocate an additional 1 GB to `/myvol`
   - After allocating more space, ensure you can successfully create a 5 MB file on `/myvol`.

Solution 22-7: Logical Volume Space Is Exhausted

Steps

1. Determine what is needed to increase the size of the `myvolg/myvol` logical volume and the file system.
   a. Use the `lvs` command to display attributes of the logical volume.

```
# lvs
 LV    VG     Attr    LSize  Pool Origin Data% Move Log Cpy%...
 myvol myvolg -wi-ao- 4.00m
```

- Note the total size of the `myvol` logical volume is 4 MB.
- Note the `myvol` logical volume belongs to the `mvolg` volume group.

   b. Use the `vgs` command to display attributes of the volume group.

```
# vgs
 VG       #PV #LV #SN Attr   VSize  VFree
 myvolg     3   1   0 wz--n- 2.99g  2.98g
```

- Note there is 2.98 GB available in the `myvolg` volume group.
- Therefore, you can expand the logical volume by using available space in the volume group.
- You do not need to add more space to the volume group.

2. Allocate an additional 1 GB to `/myvol`
   a. Use the `lvextend` command to increase the size of the `myvolg/myvol` logical volume and the file system by 1 GB.
   - Using the `-r` option resizes the underlying file system together with the logical volume.

```
# lvextend -L +1G -r myvolg/myvol
  Extending logical volume myvol to 1.00 GiB
  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 = 1, new_desc_blocks = 9
The file system on /dev/mapper/myvolg-myvol is now 1052672
blocks long.
```

   b. Use the `lvs` command to display attributes of the logical volume.

```
# lvs
 LV    VG     Attr    LSize  Pool Origin Data% Move Log Cpy%...
 myvol myvolg -wi-ao- 1.00g
```

- Note the `myvol` logical volume is now 1 GB in size.

c. Run the `df -h` command to display mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/myvolg-myvol
             1011M  3.1M   987M    1%  /myvol
```

- Note there is now 987 MB of available space on `/myvol`.

3. Attempt to create the `bigfile` again.

a. Ensure the current directory is `/myvol` before running the `dd` command.

```
# cd /myvol
# dd if=/dev/zero of=bigfile bs=1024 count=5120
5120+0 records in
5120+0 records in
5242880 bytes (5.2 MB) copied...
```

- Note you can now successfully create a 5 MB file on `/myvol`.

b. Use the `ls -l` command to display the contents of the `/myvol` directory.

```
# ls -l /myvol
-rw-r—r--. 1 root root 5242880 <date_time> bigfile
...
```

c. Run the `df -h` command to display mounted file systems.

```
# df -h
Filesystem    Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/myvolg-myvol
             1011M  5.3M   985M    1%  /myvol
```

- Note you still have 985 MB of space available on `/myvol`.

Practice 22-8: Network Connectivity Problem

Overview

In this practice, you diagnose and fix a network connectivity problem.

Assumptions

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

Tasks

1. From **dom0**, connect to **host01** VM using VNC.
   a. If you have an `ssh` connection to **host01**, use the `exit` command to close the connection.

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

   b. From **dom0**, connect to **host01** by using VNC.
      1) Run the `vncviewer&` command.

      ```
      # vncviewer&
      ```

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

      2) Enter the `localhost:<port_number>` command, substituting the correct port number for the **host01** guest. For example, if the port number is 5902, enter the following and click **Connect**:

      ```
      localhost:5902
      ```

      - The "**host01 login**:" prompt appears.
      - You might need to press Enter to see the login prompt.

   c. Log in as the `root` user.
   - The password is `oracle`.

   ```
   host01 login: root
   Password: oracle
   ```

2. Verify the network interface is configured properly.
   Use the `ping` command to contact **dom0** and **host03**. Press Ctrl + C to abort the commands after connectivity is confirmed.

   ```
   # ping dom0
   64 bytes from example.com (192.0.2.1): icmp_seq=1 ttl=64 ...
   64 bytes from example.com (192.0.2.1): icmp_seq=1 ttl=64 ...
   64 bytes from example.com (192.0.2.1): icmp_seq=1 ttl=64 ...
   CTRL-C
   # ping host03
   64 bytes from example.com (192.0.2.103): icmp_seq=1 ttl=64 ...
   64 bytes from example.com (192.0.2.103): icmp_seq=1 ttl=64 ...
   64 bytes from example.com (192.0.2.103): icmp_seq=1 ttl=64 ...
   CTRL-C
   ```

3. Execute the `eight.x` program from the `root` user's home directory.

```
# cd
# pwd
/root
# ./eight.x
```

4. Repeat Task 2 to test the network interface configuration.
   a. Use the `ping` command to contact to **dom0**.
      - Press Ctrl + C to abort the command.

```
# ping dom0
PING example.com (192.0.2.1) 56(84) bytes of data.
CTRL-C
```

   b. Use the `ping` command to contact to **host03**.
      - Press Ctrl + C to abort the command.

```
# ping host03
PING example.com (192.0.2.103) 56(84) bytes of data.
CTRL-C
```

   - The remote systems are no longer reachable.
5. Diagnose and fix the network connectivity problem.
   - Review Lesson 15, "Network Configuration".
     – How do you display the configuration of your network?
     – Are the network interfaces up?
     – What are the IP addresses of **dom0** and **host03**?
     – Are **dom0** and **host03** on the same network as **host01**?
     – Do you have a route to **dom0** and **host03**?
     – How do you view the route table?
     – In which files do you configure network interfaces?
     – Are these network interface configuration files configured properly?
     – Is the network service running?
6. Verify network connectivity to the remote hosts is working.
   a. Run the `ping dom0` command.
   b. Run the `ping host03` command.

Solution 22-8: Network Connectivity Problem

Steps

1. Diagnose the network connectivity problem.
   a. Use the `ip addr` command to display the configuration of your network interfaces.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff
    inet 192.0.3.101/24 brd 192.0.2.255 scope global eth0
    inet6 ...
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 brd 192.168.1.255 scope global eth1
    inet6 ...
...
```

   - Output shows that the `eth0` interface is `UP` and has an IP address of `192.0.3.101`.

   b. Use the `route` command to display the route table.

```
# route
Kernel IP routing table
Destination  Gateway      Genmask          Flags ...  Iface
default      example.com  0.0.0.0          UG    ...  eth0
example.com  0.0.0.0      255.255.255.255  UH    ...  eth0
192.0.3.0    0.0.0.0      255.255.255.0    U     ...  eth0
192.168.1.0  0.0.0.0      255.255.255.0    U     ...  eth1
```

   - The route table indicates the route to the `192.0.3.0` network is through `eth0`.

   c. Use the `cat` command to view the contents of the `/etc/hosts` file.

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

   - The contents of the `/etc/hosts` file indicate the IP address should be configured with `192.0.2` network addresses and not `192.0.3`.

2. Fix the network connectivity problem.

   a. Use the `vi` editor to modify `/etc/sysconfig/network-scripts/ifcfg-eth0` and change the `IPADDR` directive as shown.

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
...
IPADDR0=192.0.3.101                          (old value)
IPADDR0=192.0.2.101                          (new value)
...
```

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

   - Always restart the service whenever a configuration file is changed.
   - Because you are connected from **dom0** to **host01** using VNC, you will not lose your connection when restarting the network.

```
# systemctl restart network
```

3. Verify network connectivity to the remote hosts is working.

   Use the `ping` command to contact **dom0** and **host03**. Use Ctrl + C to abort the commands after connectivity is confirmed.

```
# ping dom0
64 bytes from example.com (192.0.2.1): icmp_seq=1 ttl=64 ...
64 bytes from example.com (192.0.2.1): icmp_seq=1 ttl=64 ...
64 bytes from example.com (192.0.2.1): icmp_seq=1 ttl=64 ...
CTRL-C
# ping host03
64 bytes from example.com (192.0.2.103): icmp_seq=1 ttl=64 ...
64 bytes from example.com (192.0.2.103): icmp_seq=1 ttl=64 ...
64 bytes from example.com (192.0.2.103): icmp_seq=1 ttl=64 ...
CTRL-C
```

   - Network connectivity to the remote hosts works.

Practice 22-9: NFS Permission Problem

Overview

In this practice, you diagnose and fix an NFS permission problem.

Assumptions

- You are the `root` user on the **host01** VM.
- You are the `root` user on the **host03** VM.

Tasks

Perform Task 1 from **host03**.

1. Configure **host03** as an NFS server.
   a. Use the `vi` editor and edit `/etc/exports` to appear as shown.
      - This entry exports `/home/oracle` to all client systems. Include the `rw` option.
      - Remove any existing entries in the `/etc/exports` file.

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

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

```
[host03]# systemctl restart nfs
```

   c. Use the `showmount` command to display exported file systems.

```
[host03]# showmount -e
Export list for host03.example.com:
/home/oracle *
```

   d. Disable the `firewalld` and `iptables` services by running the following commands.
      - A firewall rules could be written to allow NFS connectivity.
      - But for purposes of this practice, stop the services.

```
[host03]# systemctl stop firewalld
[host03]# systemctl stop iptables
```

   e. Use the `ls` command to list the contents of `/home/oracle` on **host03**.

```
[host03]# ls /home/oracle
Desktop  Documents  Downloads  jail  Music  Pictures  Public
Templates  Videos
```

Perform Tasks 2, 3, and 4 from **host01**.

2. Create a mount point in the `oracle` user's home directory on **host01**.
   a. Use the `su – oracle` command to become the `oracle` user.

```
[host01]# su - oracle
```

   b. As the `oracle` user, use the `mkdir` command to create a directory named `remote_home`.

```
[host01]$ mkdir remote_home
```

c. Use the `exit` command to log off as the `oracle` user.

```
[host01]$ exit
logout
```

- You are now the `root` user on **host01**.

3. Mount the exported `/home/oracle` file system on **host01**.

a. As the `root` user, use the `mount` command to mount the exported file system—`host03:/home/oracle`—on the local mount point, `/home/oracle/remote_home` with read-write permissions (`-o rw`).

```
[host01]# mount -t nfs -o rw host03:/home/oracle
/home/oracle/remote_home
```

b. Use the `df` command to verify that the NFS file system is mounted.

```
[host01]# df -h
Filesystem            Size  Used  Avail  Use%  Mounted on
...
host03:/home/oracle    1.9G   45M   1.8G    1%  /home/oracle...
```

4. Access the remote file system as the `oracle` user.

a. From **host01**, use the `su - oracle` command to become the `oracle` user.

- Use the `pwd` command to display the current directory.
- Use the `ls -l` command to view the content of the current directory.

```
[host01]# su - oracle
[host01]$ pwd
/home/oracle
[host01]$ ls -l
drwx------ ... remote_home
```

- Note you are in the `oracle` user's home directory and that the `remote_home` directory exists.

b. Use the `ls` command to list the contents of `remote_home`.

```
[host01]$ ls remote_home
Desktop  Documents  Downloads  jail  Music  Pictures  Public
Templates  Videos
```

- Note the contents of `/home/oracle/remote_home` on **host01** are the same as `/home/oracle` on **host03** because they are the same directories.

Perform Task 5 from both **host01** and **host03**. Note the host name in the prompt.

5. Confirm you have read-write permission on the NFS file system.

   a. As the `oracle` user on **host01**, use the `vi` editor to create a "`test`" file on `remote_home` to verify you have read-write permission.

```
[host01]$ cd ~/remote_home
[host01]$ pwd
/home/oracle/remote_home
[host01]$ vi test
created from host01
```

   b. From **host03**, use the `su - oracle` command to become the `oracle` user.

   - Use the `pwd` command to display the current directory.
   - Use the `ls` command to view the content of the current directory.

```
[host03]# su – oracle
[host03]$ pwd
/home/oracle
[host01]$ ls
Desktop  Documents  Downloads  jail  Music  Pictures  Public
Templates  test  Videos
```

   - Note the presence of the "`test`" file that was created from **host01**.

   c. As the `oracle` user on **host03**, use the `vi` editor to edit the "`test`" file.

```
[host03]$ vi test
created from host01
edited from host03
```

   d. From **host01**, use the `cat` command to view the contents of the edited "`test`" file in `/home/oracle/remote_home`.

```
[host01]$ cat ~/remote_home/test
created from host01
edited from host03
```

   - This confirms you have read-write permission.

Perform Tasks 6, 7, 8, and 9 from **host01**.

6. Unmount the NFS file system.

   a. From **host01**, use the `exit` command to log out as the `oracle` user.

```
[host01]$ exit
logout
```

   - You are now the `root` user on **host01**.

   b. Use the `umount` command to unmount `/home/oracle/remote_home`.

```
[host01]# umount /home/oracle/remote_home
```

7. On **host01**, execute the `nine.x` program from the `root` user's home directory.

```
[host01]# cd
[host01]# pwd
/root
[host01]# ./nine.x
```

8. Mount the exported `/home/oracle` file system on **host01**.
   a. As the `root` user, use the `mount` command to mount the exported file system—`host03:/home/oracle`—on the local mount point, `/home/oracle/remote_home` with read-write permissions (`-o rw`).

```
[host01]# mount -t nfs -o rw host03:/home/oracle
/home/oracle/remote_home
```

   b. Use the `df` command to verify that the NFS file system is mounted.

```
[host01]# df -h
Filesystem            Size  Used  Avail  Use%  Mounted on
...
host03:/home/oracle   1.9G   45M   1.8G    3%  /home/oracle...
```

9. Access the remote file system as the `oracle` user.
   a. Use the `su - oracle` command to become the `oracle` user.
      - Use the `pwd` command to display the current directory.
      - Use the `ls -l` command to view the content of the current directory.

```
[host01]# su - oracle
[host01]$ pwd
/home/oracle
[host01]$ ls -l
drwx------ ... remote_home
```

      - Note you are in the `oracle` user's home directory and that the `remote_home` directory exists.

   b. Use the `ls` command to list the contents of `remote_home`.

```
[host01]$ ls remote_home
ls: cannot open directory remote_home/: Permission denied
```

      - Note you get a permission error when attempting to view the contents of the NFS file system.

   c. Use the `exit` command to log off as the `oracle` user.

```
[host01]$ exit
logout
```

      - You are now the `root` user on **host01**.

10. Diagnose and fix the cause of the permission error.
    - Review Lesson 16, "File Sharing".
        - Since the `mount` command succeeded, you know that:
            - Network connectivity from **host01** to **host03** exists
            - The file system is being shared by the server
            - Host name resolution (DNS) is not a problem
            - The firewall is not blocking NFS packets
        - Check the share options on the server.
        - Check the permissions on the NFS share.
        - Check the mount options on the client.
        - Check the UIDs and the GIDs of the `oracle` user on the client and the server.
        - View the `/var/log/secure` log file for messages that could indicate the cause of the problem.
11. After fixing the problem, ensure the `oracle` user from **host01** has read-permission on the `/home/oracle/remote_home/test` file.

Note: Task 12 is a clean-up task. Perform this task only after the NFS permission problem is fixed.

12. After ensuring the problem is fixed, unmount the NFS share and restore the hosts to their beginning state.
    a. As the `root` user on **host01**, use the `umount` command to unmount `/home/oracle/remote_home`.

```
[host01]# umount /home/oracle/remote_home
```

    b. As the `root` user on **host03**, use the `vi` editor and delete all lines in `/etc/exports`.

```
[host03]# vi /etc/exports
/home/oracle *(rw)                    delete this entry
```

    c. As the `root` user on **host03**, use the `systemctl` command to stop the `nfs` service.

```
[host03]# systemctl stop nfs
```

    d. As the `root` user on **host03**, use the `systemctl` command to start the `firewalld` service.

```
[host03]# systemctl start firewalld
```

Solution 22-9: NFS Permission Problem

Steps

1. Diagnose the cause of the permission error.
   a. As the `root` user on **host03**, view the share options on the NFS file system.

```
[host03]# cat /var/lib/nfs/etab
/home/oracle
*(rw,sync,wdelay,hide,nocrossmnt,secure,root_squash,...
```

   - The NFS share has read-write (`rw`) permission, which is correct.

   b. View the permissions on the NFS file system.

```
[host03]# ls -ld /home/oracle
drwx------... oracle  oracle  ...  /home/oracle/
[host03]# ls -l /home/oracle/test
-rw-rw-r--... oracle  oracle  ...  /home/oracle/test
```

   - The `/home/oracle` directory has `rwx` permission for the owner (`oracle`), which is correct.
   - The `/home/oracle/test` file has `rw` permission for the owner (`oracle`) and the group (`oracle`), which is correct.

   c. As the `root` user on **host01**, view the `mount` options on the NFS file system.

```
[host01]# mount
...
host03:/home/oracle on /home/oracle/remote_home type nfs4
(rw,relatime,vers=4.0,...,clientaddr=192.0.2.101,...,addr=192.0.
2.103)
```

   - The NFS share is mounted with read-write (`rw`) permission, which is correct.

   d. From **host03**, use the `grep` command to view the UID and GID for `oracle` in the `/etc/passwd` and the `/etc/group` files.

```
[host03]# grep oracle /etc/passwd
oracle:x:1000:1000:Oracle Student:/home/oracle:/bin/bash
[host03]# grep oracle /etc/group
...
oracle:x:1000:oracle
...
```

   - Note the `oracle` user's UID and GID are `1000` on **host03**.

   e. From **host01**, use the `grep` command to view the UID and GID for `oracle` in the `/etc/passwd` and the `/etc/group` files.

```
[host01]# grep oracle /etc/passwd
oracle:x:1055:1055:Oracle Student:/home/oracle:/bin/bash
[host01]# grep oracle /etc/group
oracle:x:1055:
```

   - Note the `oracle` user's UID and GID are `1055` on **host01**.
   - This difference in UIDs and GIDs is the cause of the permission error.

f. On **host01**, use the `tail` command to view the latest messages written to the `/var/log/secure` file.

```
[host01]# tail /var/log/secure
...
<date_time> host01 userdel[...]: delete user 'oracle'
<date_time> host01 userdel[...]: delete 'oracle' from group ...
<date_time> host01 userdel[...]: removed group 'oracle' ...
<date_time> host01 userdel[...]: removed shadow group 'oracl...
<date_time> host01 useradd[...]: new group: name=oracle, ...
<date_time> host01 useradd[...]: new user: name=oracle, ...
...
```

- Note the `oracle` user was deleted on **host01** and re-added with UID and GID of `1055`.

2. Fix the permission problem by changing `oracle` UID and GID for `oracle` on **host01** from `1055` to `1000`.

- The `oracle` UID and GID on **host01** needs to be the same as `oracle` user and group (`1000`) on **host03**.
- The easiest way to fix this is to delete the `oracle` user on **host01** and then re-add with the proper UID and GID.
- The following steps are provided to make the changes without deleting and re-adding the user.

a. Before changing UIDs and GIDs, use the `umount` command to unmount the NFS file system on **host01**.

```
[host01]# umount /home/oracle/remote_home
```

b. Use the `usermod -u` command to change the UID for the `oracle` user.

```
[host01]# usermod -u 1000 oracle
```

c. Use the `groupmod -g` command to change the GID for the `oracle` group.

```
[host01]# groupmod -g 1000 oracle
```

d. Use the following `find` command to change the owner permissions to the new UID.

- Redirect error messages to `/dev/null`.

```
[host01]# find / -user 1055 -exec chown -h 1000 {} \; 2>
/dev/null
```

e. Use the following `find` command to change the group permissions to the new GID.

- Redirect error messages to `/dev/null`.

```
[host01]# find / -group 1055 -exec chgrp -h 1000 {} \; 2>
/dev/null
```

f. Use the `usermod -g` command to change the GID for the `oracle` user.

- This command might return a "`usermod: no changes`" message, which is of no concern.

```
[host01]# usermod -g 1000 oracle
```

3. On **host01**, mount the exported /home/oracle file system on **host01**.

   a. As the root user, use the mount command to mount the exported file system—host03:/home/oracle—on the local mount point, /home/oracle/remote_home with read-write permissions (-o rw).

```
[host01]# mount -t nfs -o rw host03:/home/oracle
/home/oracle/remote_home
```

   b. Use the df command to verify that the NFS file system is mounted.

```
[host01]# df -h
Filesystem             Size  Used  Avail  Use%  Mounted on
...
host03:/home/oracle    1.9G   45M   1.8G    3%  /home/oracle...
```

4. Attempt to access the remote file system as the oracle user.

   a. From **host01**, use the su - oracle command to become the oracle user.

   - Use the pwd command to display the current directory.
   - Use the ls -l command to view the content of the current directory.

```
[host01]# su - oracle
[host01]$ pwd
/home/oracle
[host01]$ ls -l
drwx------ ... remote_home
```

   - Note you are in the oracle user's home directory and that the remote_home directory exists.

   b. Use the ls command to list the contents of remote_home.

```
[host01]$ ls remote_home
Desktop  Documents  Downloads  jail  Music  Pictures  Public
Templates  test  Videos
```

   - You can now view the contents of the NFS file system.

   c. Ensure you have read-permission by using the vi command to edit the "test" file.

```
[host01]$ cd ~/remote_home
[host01]$ vi test
<make some changes to the content>
```

   - You are able to successfully read and write to files on the NFS file system.

5. After ensuring the problem is fixed, unmount the NFS share and restore the hosts to their beginning state.

   a. Use the `exit` command to log off as the `oracle` user.

   ```
   [host01]$ exit
   logout
   ```

   - You are now the `root` user on **host01**.

   b. As the `root` user on **host01**, use the `umount` command to unmount `/home/oracle/remote_home`.

   ```
   [host01]# umount /home/oracle/remote_home
   ```

   c. As the `root` user on **host03**, use the `vi` editor and delete all lines in `/etc/exports`.

   ```
   [host03]# vi /etc/exports
   /home/oracle *(rw)                    delete this entry
   ```

   d. As the `root` user on **host03**, use the `systemctl` command to stop the `nfs` service.

   ```
   [host03]# systemctl stop nfs
   ```

   e. As the `root` user on **host03**, use the `systemctl` command to start the `firewalld` service.

   ```
   [host03]# systemctl start firewalld
   ```

Practice 22-10: Remote Access Problem

Overview

In this practice, you diagnose and fix a remote access problem.

Assumptions

- You are the `root` user on the **host01** VM.
- You are the `root` user on the **host03** VM.

Tasks

1. Ensure remote connectivity is working from **host03** to **host01**.
   a. As the `root` user on **host03**, confirm that you can use the `ssh` command to connect to **host01**.
      - Password is `oracle`.
      - Use the `hostname` command to confirm that you did connect.

```
[host03]# ssh host01
root@host01's password: oracle
[host01]# hostname
host01.example.com
```

   b. Use the `logout` command to log off.

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

2. Ensure remote connectivity is working from **host01** to **host03**.
   a. As the `root` user on **host01**, confirm that you can use the `ssh` command to connect to **host03**.
      - Password is `oracle`.
      - Use the `hostname` command to confirm that you did connect.

```
[host01]# ssh host03
root@host03's password: oracle
[host03]# hostname
host03.example.com
```

   b. Use the `logout` command to log off.

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

3. From **host01**, execute the `ten.x` program from the `root` user's home directory.

```
[host01]# cd
[host01]# pwd
/root
[host01]# ./ten.x
```

4. Ensure remote connectivity is working from **host01** to **host03**.

a. As the `root` user on **host01**, confirm that you can use the `ssh` command to connect to **host03**.

```
[host01]# ssh host03
root@host03's password: oracle
[host03]# hostname
host03.example.com
```

b. Use the `logout` command to log off.

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

5. Ensure remote connectivity is working from **host03** to **host01**.

   As the `root` user on **host03**, confirm that you can use the `ssh` command to connect to **host01**.

```
[host03]# ssh host01
ssh: connect to host host01 port 22: No route to host
```

   - Note the `ssh` command fails.

6. Diagnose and fix the remote connectivity problem.
   - Think about what you can and what you cannot do:
     - You can `ssh` from **host01** to **host03**
     - You cannot `ssh` from **host03** to **host01**
   - Can you `ping` **host01** from **host03**? If so, then there is nothing wrong with the network interface configuration.
   - Review Lesson 17, "OpenSSH".
     - Ensure the `sshd` service is configured properly and that the service is running.
   - Note the error message, "`ssh: connect to host host01 port 22: No route to host`."
     - View the `/var/log/messages` log file for entries related to `ssh` or port `22`.
     - View the `/var/log/secure` log file for entries related to `ssh` or port `22`.
   - Review Lesson 18, "Security Administration".
     - Is the firewall prohibiting a connection on port `22`?
     - Is there a TCP wrapper configured that is causing the problem?

7. After fixing the problem, ensure you can `ssh` from **host03** to **host01**.

Solution 22-10: Remote Access Problem

Steps

1. Attempt to diagnose the problem by testing network connectivity.

   From **host03**, use the `ping host01` command to test network connectivity to **host01**.

```
[host03]# ping host01
64 bytes from example.com (192.0.2.101): icmp_seq=1 ttl=64 ...
64 bytes from example.com (192.0.2.101): icmp_seq=1 ttl=64 ...
64 bytes from example.com (192.0.2.101): icmp_seq=1 ttl=64 ...
CTRL-C
```

   - Note the `ping` command verifies network connectivity to **host01** exists.

2. Attempt to diagnose the problem by checking `sshd` configuration.

   a. From **host01**, use the `systemctl` command to view the status of the `sshd` service.

```
[host01]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service...
   Active: active (running) since ...
...
```

   - The `sshd` service is running.

   b. From **host03**, use the `systemctl` command to view the status of the `sshd` service.

```
[host03]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service...
   Active: active (running) since ...
...
```

   - The `sshd` service is running.

   c. Use the `grep` command to search for "`ssh`" in the `/var/log/messages*` files.

   - Repeat the command but search for "`port 22`".

```
[host01]# grep ssh /var/log/messages*
...
[host01]# grep "port 22" /var/log/messages*
...
```

   - Note that neither command returned output that indicates the cause of the problem.

   d. Use the `grep` command to search for "`ssh`" in the `/var/log/secure` file.

   - Repeat the command but search for "`port 22`".

```
[host01]# grep ssh /var/log/secure
...
[host01]# grep "port 22" /var/log/secure
...
```

   - Note that the <*date_time*> stamp on the log file entries does not correspond to the <*date_time*> of the problem.

- Note that there are several "`pam`" messages related to the `sshd` service.
  - PAM (Pluggable Authentication Modules) is covered in another course.
  - PAM configuration could cause this problem in the real world, but because PAM has not been covered in this course, you can conclude PAM is not the cause of the problem.

3. Attempt to diagnose the problem by checking firewall (`firewalld` and `iptables`) configuration.
   a. Use the `systemctl` command to view the status of the `iptables` service.

```
[host01]# systemctl status iptables
iptables.service
   Loaded: not-found (Reason: No such file or directory)
   Active: inactive (dead)
```

   - Note that the `iptables` service is not running and not even installed on **host01**.

   b. Use the `systemctl` command to view the status of the `firewalld` service.

```
[host01]# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service...
   Active: active (running) since ...
...
```

   - Note that the `firewalld` service is running.

   c. Use the `firewall-cmd` command to list the trusted services.

```
[host01]# firewall-cmd --list-services
dhcpv6-client
```

   - Note that the `dhcpv6-client` service is trusted.
   - Note that the `ssh` service is not trusted.
     - This is the cause of the problem.
   - To fix the problem, either stop the `firewalld` service or trust the `ssh` service.

4. Fix the remote connectivity problem by trusting the `ssh` service when `firewalld` is running.
   a. Use the `firewall-cmd` command to trust the `ssh` service.

```
[host01]# firewall-cmd --add-service=ssh
success
```

   b. Use the `firewall-cmd` command to list the trusted services.

```
[host01]# firewall-cmd --list-services
dhcpv6-client ssh
```

   - Note that the `ssh` service is now trusted.

5. Ensure remote connectivity is working.
   a. From **host03**, confirm that you can use the `ssh` command to connect to **host01**.
      - Password is `oracle`.
      - Use the `hostname` command to confirm that you did connect.

```
[host03]# ssh host01
root@host01's password: oracle
[host01]# hostname
host01.example.com
```

   b. Use the `logout` command to log off.

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

Practice 22-11: Log File Is Not Getting Updated

Overview

In this practice, you diagnose and fix a problem with a log file not getting updated.

Assumptions

- You are the `root` user on the **host01** VM.
- You are the `root` user on the **host03** VM.

Tasks

1. From **host01**, set the `oracle` user's password to `oracle`.

   Use the `passwd oracle` command to change the password.

   - Ignore the "BAD PASSWORD" message.

```
[host01]# passwd oracle
Changing password for user oracle.
New password: oracle
BAS PASSWORD: The password is shorter than 8 characters
Retype new password: oracle
passwd: all authentication tokens updated successfully.
```

2. From **host03**, confirm you can use the `ssh` command to log in to **host01** as the `oracle` user.

   a. Use the `exit` command to log out as the `root` user.

   - Use the `whoami` command to confirm you are the `oracle` user.

```
[host03]# exit
logout
[host03]$ whoami
oracle
```

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

   - Password is `oracle`.
   - Use the `hostname` command to verify you were able to log in.

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

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

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

3. From **host01**, save a copy of the `/var/log/secure` file.

   Use the `cp` command to copy `/var/log/secure` to `~/secure_before`.

```
[host01]# cp /var/log/secure ~/secure_before
```

- You review the differences in these copies later in this practice.

4. From **host01**, configure the `pam_nologin` authentication module to prevent non-`root` login.
   - PAM is covered in another course.
     – For the purposes of this practice, PAM checks for the existence of the `/etc/nologin` file and if the file exists, remote logins by non-`root` users are denied and the content of this file is displayed as an error message.

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

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

5. As the `oracle` user on **host03**, attempt to log in to **host01**.

   Use the `ssh` command to connect to **host01**.
   - Password is `oracle`.

```
[host03]$ ssh host01
oracle@host01's password: oracle
No logins allowed at this time.
Connection closed by 192.0.2.101
```

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

6. From **host01**, view the new entries in the `/var/log/secure` log file.

   Use the `diff` command to view the differences in the `/var/log/secure` log file and the copy of the log file you made in Task 3.

```
[host01]# diff /var/log/secure ~/secure_before
< <date_time> host01 [sshd[...]: Failed password for oracle from
192.0.2.103 port ... ssh2
< <date_time> host01 [sshd[...]: fatal: Access denied for user
oracle by PAM account configuration
```

   - Note that two entries were written to this file when the `oracle` user attempted to `ssh` from **host03** to **host01**.
   - Also note that one entry specifically references PAM account configuration.

7. From **host01**, execute the `eleven.x` program from the `root` user's home directory.

```
[host01]# cd
[host01]# pwd
/root
[host01]# ./eleven.x
```

8. From **host01**, save a copy of the `/var/log/secure` file.

   Use the `cp` command to copy `/var/log/secure` to `~/secure_before`.
   - Answer **y** to overwrite the file.

```
[host01]# cp /var/log/secure ~/secure_before
cp: overwrite '/root/secure_before'? y
```

   - You review the differences in these copies later in this practice.

9. As the `oracle` user on **host03**, attempt to log in to **host01**.

   Use the `ssh` command to connect to **host01**. Password is `oracle`.

```
[host03]$ ssh host01
oracle@host01's password: oracle
No logins allowed at this time.
Connection closed by 192.0.2.101
```

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

10. From **host01**, view the new entries in the `/var/log/secure` log file.

    Use the `diff` command to view the differences in the `/var/log/secure` log file and the copy of the log file you made in Task 8.

```
[host01]# diff /var/log/secure ~/secure_before
```

    - Note that there are no differences in the files.
    - The `/var/log/secure` log file is not getting updated as expected.

11. Diagnose and fix the problem of the log file not getting updated.
    - Review Lesson 21, "System Logging".
      – Is the logging daemon running?
      – Is logging configured for the `/var/log/secure` log file?

12. After fixing the problem, ensure the `/var/log/secure` log file is getting updated when you attempt to `ssh` from **host03** to **host01** as the `oracle` user.

13. Return **host01** to the original state.

    Use the `rm` command to remove the `/etc/nologin` file.

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

Solution 22-11: Log File Is Not Getting Updated

Steps

1. Diagnose the cause of the logging problem.
   a. As the `root` user on **host01**, view the status of the `rsyslog` service.

```
[host01]# systemctl status rsyslog
rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service...
   Active: active (running) since ...
...
```

   - The `rsyslogd` service is running.

   b. Use the `grep` command to search for `/var/log/secure` in the `rsyslog` configuration file.

```
[host01]# grep /var/log/secure /etc/rsyslog.conf
#authpriv.*                          /var/log/secure
```

   - Note the configuration file contains an entry for `/var/log/secure`; however, the entry is commented out (preceded by a `#` sign).

2. Fix the problem.
   a. Use the `vi` editor to remove the `#` from the beginning of the line containing `/var/log/secure`.

```
[host01]# vi /etc/rsyslog.conf
...
#authpriv.*           /var/log/secure           (old entry)
authpriv.*            /var/log/secure           (new entry)
...
```

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

```
[host01]# systemctl restart rsyslog
```

3. Verify the `/var/log/secure` log file is getting updated.
   a. From **host01**, use the `cp` command to copy `/var/log/secure` to `~/secure_before`.
   - Answer **y** to overwrite the file.

```
[host01]# cp /var/log/secure ~/secure_before
cp: overwrite '/root/secure_before'? y
```

   b. As the `oracle` user on **host03**, use the `ssh` command to connect to **host01**.
   - Password is `oracle`.

```
[host03]$ ssh host01
oracle@host01's password: oracle
No logins allowed at this time.
Connection closed by 192.0.2.101
```

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

c. From **host01**, use the `diff` command to view the differences in the `/var/log/secure` log file and the copy, `~/secure_before`.

```
[host01]# diff /var/log/secure ~/secure_before
< <date_time> host01 [sshd[...]: Failed password for oracle from 192.0.2.103 port ... ssh2
< <date_time> host01 [sshd[...]: fatal: Access denied for user oracle by PAM account configuration
```

- Note that entries are now being written to the `/var/log/secure` file as expected.

4. Return **host01** back to the original state.

   Use the `rm` command to remove the `/etc/nologin` file.

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

Appendix A: Source Code for Problem-Causing Executables

Overview

In this appendix, the source code for the executables that cause the problems are given.

Practice 22-2: System Boots into Single-User Mode

The "`two.x`" script appends the word `single` to the end of the kernel line for the default kernel using the following commands:

```
#!/bin/bash
cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.save
sed -i -e '/vmlinuz-3.8.13/s/$/ single/' /boot/grub2/grub.cfg
```

This result of this program follows:

```
# cat /boot/grub2/grub.cfg
...
linux16 /vmlinuz-3.8.13-35... single...
...
```

Practice 22-3: Status Commands Fail

The "`three.x`" script runs the following `umount` command to unmount the `proc` file system.

```
#!/bin/bash
umount -l /proc
```

Practice 22-4: `cron` Job Fails to Run

The "`four.x`" script stops the `crond` service by running the following command:

```
#!/bin/bash
systemctl stop crond stop > /dev/null
```

Practice 22-5: User Cannot Log In

The "`five.x`" script inserts the "`#`" sign at the beginning of the `john` line in the `/etc/shadow` file by running the following commands:

```
#!/bin/bash
cp /etc/shadow /etc/shadow.save
sed -i -e '/john/s/^/#/' /etc/shadow
```

The result of this program follows:

```
# cat /etc/shadow
...
#john:$6$...:0:99999:7:::
```

Practice 22-6: File System Troubleshooting

The “six.x” script runs the following dd command to corrupt the file system superblock.

```
#!/bin/bash
dd if=/dev/zero of=/dev/xvdb1 bs=1024 skip=1000 count=300 2>
/dev/null
```

Practice 22-8: Network Connectivity Problem

The “eight.x” script changes the IP address of eth0 on **host01** from 192.0.2.101 to 192.0.3.101 by running the following commands:

```
#!/bin/bash
cd /etc/sysconfig/network-scripts
/bin/cp ifcfg-eth0 ~/ifcfg-eth0.save
sed –e 's/2.101/3.101/' ifcfg-eth0 > ifcfg-eth0.new
/bin/mv ifcfg-eth0.new ifcfg-eth0
cd
systemctl restart network > /dev/null
```

The result of this program follows:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
...
IPADDR=192.0.3.101
...
# ifconfig eth0
eth0      Link encap: Ethernet  HWaddr: 00:16:3E:00:01:01
          inet addr: 192.0.3.101 ...
...
```

Practice 22-9: NFS Permission Problem

The “nine.x” script deletes the oracle user, re-adds the oracle user with UID=1055, and re-creates the /home/oracle/remote_home directory by running the following commands:

```
#!/bin/bash
userdel –r oracle
useradd –u 1055 oracle
mkdir /home/oracle/remote_home
chown –R oracle.oracle /home/oracle
```

The result of this program follows:

```
# grep oracle /etc/passwd
oracle:x:1055:1055::/home/oracle:/bin/bash
# ls –l /home/oracle
drwxr-xr-x. 2 oracle oracle 4096 <date_time> remote_home
```

Practice 22-10: Remote Access Problem

The "ten.x" script removes the firewalld rule on **host01**, which trusts the sshd service by running the following commands:

```
#!/bin/bash
firewall-cmd --remove-service=ssh > /dev/null
```

Practice 22-11: Log File Is Not Getting Updated

The "eleven.x" script inserts the "#" sign at the beginning of the secure line in the /etc/rsyslog.conf file and restarts the rsyslog service by running the following commands:

```
#!/bin/bash
cp /etc/rsyslog.conf /etc/rsyslog.conf.save
sed -i -e '/secure/s/^/#/' /etc/rsyslog.conf
systemctl restart rsyslog > /dev/null
```

The result of this program follows:

```
# cat /etc/rsyslog.conf
...
#authpriv.*                                        /var/log/secure
```

Appendixes: Remote Access Options

Chapter 23

Appendixes: Remote Access Options – Overview

Appendixes Overview

The four appendixes show various options for accessing your student PC remotely:

- Appendix A: Using an NX Client to Connect to **dom0**
- Appendix B: Using an NX Player to Connect to **dom0**
- Appendix C: Using VNC (TightVNC) to Connect Directly to VM Guests
- Appendix D: Using NoMachine Version 4 to Connect to **dom0**

Appendix A: Using an NX Client to Connect to dom0

Overview

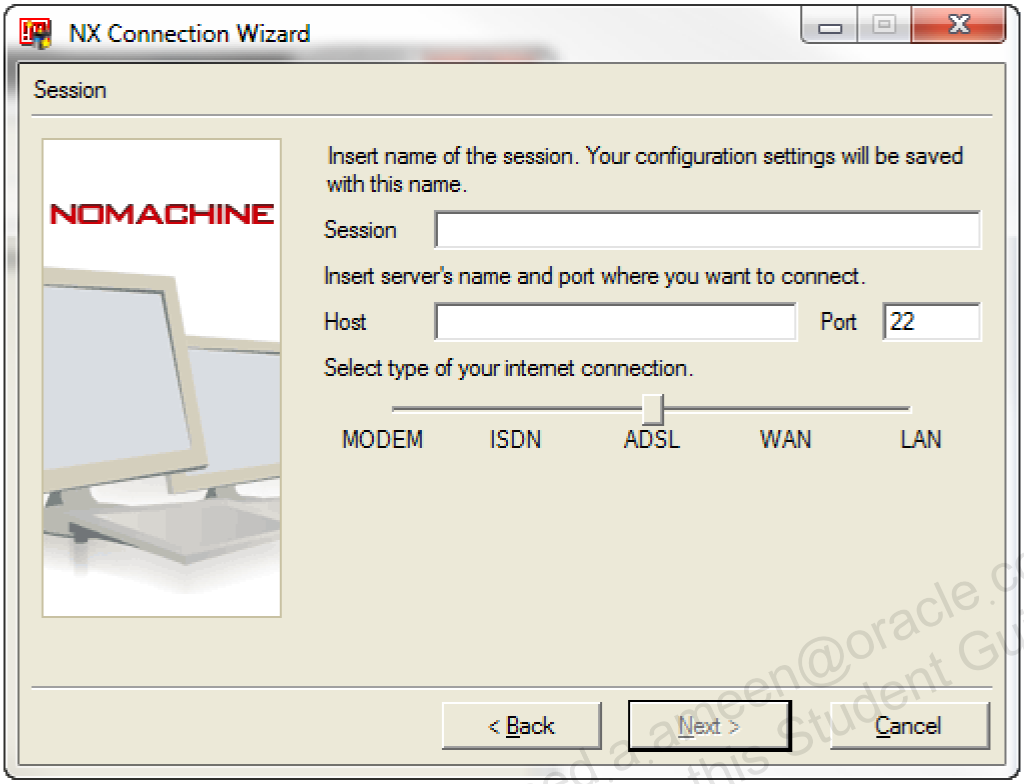
This appendix discusses accessing your student PC (**dom0**) remotely by using NX Client. The NX Client in this appendix is NX Client for Windows, Version 3.5.0-9.

Steps

1. Install **NX Client** (if necessary) from http://www.nomachine.com/download.php.
2. Run **NX Client** (for example, select **NX Client for Windows** from the Windows Start menu).
   - An NX Connection Wizard steps you through creating the initial session.
   - The following Welcome window appears.

   a. Click **Next**.

- The following Session window appears.



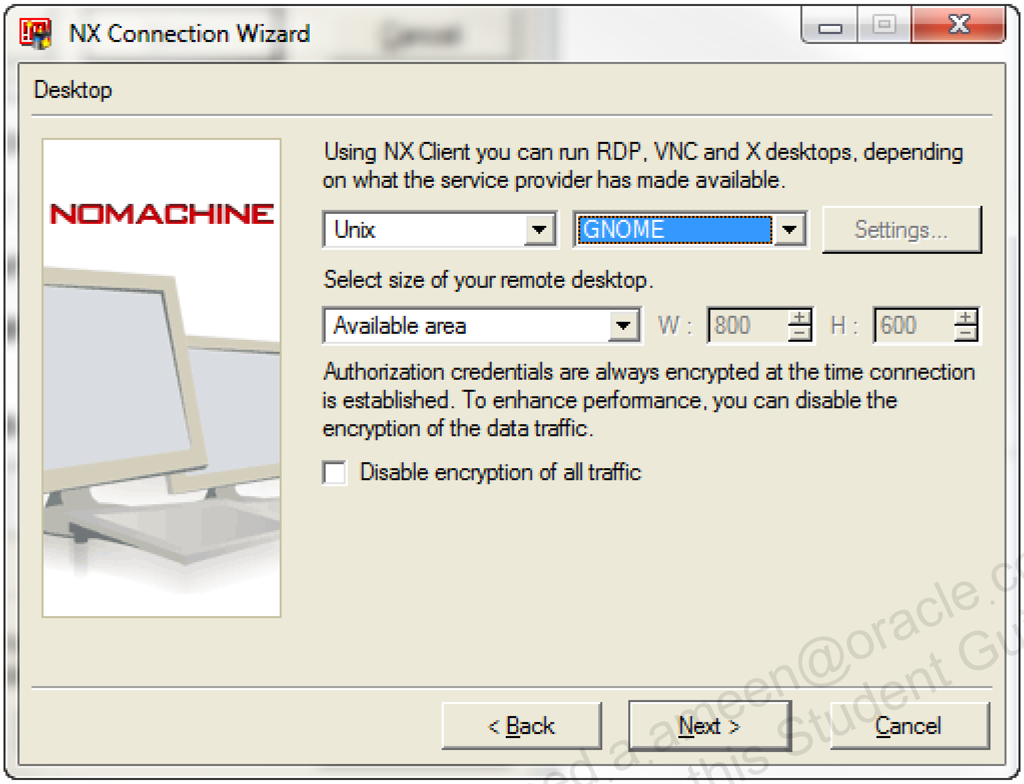
b. Enter anything you like for **Session** (for example, `OracleLinux`).

c. Enter the IP address (provided by your instructor) for **Host**.

d. Accept the remaining defaults and click **Next**.

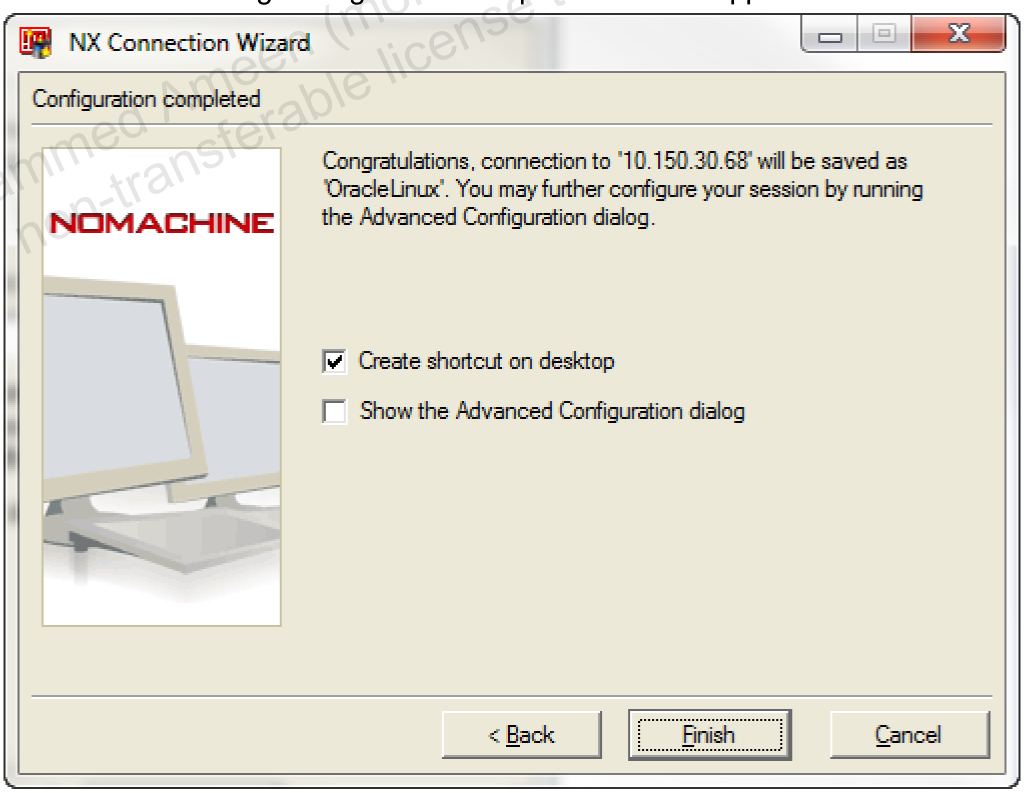
e. The Desktop window appears. Change KDE to **GNOME** by selecting from the drop-down list.

- Your window must look like the following:



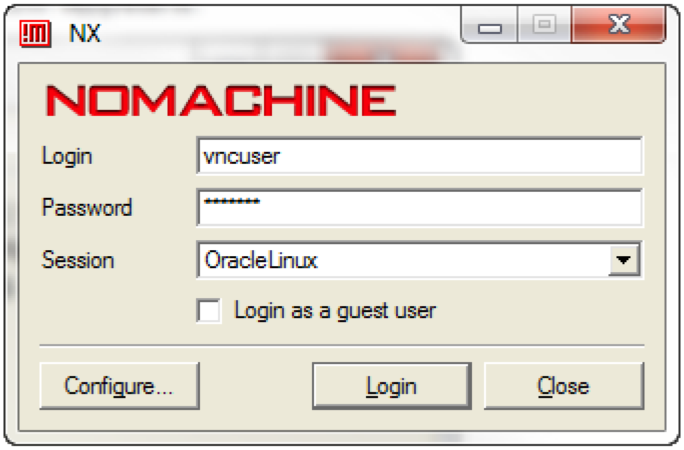
f. Accept all other defaults and click **Next**.

- The following Configuration completed window appears.



g. Click **Finish**.

- The NX Login window appears.



h. For **Login**, enter `vncuser`.

i. For **Password**, enter `vnctech`.

j. Your **Session** defaults to the session that you just created. In this example, the **Session** is OracleLinux. Your session name may be different.

k. Click **Login**.

- The **dom0** GNOME virtual desktop window appears.
- Future connections will bypass the configuration wizard and only bring up the NX Login window.

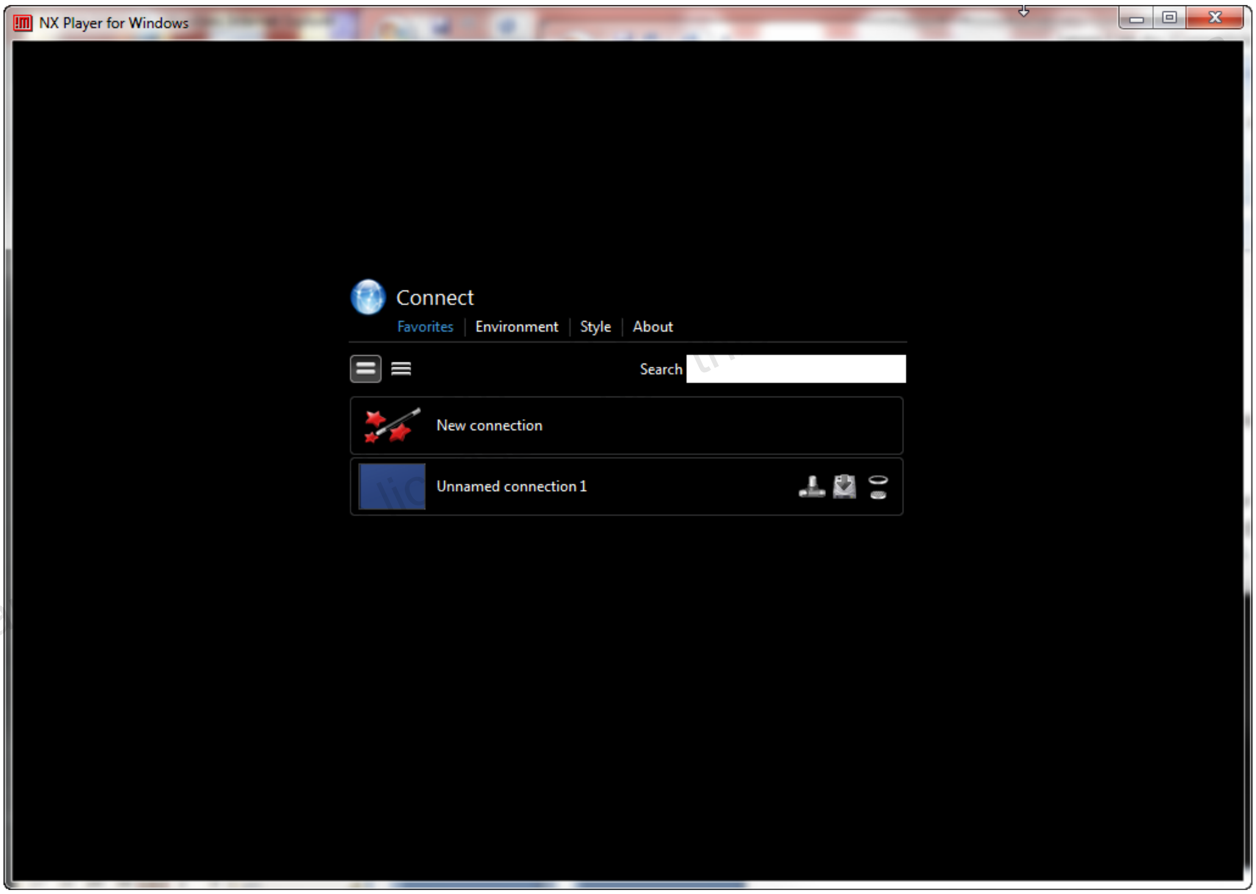
Appendix B: Using an NX Player to Connect to dom0

Overview

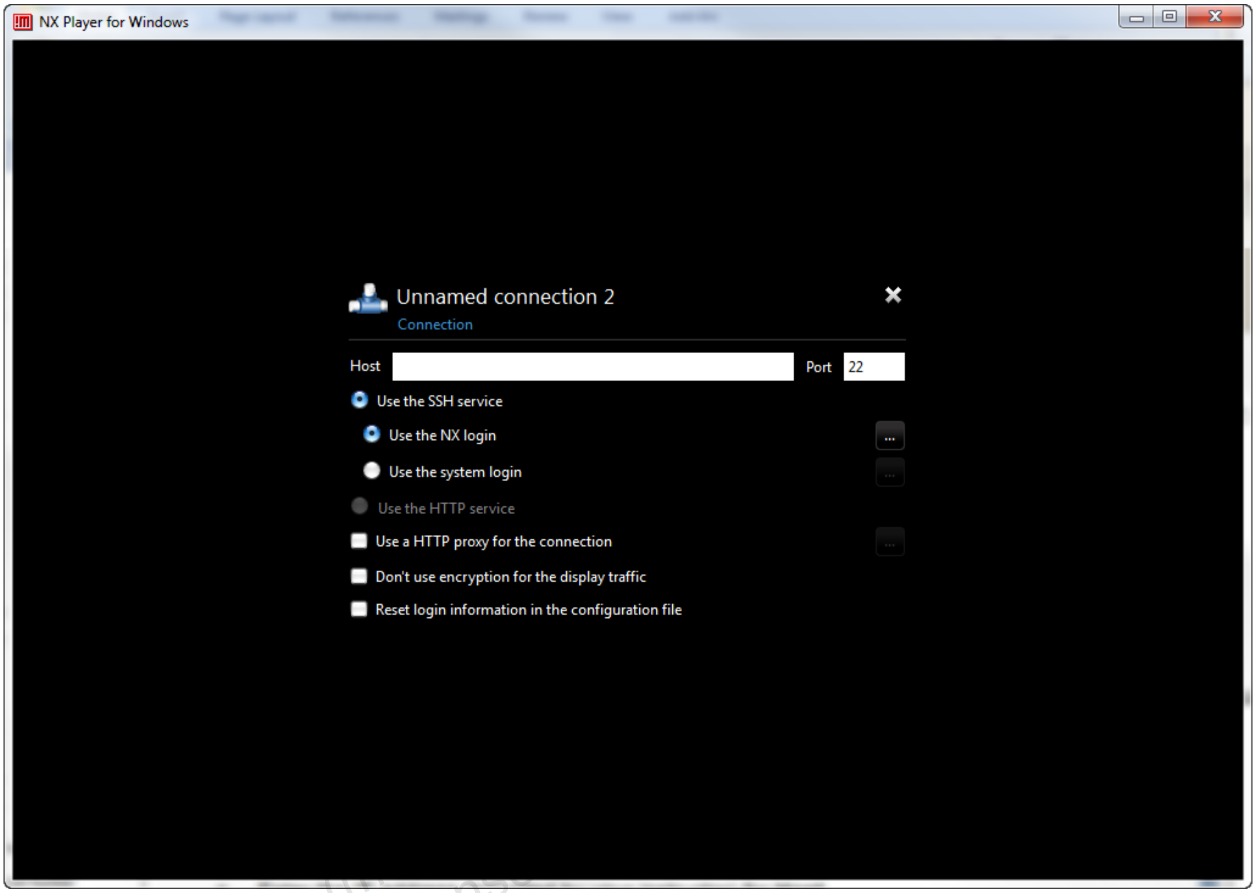
This appendix discusses accessing your student PC (**dom0**) remotely using NX Player. The NX Player in this appendix is NX Player for Windows, Preview 5, version 4.0.132.

Steps

1. Install **NX Player** (if necessary) from http://www.nomachine.com/download.php.
2. Run **NX Player** (for example, select **NX Player for Windows** from the Windows Start menu).
   a. Ensure that the **Favorites** tab is selected.
      - The Connect window appears.



b. Click **New connection** to display the following window.



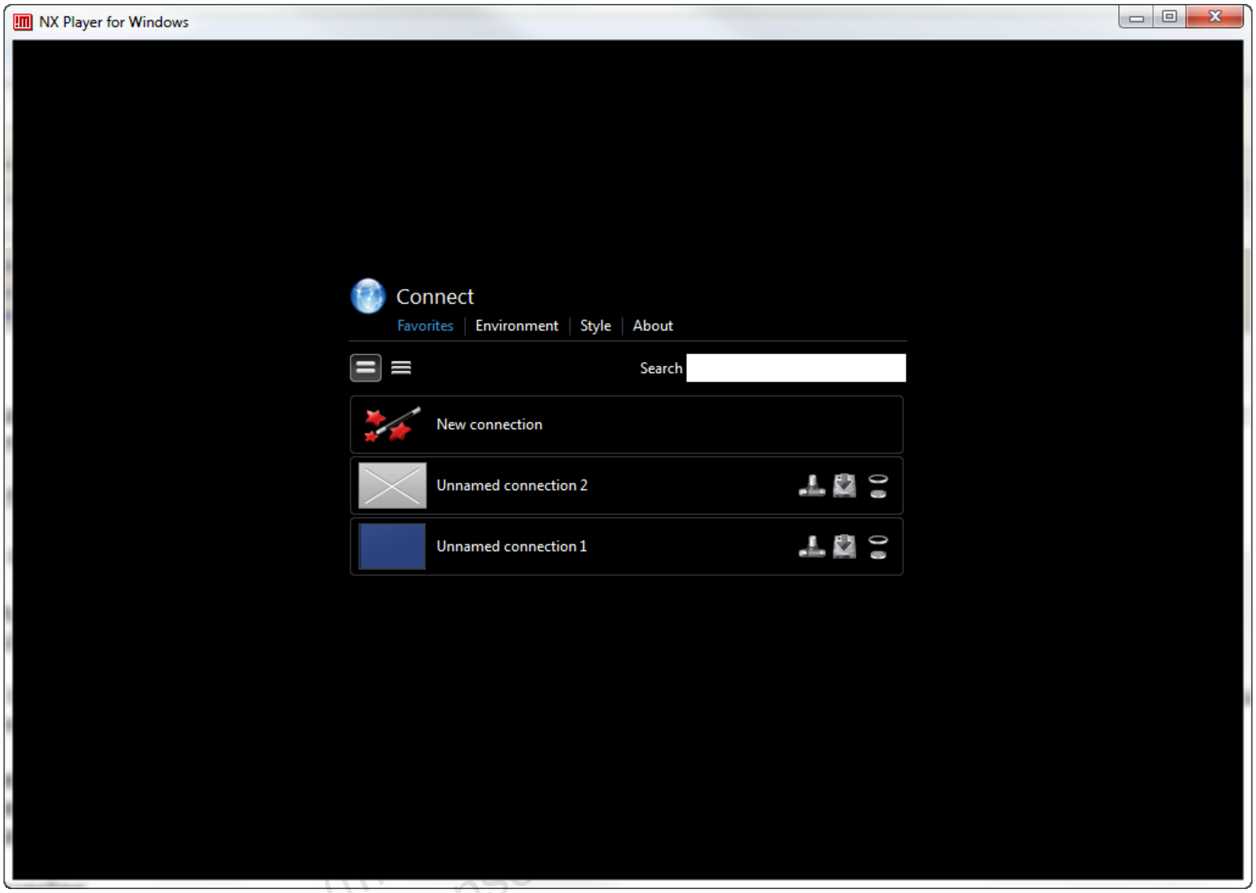
c. Enter the IP address (provided by your instructor) for **Host**.

d. Accept the defaults:

1) Port 22
2) Use the SSH service
3) Use the NX login

e. Note the connection name. In this example it is **Unnamed connection 2**. Yours is most likely **Unnamed connect 1**.

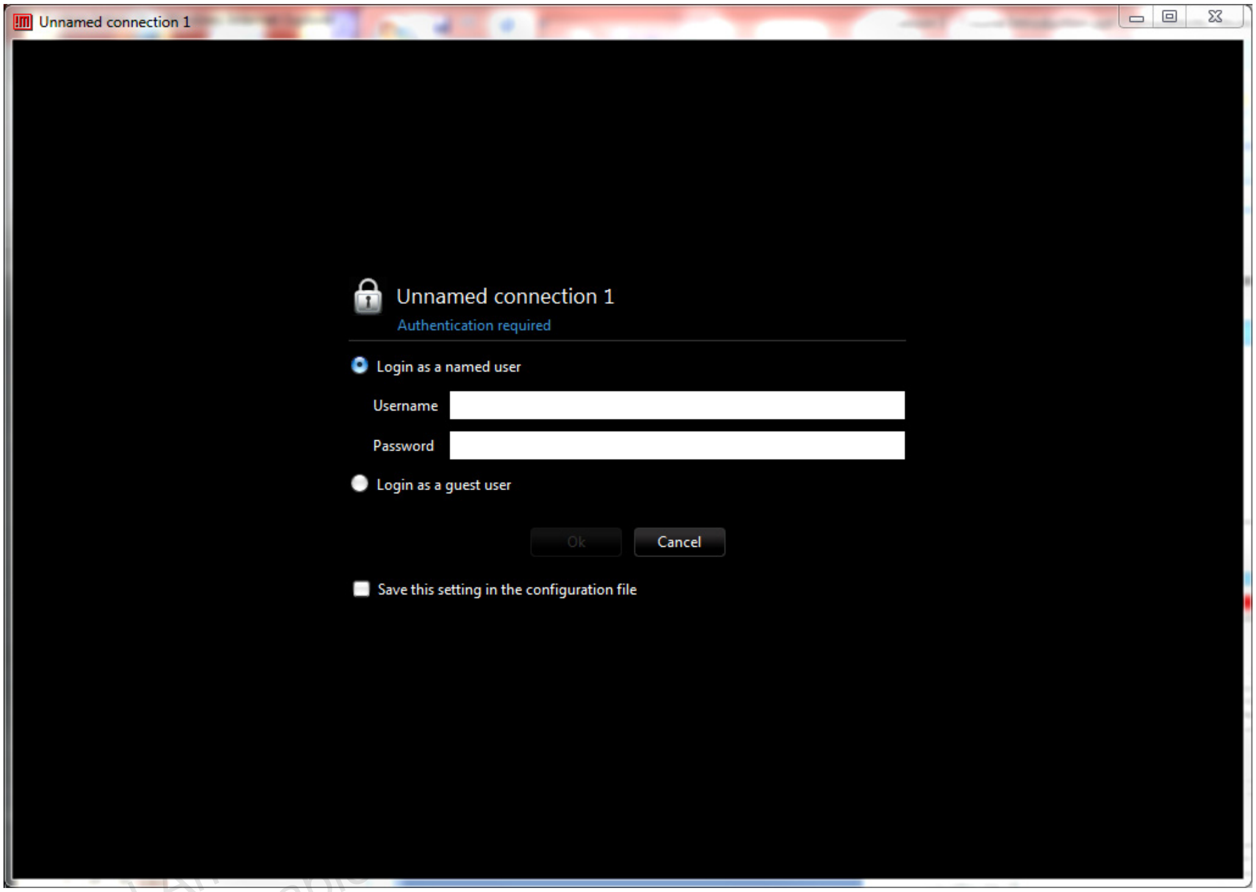
f. Press **Enter**. The following window appears.



g. Click the connection that you just created (**Unnamed connection 1**, for example).

- The Login window appears.

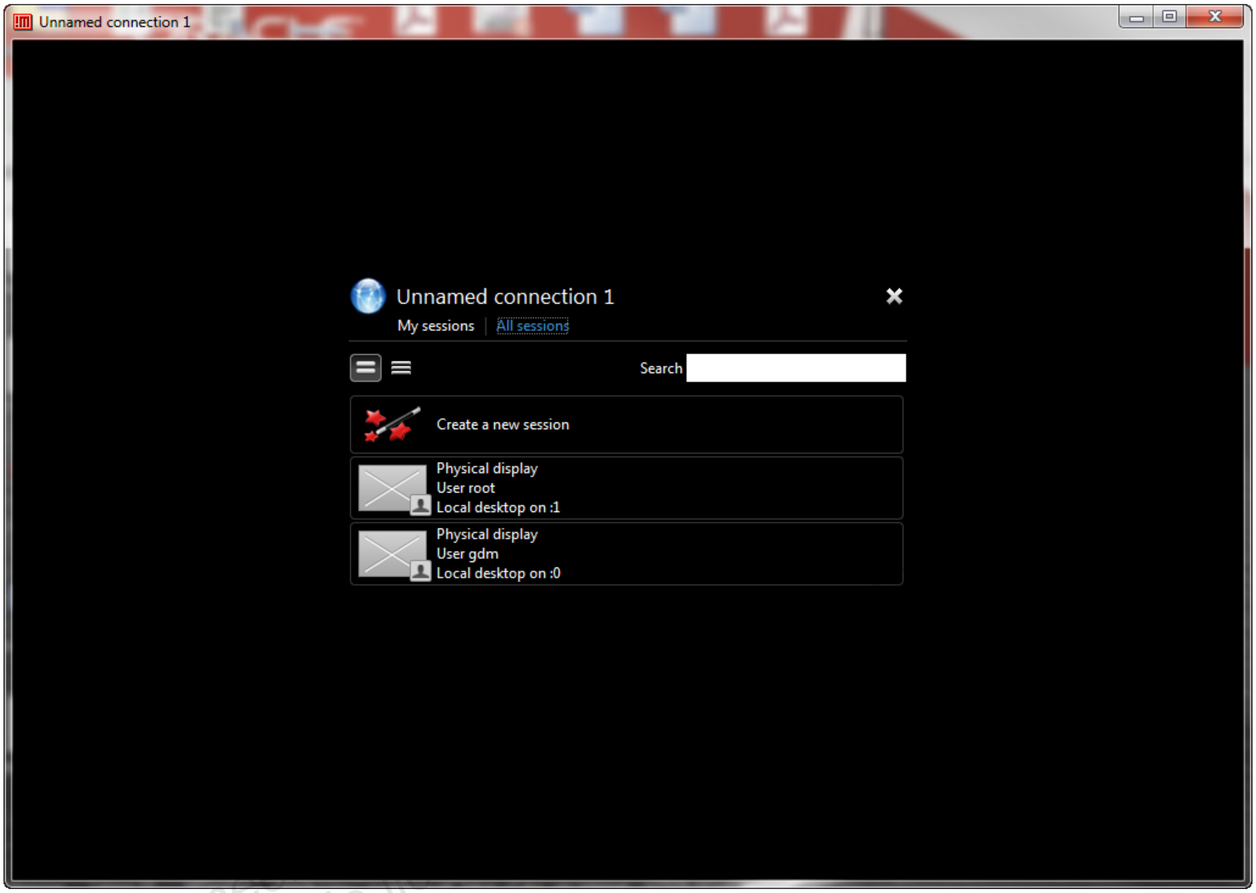
3. Log in.
   - The window shown in the following screenshot appears, prompting for login authentication.



   a. Ensure that **Login as a named user** is selected.
   b. For **Username**, enter `vncuser`.
   c. For **Password**, enter `vnctech`.
   d. Click **Ok**.

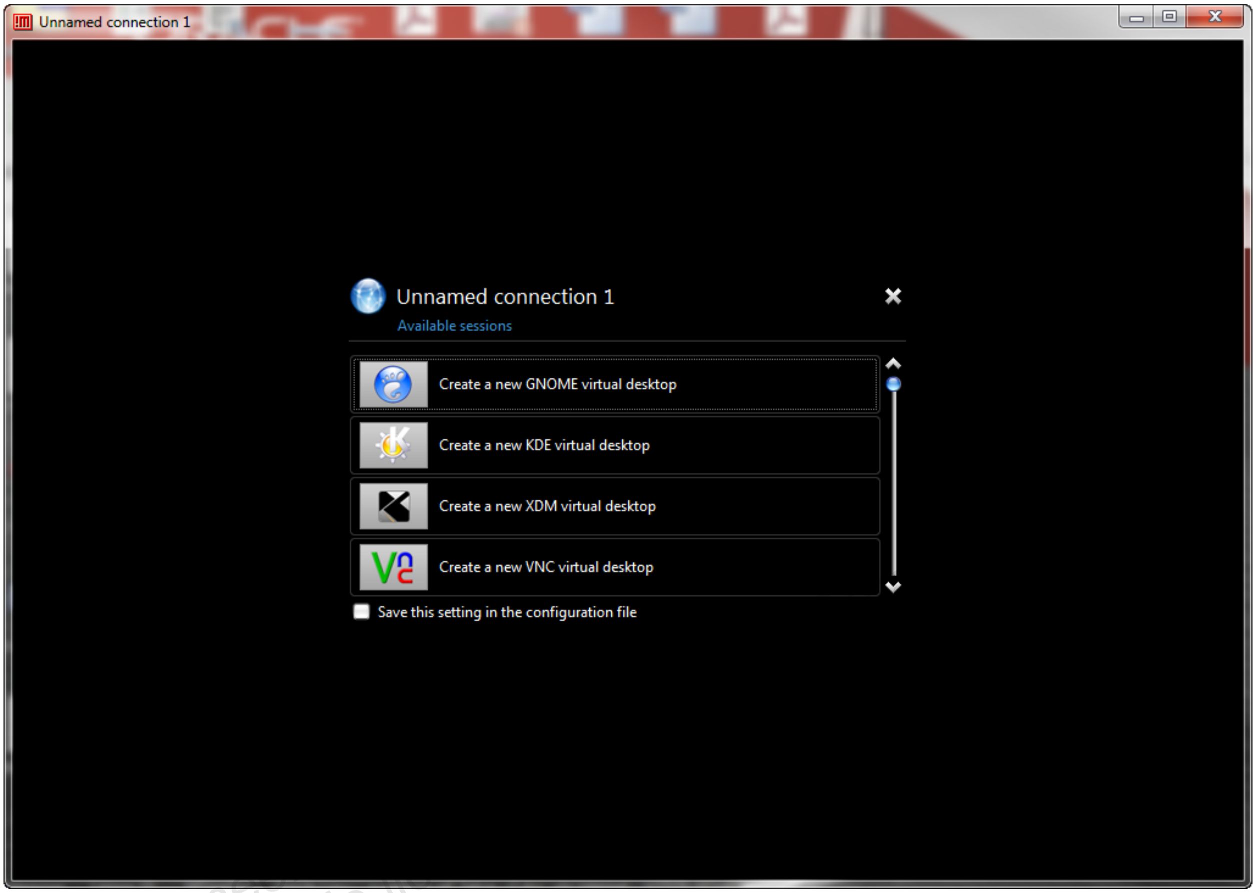
4. Create a new session.

- The window shown in the following screenshot appears.

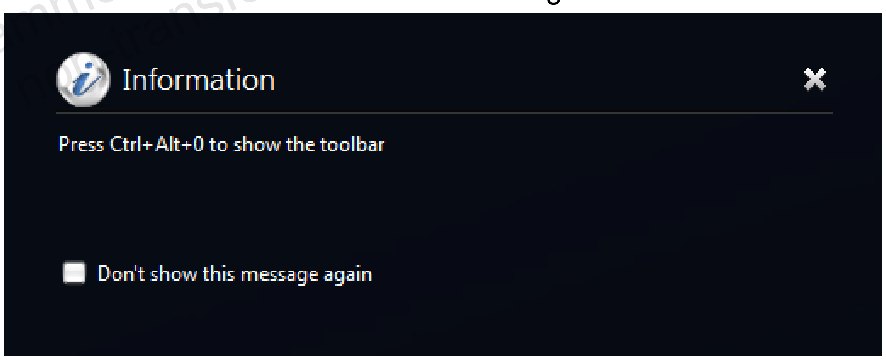


a. Click Create a new session.

5. Create a new GNOME virtual desktop.
   - The window shown in the following screenshot appears.



   a. Click Create a new GNOME virtual desktop.
   b. Click the **X** in the **Information** message box to close the box.



      – The **dom0** GNOME virtual desktop window appears.

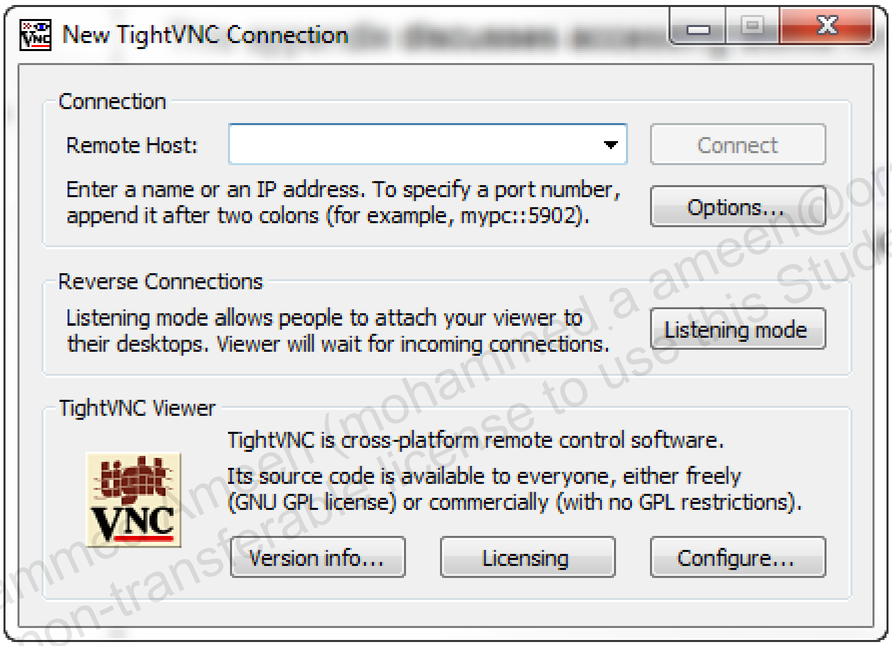
Appendix C: Using VNC (TightVNC) to Connect Directly to VM Guests

Overview

This appendix discusses accessing the VM guest systems that directly uses VNC (TightVNC). It is not recommended to connect to **dom0** or to the **host03** VM by using VNC. Both **dom0** and **host03** have the GNOME user interface, which causes various problems when connecting using VNC.

Steps

1. Install **tightvnc** (if necessary) from http://www.tightvnc.com/.
2. Run **TightVNC Viewer** (for example, select **TightVNC Viewer** from the Windows Start menu).
   - The following New TightVNC Connection window appears.



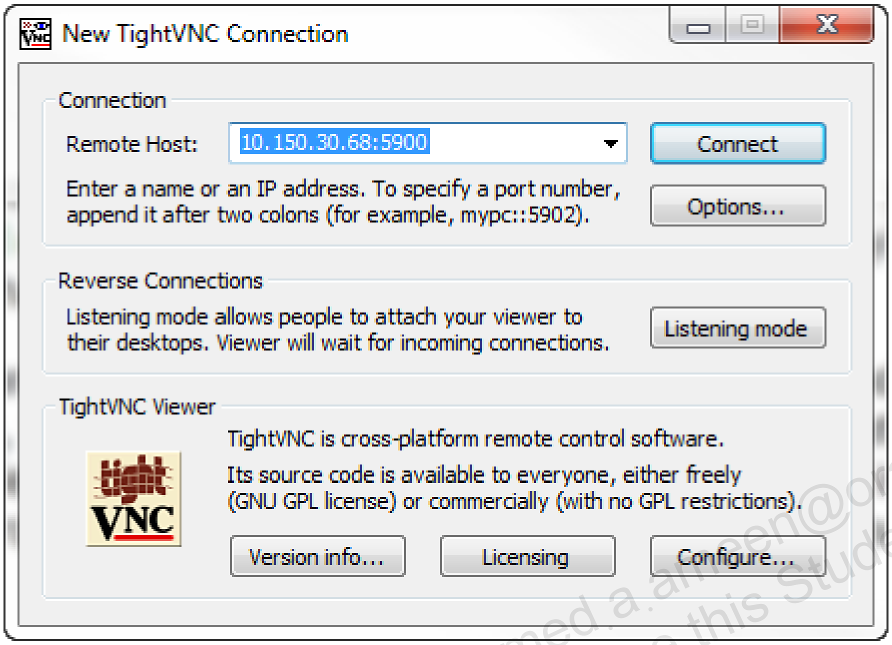
3. Connect directly to your **host01** virtual machine.
   - The following assumptions are made:
     – The **host01** VM was created first (has a port number of `5900`).
     – The **host02** VM was created next (has a port number of `5902`).
     – The **host03** VM was created last (has a port number of `5903`).
   - The output of the following commands (from **dom0** as `root`) indicates that this assumption is true.

```
# xm list -l host01 | grep location
          (location 0.0.0.0:5900)
# xm list -l host02 | grep location
          (location 0.0.0.0:5902)
# xm list -l host03 | grep location
          (location 0.0.0.0:5903)
```

- Enter the IP address (provided by your instructor), followed by the port number to connect directly a VM guest.

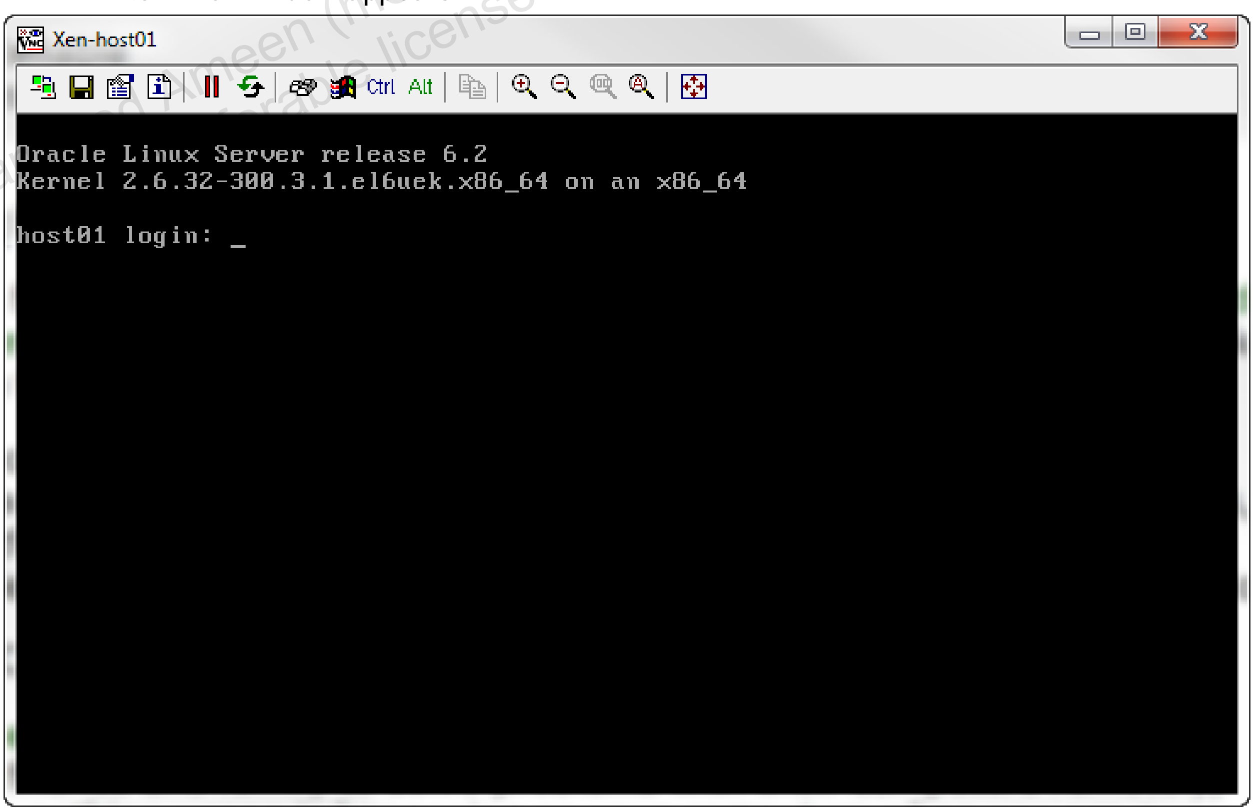
a. To connect directly to the **host01** VM, enter the following.

- In this example, the IP address of your student PC is 10.150.30.68. Your IP address is different.



b. Click **Connect**.

- A terminal window appears.



c. Log in as `root` password with `0racle` (leading zero, not letter O).

d. Enter the `hostname` command to confirm that you are logged in to **host01**.

```
# hostname
host01.example.com
```

e. Log off by entering the `exit` command.

f. Close the VNC window by clicking the **X** in the top-right corner of the window.

Appendix D: Using NoMachine Version 4 to Connect to dom0

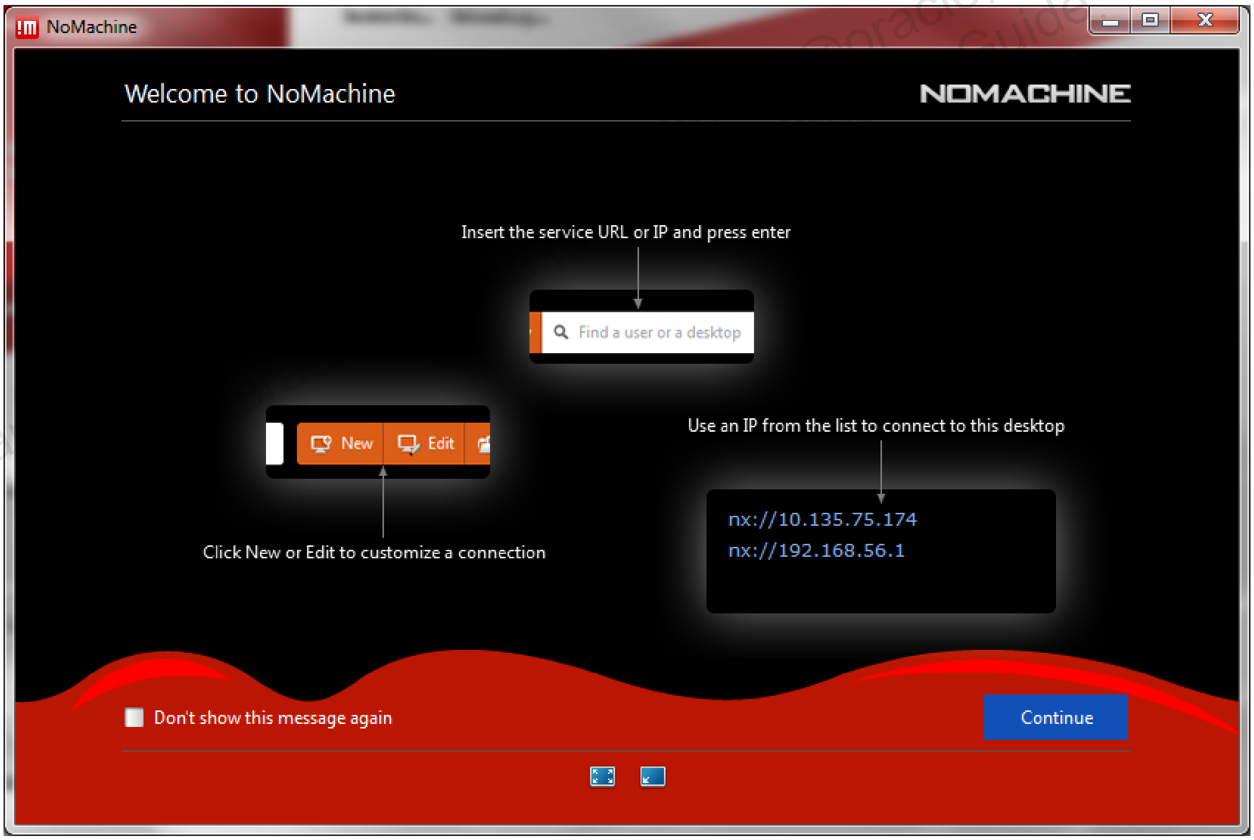
Overview

This appendix describes the procedure to access your lab machine remotely by using the NoMachine client for Windows. This procedure assumes that you have downloaded and installed the NoMachine client from the following location: http://www.nomachine.com/download.php.

Note: If you are accessing your lab environment remotely, you have received instructions on how to access your lab machine. The following steps summarize the configuration and connection tasks when using the NoMachine client for Windows 7.

Tasks

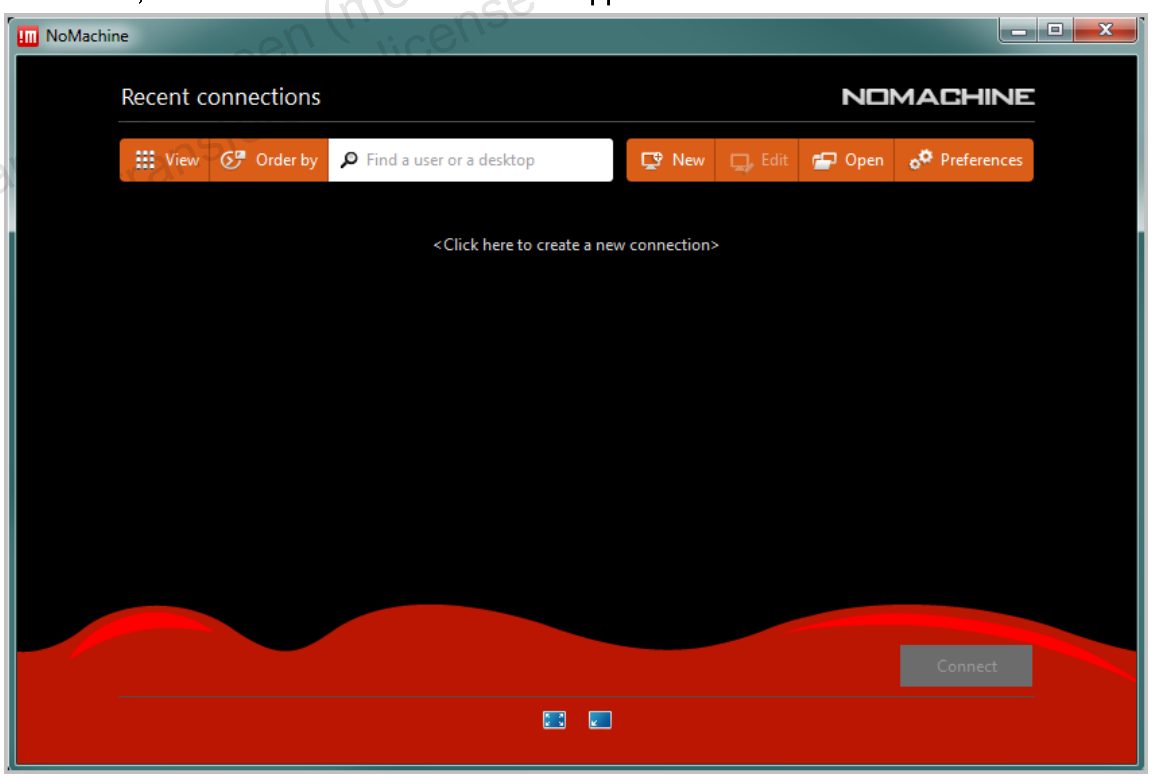
1. Create a session to your assigned lab machine by using the NoMachine Connection Wizard.
   a. Select the NoMachine program from the Windows Start menu.
   The Welcome to NoMachine window appears.



   b. Click Continue.

If this is the first time you are using NoMachine, the following window appears:

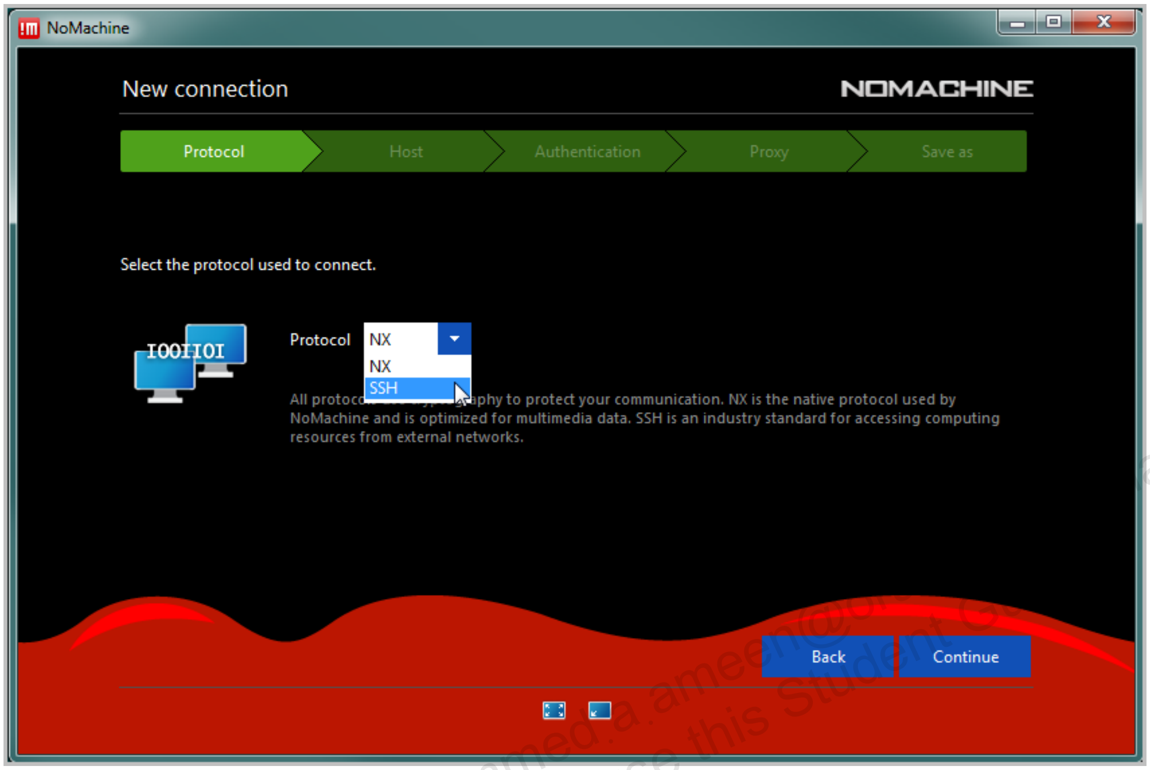
Otherwise, the Recent connections window appears.



c. Click “New” to add a new connection.

The Protocol window of the New Connection Wizard appears.

d. Select SSH from the Protocol drop-down list.



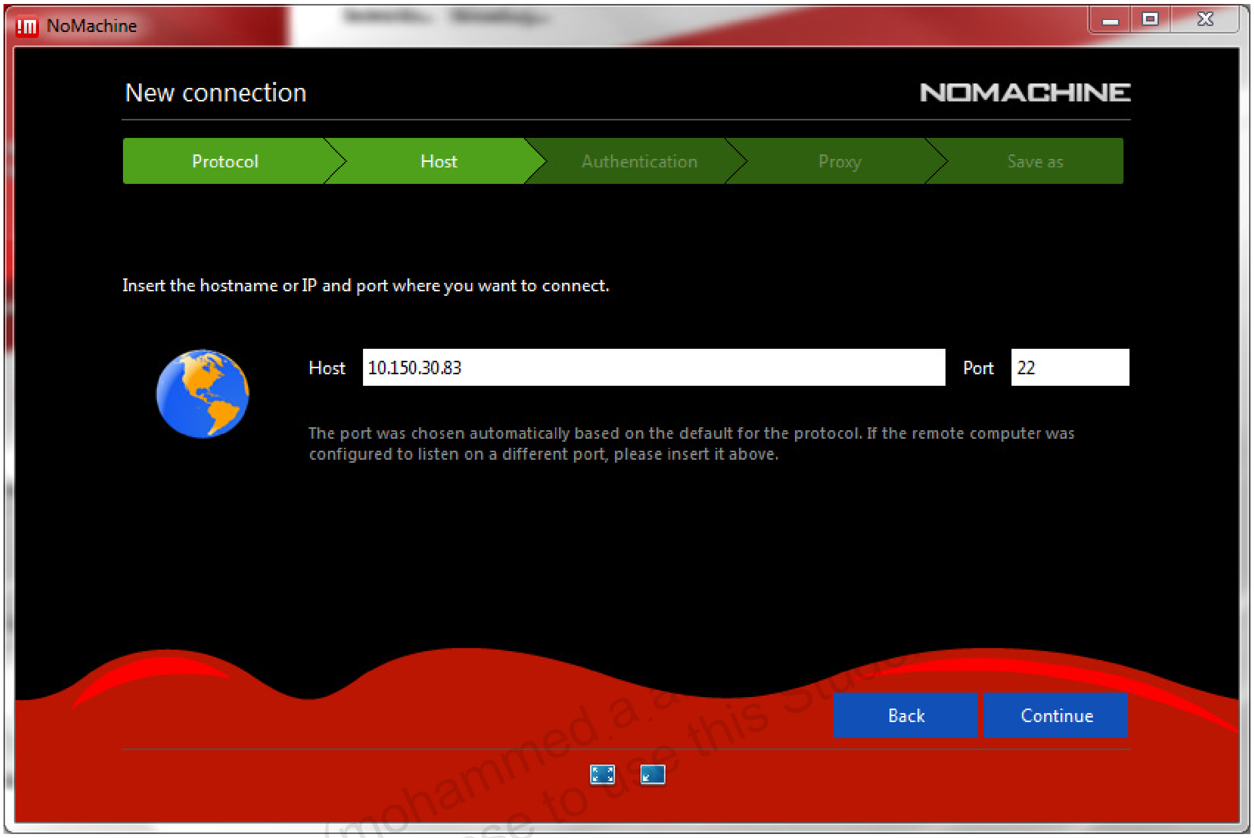
e. Click the Continue button.

The Host window appears.

f. In the Host field, enter the IP address that was provided to you by your instructor.

In this example, the `10.150.30.83` IP address is used.

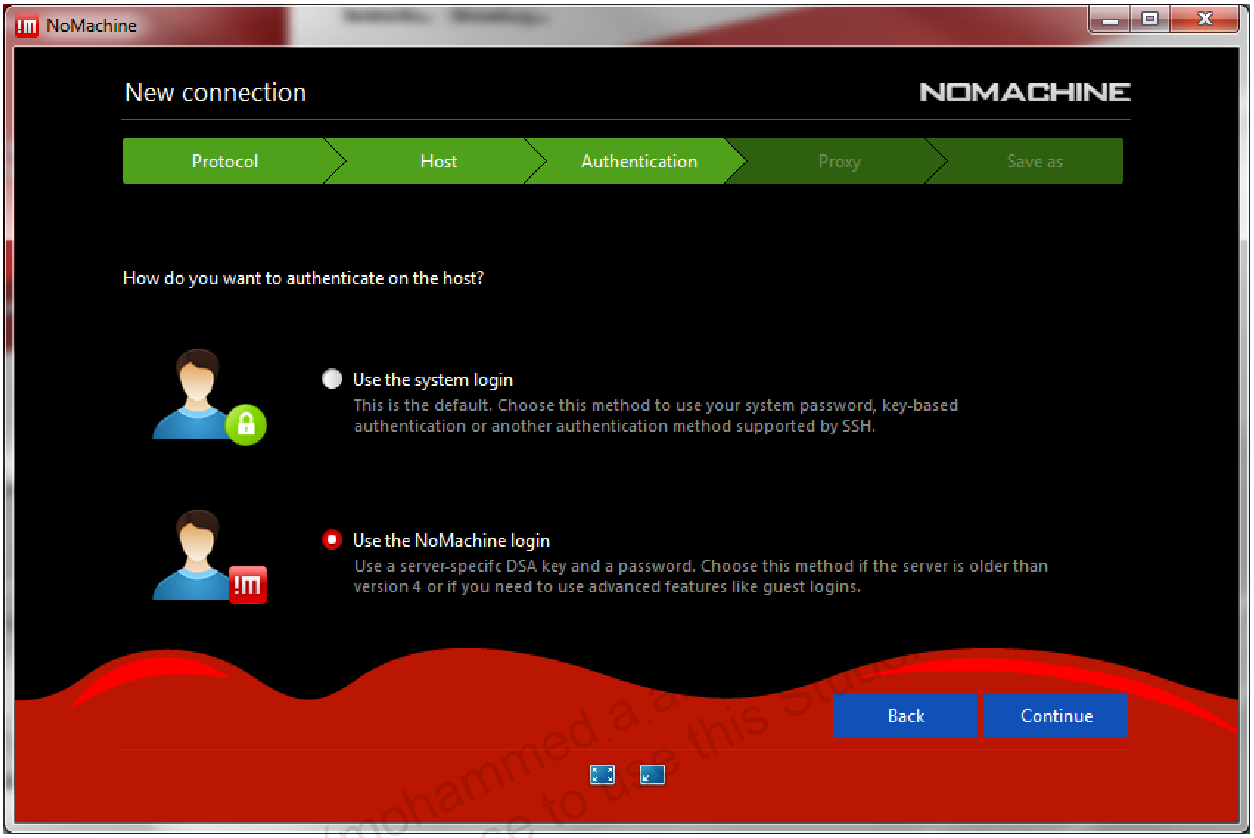
Accept 22 for the port number.



g. Click Continue.

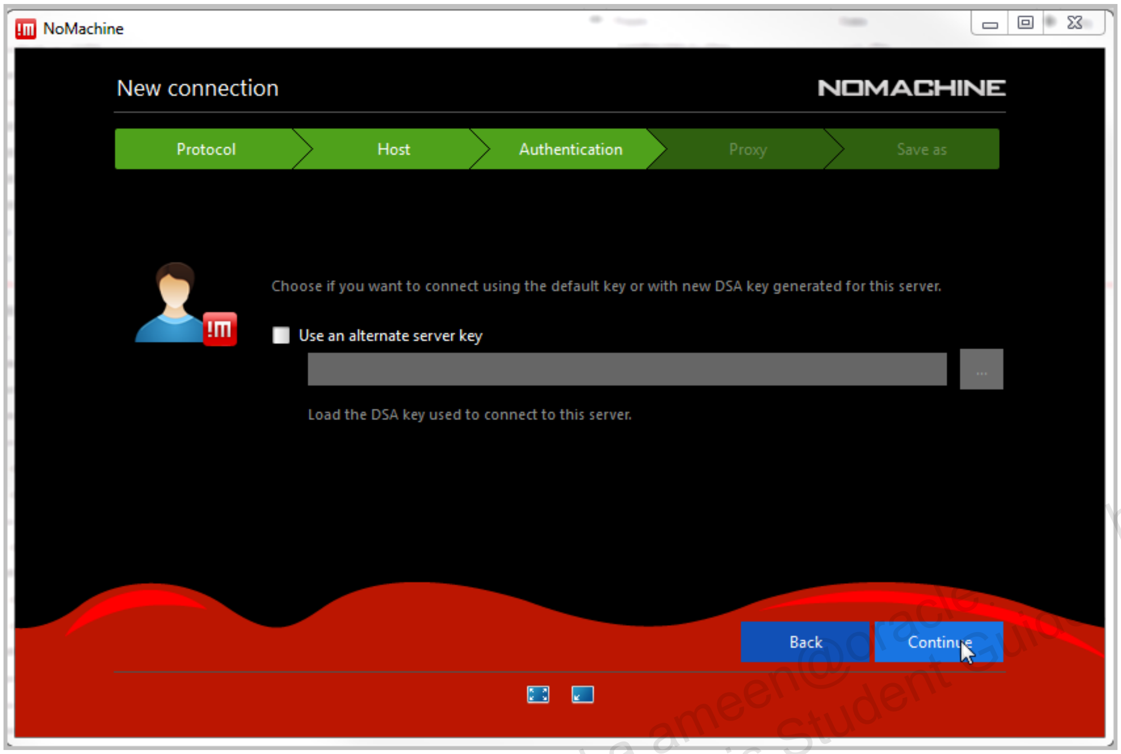
The Authentication window appears.

h. Select the “Use the NoMachine login” radio button.

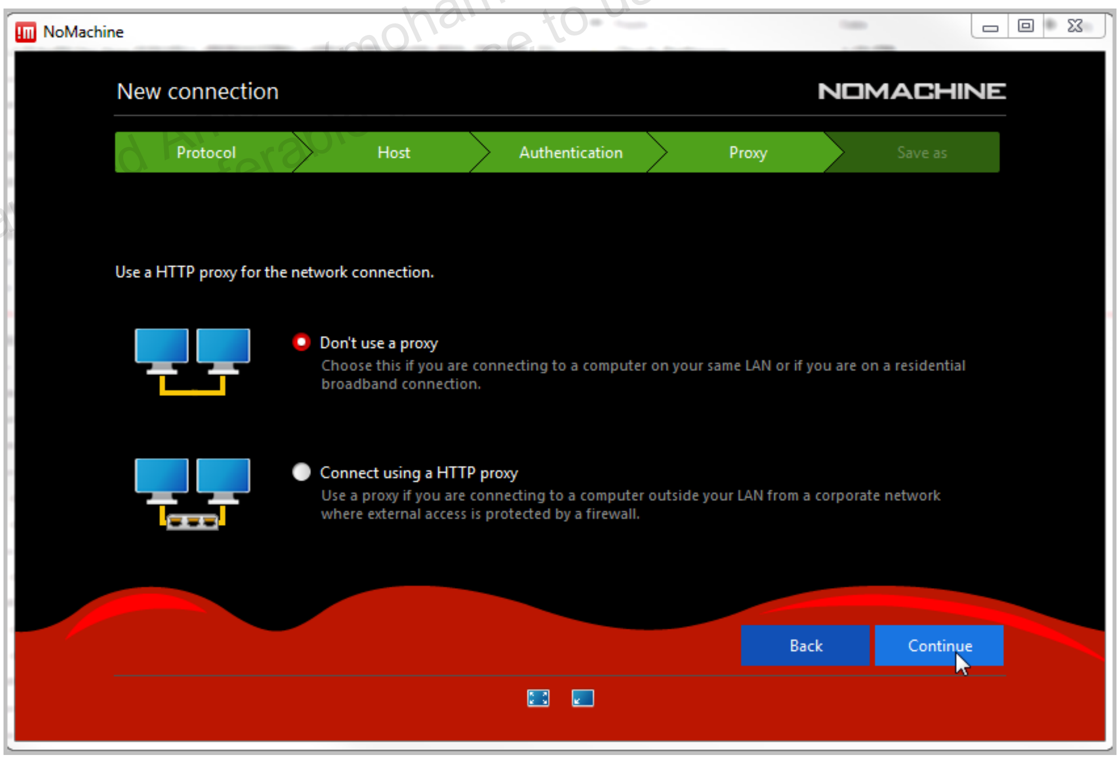


i. Click Continue.

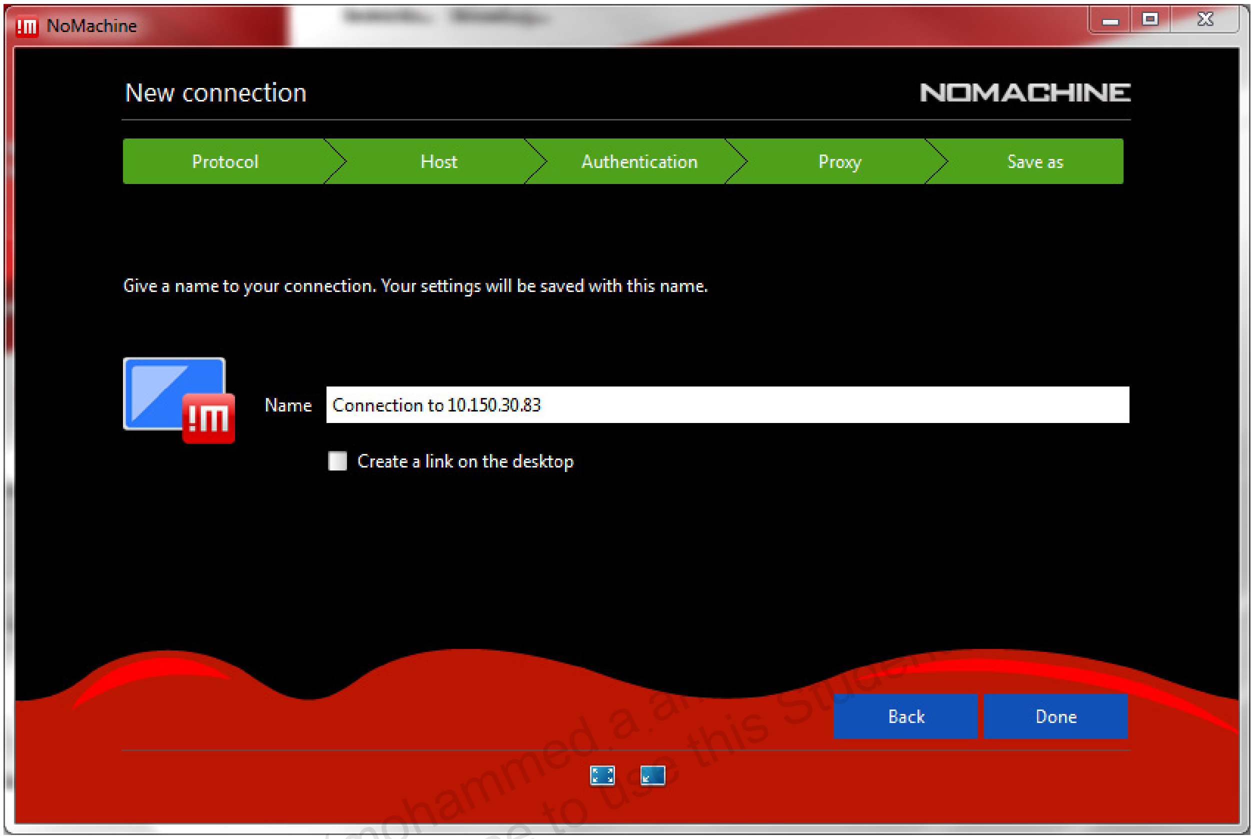
j. Leave the "Use an alternate server key" check box unselected and click Continue.



k. In the Proxy window, leave the "Don't use a proxy" radio button selected and click Continue.

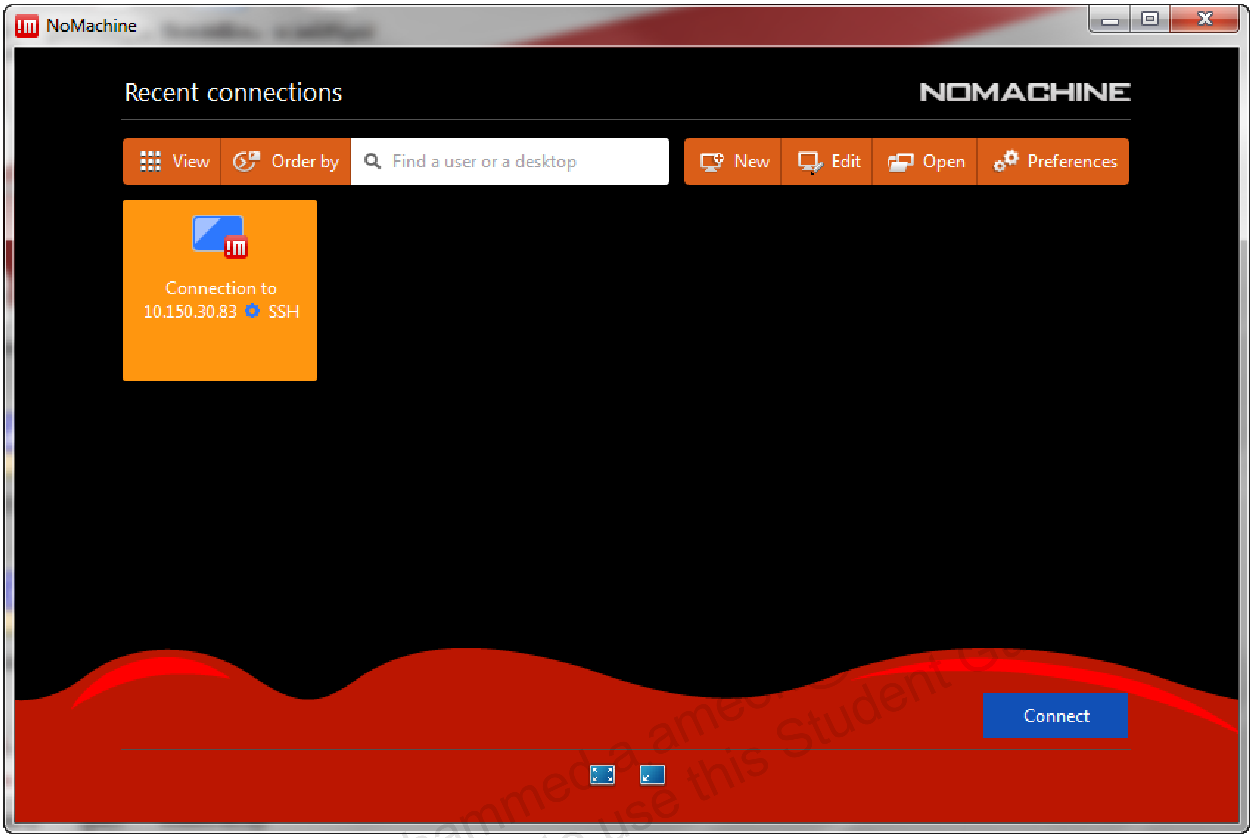


l. In the “Save as” window, enter a name for your connection or accept the proposed name.



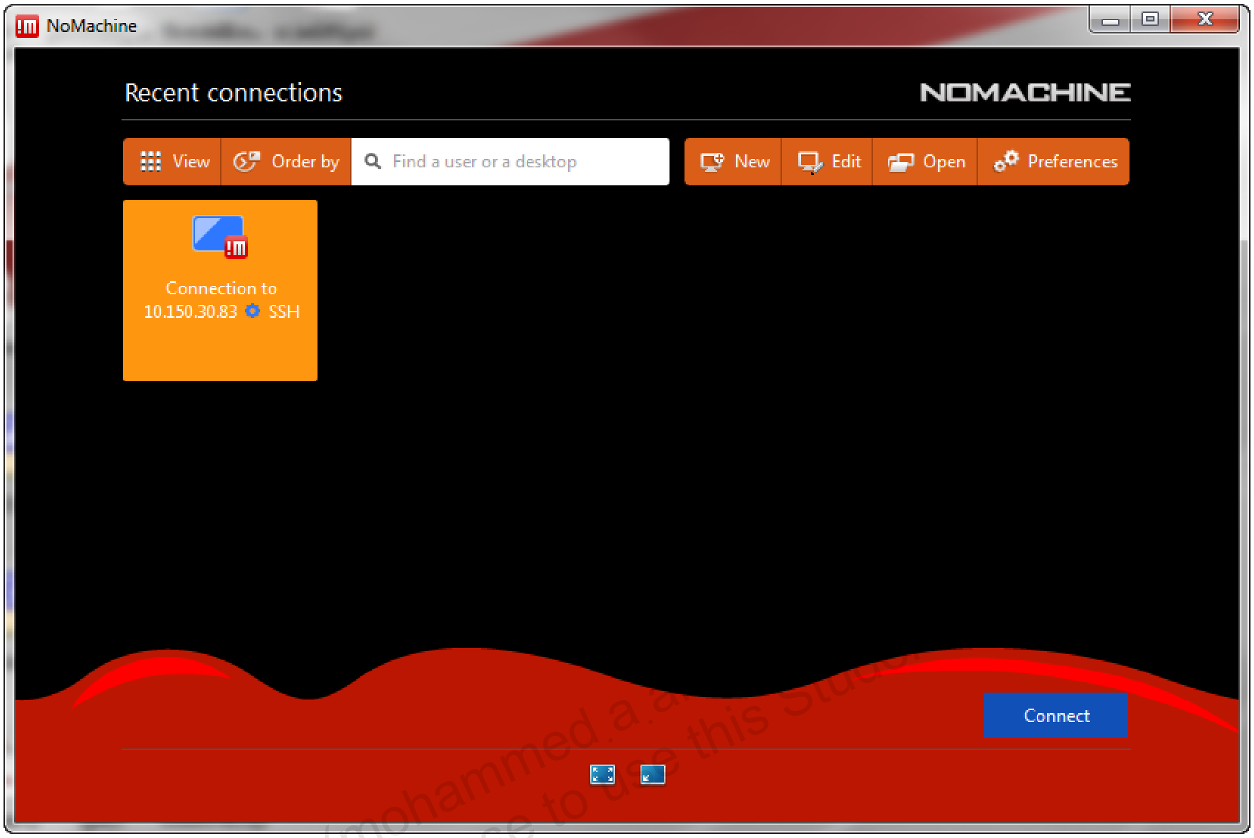
m. Click Done.

Your new connection appears in the Recent connections window.

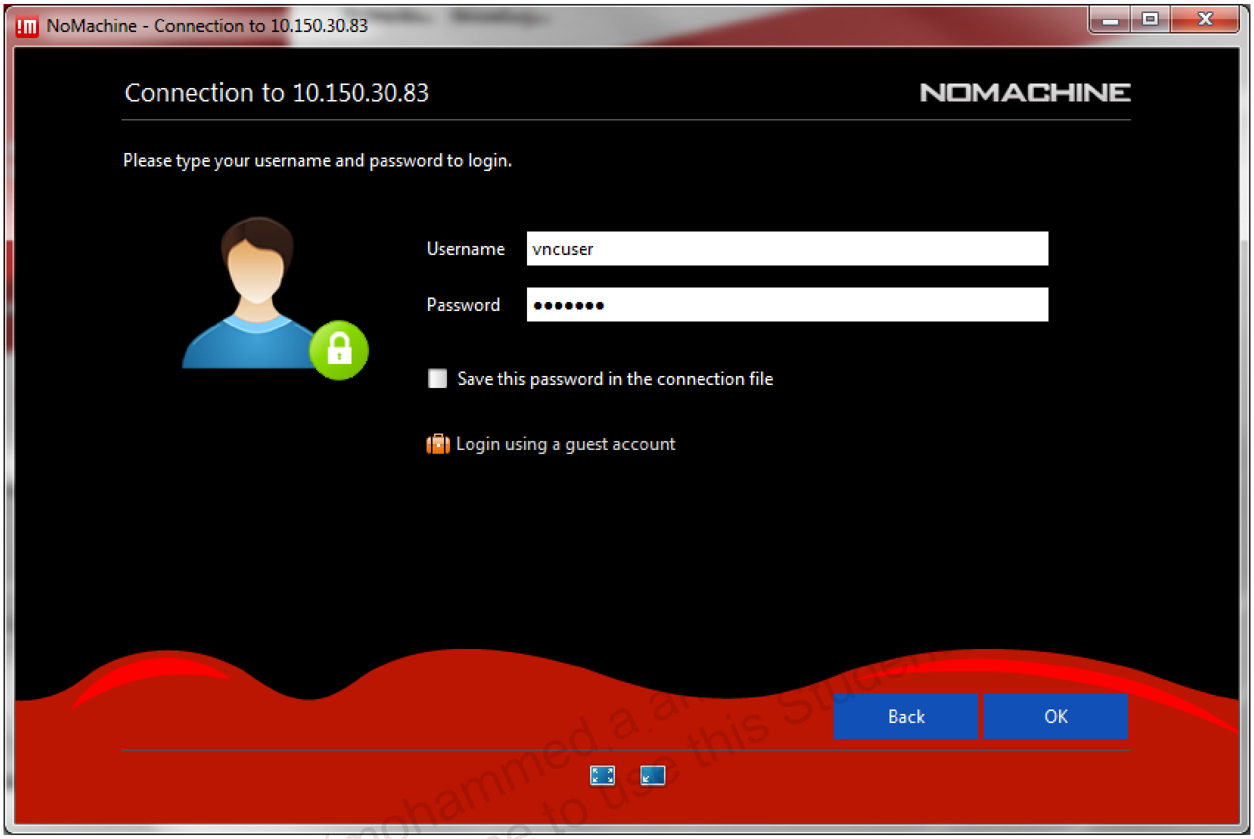


2. Access your lab machine by using your newly created connection.

   a. In the Recent connections window, highlight your new connection and click Connect.



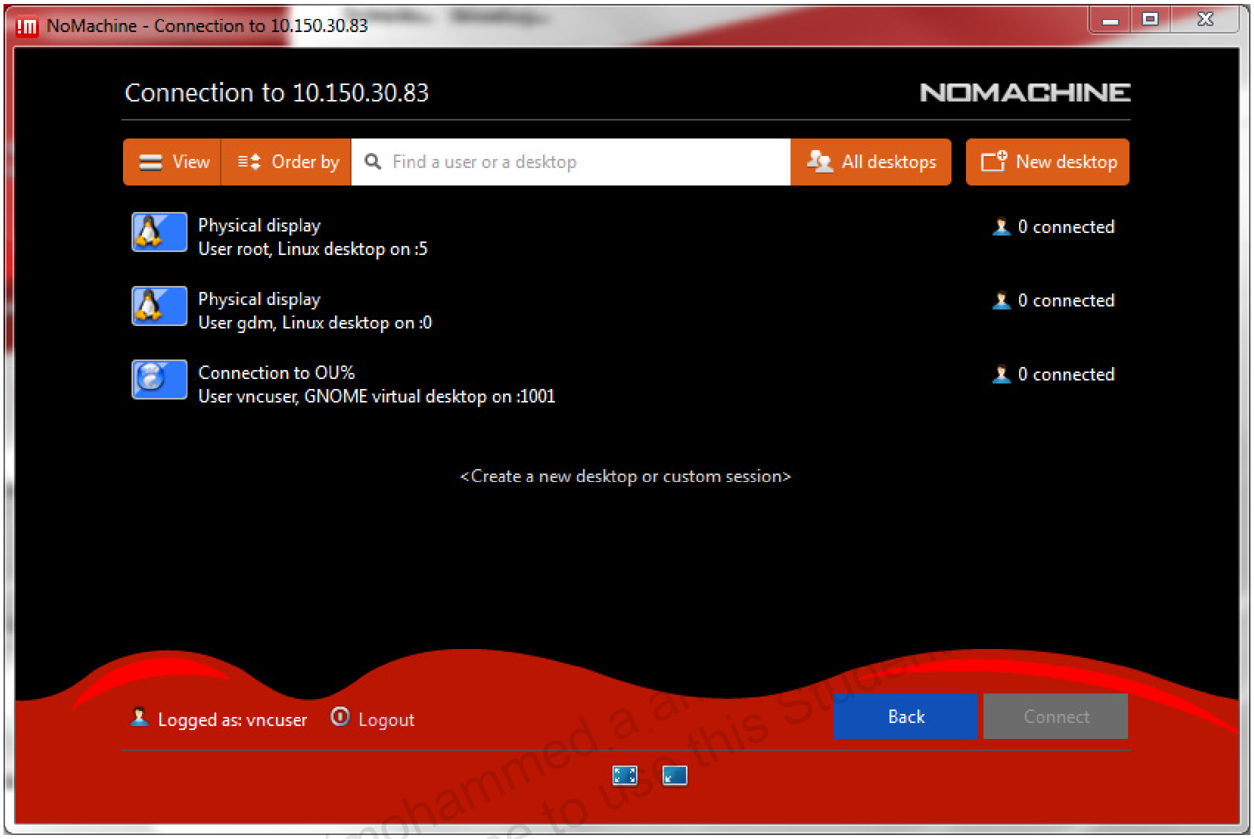
b. On your new connection login screen, enter the login credentials provided by your instructor.



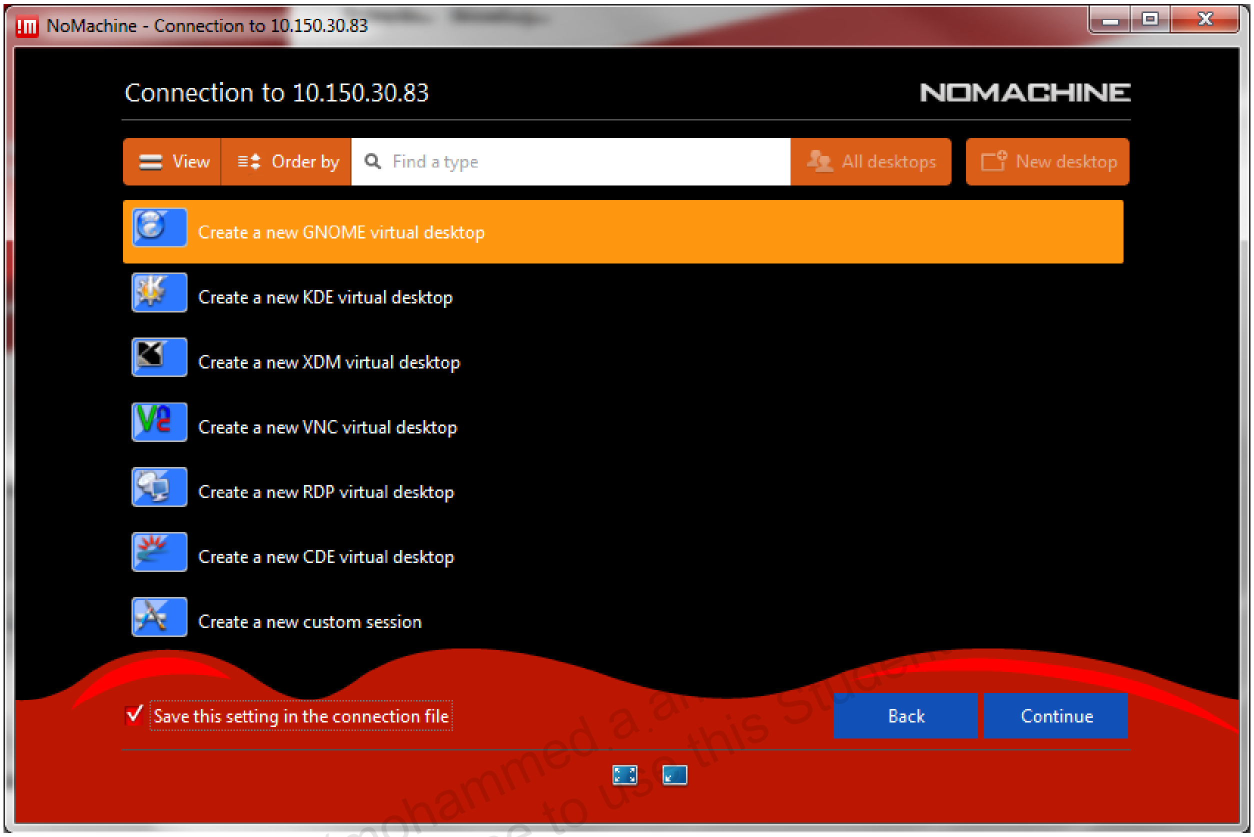
Optionally, you can select the “Save this password in the configuration file” check box.

c. Click OK.

d. In the next window, click the “New desktop” session link located in the top-right area of the screen.



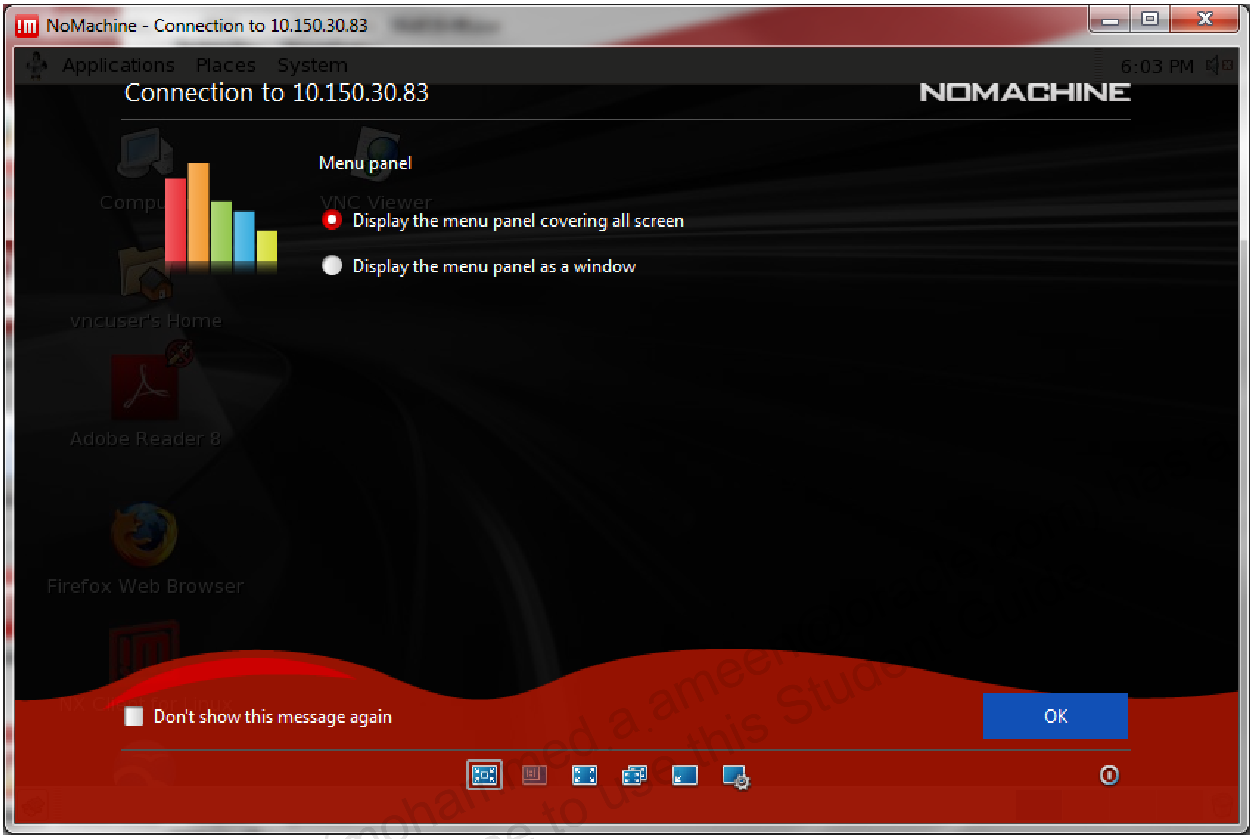
e. Select “Create a new GNOME virtual desktop” and check the “Save this setting in the configuration file” selection box.



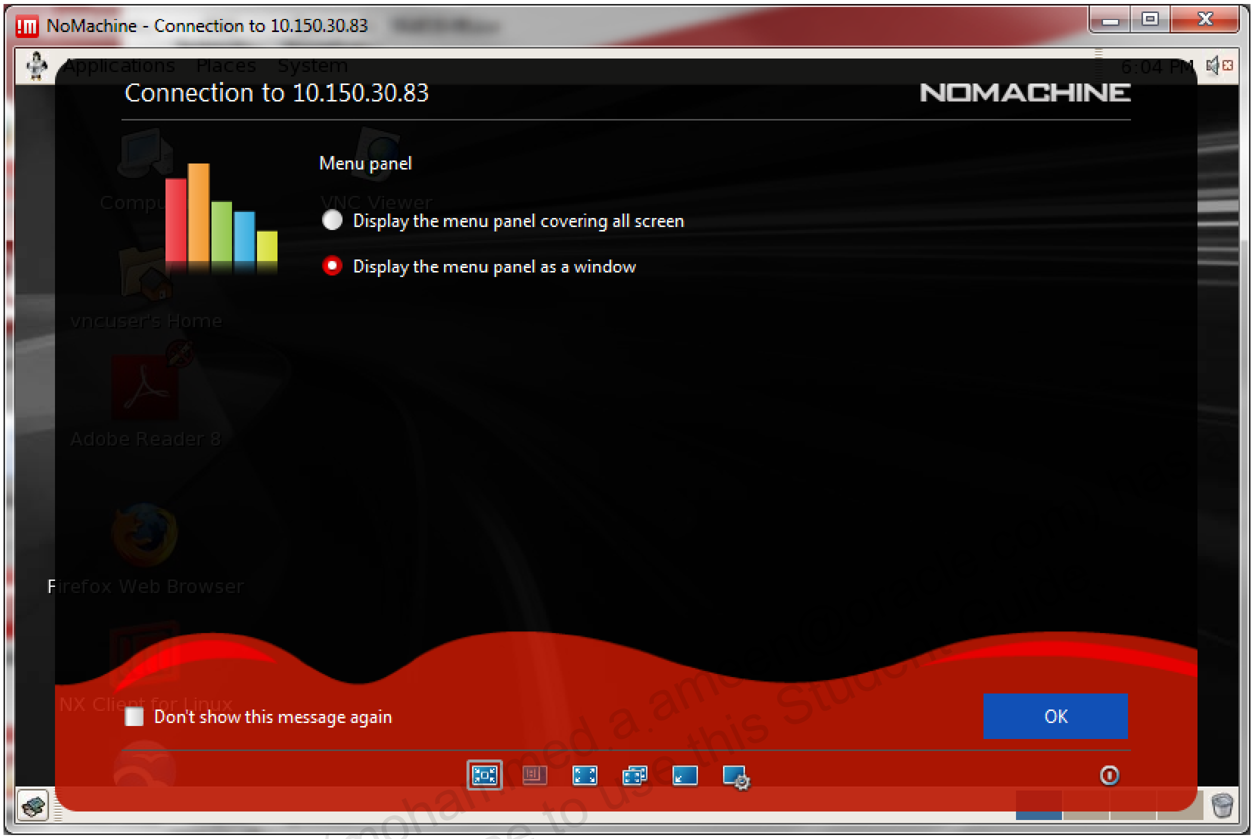
f. Click Continue.

g. Click OK to dismiss the next two windows that provide tips for navigation. Optionally you can select the “Don’t show this message again” check box on both screens.

h. The Menu panel options window appears. You can accept the default for the Menu panel as shown in the following screenshot:



i. Or select the “Display the menu panel as a window” option to display the Menu panel as a centered window.



j. Click OK.

You are now connected to your lab machine.