



# Red Hat Training and Certification

## Instructor Guide

Red Hat Enterprise Linux 8.2 RH124

## Red Hat System Administration I

Edition 1





# Red Hat System Administration I

# **Red Hat Enterprise Linux 8.2 RH124**

## **Red Hat System Administration I**

### **Edition 1 20200928**

### **Publication date 20200928**

Authors: Fiona Allen, Victor Costea, Snehangshu Karmakar, Marc Kesler,  
Ed Parenti, Saumik Paul, Hervé Quatremain, Dallas Spohn  
Editor: Steven Bonneville, Ralph Rodriguez, David Sacco, Nicole Muller, Heather  
Charles, David O'Brien, Seth Kenlon

Copyright © 2020 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2020 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed, please send email to [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Red Hat logo, JBoss, Hibernate, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

The OpenStack® word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission. Red Hat, Inc. is not affiliated with, endorsed by, or sponsored by the OpenStack Foundation or the OpenStack community.

All other trademarks are the property of their respective owners.

Contributors: Artur Glogowski, Latha Murthy, Samik Sanyal, Chetan Tiwary, Achyut Madhusudan, Rudolf Kastl, Rob Locke, Michael Phillips

<b>Document Conventions</b>	<b>vii</b>
<b>Course Timing</b>	<b>ix</b>
ILT/VT .....	
<b>Introduction</b>	<b>xi</b>
Chapter Information .....	xi
Course Introduction .....	xii
Orientation to the Classroom Network .....	xiii
: Internationalization .....	xv
<b>1. Getting Started with Red Hat Enterprise Linux</b>	<b>1</b>
Chapter Information .....	
Objectives .....	2
Key Takeaways .....	2
Instructor Tips and Suggestions .....	2
<b>2. Accessing the Command Line</b>	<b>3</b>
Chapter Information .....	
Objectives .....	4
Key Takeaways .....	4
Instructor Tips and Suggestions .....	4
<b>3. Managing Files From the Command Line</b>	<b>7</b>
Chapter Information .....	
Objectives .....	8
Key Takeaways .....	8
Instructor Tips and Suggestions .....	8
<b>4. Getting Help in Red Hat Enterprise Linux</b>	<b>11</b>
Chapter Information .....	
Objectives .....	12
Key Takeaways .....	12
Instructor Tips and Suggestions .....	12
<b>5. Creating, Viewing, and Editing Text Files</b>	<b>15</b>
Chapter Information .....	
Objectives .....	16
Key Takeaways .....	16
Instructor Tips and Suggestions .....	16
<b>6. Managing Local Users and Groups</b>	<b>19</b>
Chapter Information .....	
Objectives .....	20
Key Takeaways .....	20
Instructor Tips and Suggestions .....	20
<b>7. Controlling Access to Files</b>	<b>23</b>
Chapter Information .....	
Objectives .....	24
Key Takeaways .....	24
Instructor Tips and Suggestions .....	24
<b>8. Monitoring and Managing Linux Processes</b>	<b>27</b>
Chapter Information .....	
Objectives .....	28
Key Takeaways .....	28
Instructor Tips and Suggestions .....	28
<b>9. Controlling Services and Daemons</b>	<b>31</b>

Chapter Information .....	
Objectives .....	32
Key Takeaways .....	32
Instructor Tips and Suggestions .....	32
<b>10. Configuring and Securing SSH</b>	<b>35</b>
Chapter Information .....	
Objectives .....	36
Key Takeaways .....	36
Instructor Tips and Suggestions .....	36
<b>11. Analyzing and Storing Logs</b>	<b>39</b>
Chapter Information .....	
Objectives .....	40
Key Takeaways .....	40
Instructor Tips and Suggestions .....	40
<b>12. Managing Networking</b>	<b>43</b>
Chapter Information .....	
Objectives .....	44
Key Takeaways .....	44
Instructor Tips and Suggestions .....	44
<b>13. Archiving and Transferring Files</b>	<b>47</b>
Chapter Information .....	
Objectives .....	48
Key Takeaways .....	48
Instructor Tips and Suggestions .....	48
<b>14. Installing and Updating Software Packages</b>	<b>51</b>
Chapter Information .....	
Objectives .....	52
Key Takeaways .....	52
Instructor Tips and Suggestions .....	52
<b>15. Accessing Linux File Systems</b>	<b>55</b>
Chapter Information .....	
Objectives .....	56
Key Takeaways .....	56
Instructor Tips and Suggestions .....	56
<b>16. Analyzing Servers and Getting Support</b>	<b>59</b>
Chapter Information .....	
Objectives .....	60
Key Takeaways .....	60
Instructor Tips and Suggestions .....	60
<b>17. Comprehensive Review</b>	<b>63</b>
Chapter Information .....	
Objectives .....	64
Instructor Tips and Suggestions .....	64

# Document Conventions

---



## References

"References" describe where to find external documentation relevant to a subject.



## Note

"Notes" are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



## Important

"Important" boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled "Important" will not cause data loss, but may cause irritation and frustration.



## Warning

"Warnings" should not be ignored. Ignoring warnings will most likely cause data loss.





# Course Timing

## ILT/VT

### Day 1

Activity	Time
Introduction	20 minutes (40 minutes with i18n)
Chapter 1	20 minutes
Chapter 2	70 minutes
Chapter 3	140 minutes
Chapter 4	55 minutes
Chapter 5 (started)	50 minutes
Total for day	355 minutes (375 minutes)

### Day 2

Activity	Time
Chapter 5 (continued)	40 minutes
Chapter 6	120 minutes
Chapter 7	90 minutes
Chapter 8	130 minutes
Total for day	380 minutes

### Day 3

Activity	Time
Chapter 9	70 minutes
Chapter 10	110 minutes
Chapter 11	150 minutes
Total for day	330 minutes

**Day 4**

Activity	Time
Chapter 12	125 minutes
Chapter 13	115 minutes
Chapter 14	110 minutes
Total for day	350 minutes

**Day 5**

Activity	Time
Chapter 15	100 minutes
Chapter 16	75 minutes
Chapter 17	170 minutes
Total for day	345 minutes

# Introduction

## Chapter Information Overview

Cover introductory material for this class.

### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
1	Course Objectives and Structure	P: Lecture	10
2	Orientation to the Classroom Network	P: Lecture	10
3	Internationalization	P: Lecture	20

Total Time: 20 minutes (40 minutes with Internationalization)

# Course Introduction

---

## Objectives

- Welcome students and provide an orientation to the class, classroom hardware, and facility or VT environment.

## Student Benefits

- Prepares students for this class.

## Presentation Notes

Introduce yourself and welcome students to the class. Before starting make sure any operational requirements, including taking attendance and providing students with materials, have been met. For an in-person training event, orient students to the facility. Make sure students know the classroom hours and plans for any rest breaks and lunch.

Discuss the basic structure of the course and course timing with the students.

## Objectives

Introduce your students to the main objectives of this course.

## Audience/Prerequisites

Discuss the intended audience and prerequisites for this course.

# Orientation to the Classroom Network

## Objectives

- Orient students to their classroom hardware and how to access it for upcoming lab exercises.

## Presentation Notes

Discuss the classroom environment from a single student's perspective. Focus on the machines that a student will directly interact with. This course has four student machines in **lab.example.com**:

- **workstation**, which has a GNOME desktop environment and will be used for most of their work
- **servera** and **serverb** which are servers to be used during the guided exercises and labs.

Discuss the appropriate student guide material on how students start and access their machines in the ILT or VT classroom environment.



### Warning

Be sure to warn students that performing a reset with **rht-vmctl** (in ILT) or with the VT interface will cause the system to be reverted to its starting point and *all work they have saved on that system will be lost*.

## Instructor-only notes

The following notes are information provided for the instructor's reference and convenience.



### Important

In the VT environment, the instructor initially has the same virtual machines as the student. *A VT instructor will not initially have the **demo** command available in their environment.*

To enable demos, you must log in as **root** on your **workstation** machine and run the following commands:

```
[root@workstation ~]# curl -O http://materials/.instructor/rht-demo-setup
[root@workstation ~]# chmod 755 rht-demo-setup
[root@workstation ~]# ./rht-demo-setup
```

In both ILT and VT, if the student workstation is down or stopped, the other machines will not be able to see the classroom network or shared servers, because it is the NAT router for each student.

In ILT, if you need to access a particular student's machines over the network, you will need to have them use **ip** on **workstation** to find out and tell you what their address on the 172.25.252.0/24 network is so you can **ssh** to that. From there you can reach that student's other machines.



The main difference between the **content.example.com** and **materials.example.com** "servers" is that **content** is used for large software images and packages, while **materials** is used for code examples and smaller supporting files. In the current classroom implementation, both servers are aliases of **classroom.example.com**.

# Internationalization



## Note

Some regions run Red Hat training in classrooms which by policy should be localized for the language in which the course is being taught. The classroom setup process deploys all machines and users localized for **en-US.utf8** (US English).

If your classroom needs to be set to a different locale, this section must be covered. The instructor should guide students through appropriate language and input configuration for their locale and hardware. The instructor should then have students save the settings to their baseline machine images with **rht-vmctl**.

Per the directions in **ClassroomSetup.txt**, the locale settings for the physical layer (**foundationX**) are inherited from the manual selections made when **foundation0** was installed. Modifying the locale in the pre-built virtual machine images should be done as an exercise with the students updating and saving each of their virtual machines using **rht-vmctl save VMNAME** as described in the final step of the "DETAILED INSTRUCTIONS" in **ClassroomSetup.txt**.

For other locations, this section is optional.

1. Explain to your students that Red Hat Enterprise Linux officially supports a large number of languages. The book references the list at the time of writing.
2. *Official support* means that there is a certain level of support for the language in the operating system, customers can receive support for technical issues with those languages, and can ask for correction of internationalization/localization (i18n/l10n) issues if there are problems. It does not imply that communication with technical support representatives will be available in their native language.
3. It may be possible to use unsupported languages on the system, but customers might not receive support for technical issues with those languages.

## Per-user Language Selection

This subsection discusses GNOME 3 language settings for an individual user. It is divided into two chunks: how to set the language/locale correctly, and how to set the input method for the locale correctly.

If your classroom machines need to have a locale and input method other than the defaults set, use this subsection as a workshop. Have the students follow along with you in order to set up their machines. At the end of the section, they will need to save their settings to their baseline machine images so that the localization persists across server resets.

1. Mention that GNOME 3 may prompt a user for their preferred language on first login (through **/usr/libexec/gnome-initial-setup**). This may be disabled by the classroom setup process, but if it's available that's one way to set preferred locale.
2. Demonstrate **gnome-control-center region**. For System Administration I, one way would be to use the GNOME GUI, on the top bar, from the system menu in the right corner,

select the settings button (which has a crossed screwdriver and wrench for an icon) from the bottom left of the menu select **Settings** and open the Region & Language application.

3. Mention that these settings will only affect GNOME 3 sessions, not **ssh** logins or logins on a text console. Optionally mention the Note box with the shell snippet on how to tie together the GNOME locale for ssh and console logins in RHEL 8.



### Important

The kernel's physical console/virtual consoles (**\$TERM="linux"**) barely support Western European fonts, and do not support non-Latin text well at all. Locales this definitely impacts: **ja-JP**, **ko-KR**, **zh-CN**, and **zh-TW**.

For affected languages, the sample code either shouldn't be used, or should check to see if **\$TERM="linux"** and **\$LANG** is one of the affected languages, and if so sets it to **en\_US.UTF-8**. A good example of how to do this is in **/etc/profile.d/lang.sh**.

4. Explain the format of the **LANG** variable: *language\_REGION.ENCODING*.
5. Explain that single commands can be run in a different language by setting the **LANG** variable just for that command.

Demo the following command:

```
LANG=ja_JP.UTF-8 ls nosuchfile
```

If Japanese fonts are not yet installed a dialog will pop-up to ask if you want to install them. Click on **Install** and enter the **root** password when prompted.

1. Demonstrate how to add additional input methods to the system with the Region & Language application. Either add the input method appropriate to your locale, or add the **English (international AltGr dead keys)** method as an example.
2. Explain to your students that to switch between input methods they can use **Super+Space** (also known as **Windows+Space**).
3. If your chosen input method has special features (such as the **Japanese (Kana Kanji)** method), demonstrate them here.
1. Discuss (and optionally demonstrate) how **localectl set -locale** can be used to set the system-wide default language. Mention how settings are saved in **/etc/locale.conf**.
2. We do not discuss setting console keyboard settings here in depth, but if you need to do so, cover **localectl** and **/etc/vconsole.conf**.
3. Optionally, you may mention that **/etc/profile.d/lang.sh** will cause the system console to use **en\_US.UTF-8** instead of the configured system-wide locale for languages which are not well supported by the locale. (If the student sets **\$LANG** manually after that startup script runs, that will not happen.)
1. Explain that language packs install helper packages that include translated man pages, help pages, spelling dictionaries, and other supporting files.
2. **yum list langpacks\*** shows available packs
3. **yum list installed langpacks\*** shows installed packs
4. **yum install langpacks-locale\_code** installs a language pack

## Chapter 1

# Getting Started with Red Hat Enterprise Linux

### Overview

Describe and define open source, Linux, Linux distributions, and Red Hat Enterprise Linux.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	What is Linux?	P: Lecture	15
	Lab	Quiz	5
Conclusion			2

Total Time: 25 minutes

## Objectives

---

- Define and explain the purpose of Linux, open source, Linux distributions, and Red Hat Enterprise Linux.

## Key Takeaways

---

- Open source software is software with source code that anyone can freely use, study, modify, and share.
- A Linux distribution is an installable operating system constructed from a Linux kernel and supporting user programs and libraries.
- Red Hat participates in supporting and contributing code to open source projects, sponsors and integrates project software into community-driven distributions, and stabilizes the software to offer it as supported enterprise-ready products.
- Red Hat Enterprise Linux is Red Hat's open source, enterprise-ready, commercially-supported Linux distribution.

## Instructor Tips and Suggestions

---

### What is Linux?

- Do not divulge information on various open source licenses.
- Point students to Software Contributions [<https://community.redhat.com/software/>] which lists open source software where Red Hat contributes.
- Point students to Red Hat on Github [<https://redhatofficial.github.io/>] which lists the projects on Github and project websites which developers can use to contribute.



## Chapter 2

# Accessing the Command Line

### Overview

Log in to a Linux system and run simple commands using the shell.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Accessing the Command Line	P: Lecture	15
		A: Quiz	5
2	Accessing the Command Line Using the Desktop	P: Lecture	10
		A: Guided Exercise	10
3	Executing Commands Using the Bash Shell	P: Lecture	10
		A: Guided Exercise	5
	Lab	Review Lab	15
Conclusion			2

Total Time: 75 minutes

## Objectives

---

- Log in to a Linux system on a local text console and run simple commands using the shell.
- Log in to a Linux system using the GNOME 3 desktop environment and run commands from a shell prompt in a terminal program.
- Save time by using tab completion, command history, and command editing shortcuts to run commands in the Bash shell.

## Key Takeaways

---

- The Bash shell is a command interpreter that prompts interactive users to specify Linux commands.
- Many commands have a **--help** option that displays a usage message or screen.
- Using workspaces makes it easier to organize multiple application windows.
- The **Activities** button located at the upper-left corner of the top bar provides an overview mode that helps a user organize windows and start applications.
- The **file** command scans the beginning of a file's contents and displays what type it is.
- The **head** and **tail** commands display the beginning and end of a file, respectively.
- You can use **Tab** completion to complete file names when typing them as arguments to commands.

## Instructor Tips and Suggestions

---

### Accessing the Command Line

- Point out to students that in Red Hat Enterprise Linux 8, if the graphical environment is available, the login screen will run on the first virtual console, called **tty1**. Normally, the graphical session will replace the login prompt on the second virtual console (**tty2**).
- The *Logging in over the Network* topic in this section is to provide students an introduction on how to remotely log in to a server.



#### Warning

**DO NOT** discuss the OpenSSH key-based authentication as this covered in a later chapter.

## Accessing the Command Line Using the Desktop

- Highlight that the Gnome Display Manager (GDM) uses Wayland as the default display server in Red Hat Enterprise Linux 8, replacing the X.org server that was used with Red Hat Enterprise Linux 7.
- RHEL 8 uses GNOME Standard session as default as opposed to the GNOME Classic session which is the default for RHEL 7. Also RHEL 8 do not include KDE desktop.
- Point students at Useful keyboard shortcuts [<https://help.gnome.org/users/gnome-help/stable/shell-keyboard-shortcuts.html.en>] which provides list of keyboard shortcuts.



### Warning

In the online learning environment, the keys needs to be pressed twice when using visual keyboard.

## Executing Commands Using the Bash Shell

- Demonstrate `;` use to run multiple commands in order on the same command line.



### Warning

*DO NOT* introduce `||` or `&&` command chaining.

- Demonstrate the use of the **file**, **head**, **tail**, and **wc** commands. Use a text file such as **/etc/passwd**.



### Warning

*DO NOT* get into path names. They are explained in another chapter.

Demonstrate on how to use shell history:

### History Shortcuts/Commands

Shortcuts/Commands	Description
<b>history</b>	lists the shell history
<b>!</b>	history expansion
<b>UpArrow, DownArrow</b>	traverse through the commands earlier executed using the shell.
<b>Esc+.</b>	copy the last word of the previous command to the current command (where the cursor is).
<b>Alt+.</b>	iterate through the arguments earlier executed with commands using the shell.



## Chapter 3

# Managing Files From the Command Line

### Overview

Copy, move, create, delete, and organize files while working from the Bash shell.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Describing Linux File System Hierarchy Concepts	P: Lecture	10
		A: Quiz	5
2	Specifying Files by Name	P: Lecture	20
		A: Quiz	5
3	Managing Files Using Command-line Tools	P: Lecture	15
		A: Guided Exercise	15
4	Making Links Between Files	P: Lecture	15
		A: Guided Exercise	10
5	Matching File Names with Shell Expansions	P: Lecture	20
		A: Quiz	5
	Lab	Review Lab	20
Conclusion			2

Total Time: 145 minutes



## Objectives

---

- Describe how Linux organizes files, and the purposes of various directories in the file-system hierarchy.
- Specify the location of files relative to the current working directory and by absolute location, determine and change your working directory, and list the contents of directories.
- Create, copy, move, and remove files and directories.
- Make multiple file names reference the same file using hard links and symbolic (or "soft") links.
- Efficiently run commands affecting many files by using pattern matching features of the Bash shell.

## Key Takeaways

---

- Files on a Linux system are organized into a single inverted tree of directories, known as a file-system hierarchy.
- Absolute paths start with a / and specify the location of a file in the file-system hierarchy.
- Relative paths do not start with a / and specify the location of a file relative to the current working directory.
- Five key commands are used to manage files: **mkdir**, **rmdir**, **cp**, **mv**, and **rm**.
- Hard links and soft links are different ways to have multiple file names point to the same data.
- The Bash shell provides pattern matching, expansion, and substitution features to help you efficiently run commands.

## Instructor Tips and Suggestions

---

### Describing Linux File System Hierarchy Concepts

- The information presented in the "Important Red Hat Enterprise Linux Directories" table covers the bulk of what should be taught. However, the information in the Note prior to the table is equally important because learning the terms will help to build an understanding of data types and the relationship to their location in the filesystem hierarchy.
- For in-depth explanations, refer to the Filesystem Hierarchy Standard (FHS), where the issues and resolutions are discussed.

## Specifying Files by Name

- Files have fixed locations in file systems and are reached by either absolute or relative path name syntax. While both work, the use of one or the other in specific situations may be much less typing, especially with repetitive tasks. When working primarily from a home directory, shortcuts like the tilde can shorten typing even more.

## Managing Files Using Command-line Tools

- All of the commands in this section are common and basic, but can easily overwhelm novice students. Try to limit discussions to the minimum necessary command options. However, demonstrate at least one single-file task and one multiple-file task for each of the commands; create, copy, move, and remove. Then, do the same for directories.
- Point out that the **force** and **recursive** options have no safety net, so be sure to pay attention while using them.

## Making Links Between Files

- Linking concepts may be confusing to novice Linux users. Novices commonly reverse the arguments for the link command.
- Be prepared to clarify the "Important" admonition that refers to hard link behavior.
- In the first paragraph following "Creating Soft Links" it is important to identify the distinction between hard and soft links, such as hard links can not span filesystems, and more. A soft link is not a regular file, but a special type of file that points to an existing file or directory. Unlike hard links, soft links can point to a directory, and the target to which a soft link points can be on a different file system.

## Matching File Names with Shell Expansions

- Pattern matching is a feature of most shells, since command-line parsing routines are commonly available library calls. Since Bash is the default and expected shell, the focus is on pattern matching as recognized by Bash. However, be prepared in case the subject of Regular Expression syntax is mentioned.
- Be prepared to demonstrate and further explain the screen examples. When covering the "Protecting arguments" subsection, be sure that students correlate each type of expansion with the different choices for protection. For example, escape character for protecting single meta-characters, double-quotes to suppress expansion, but single-quotes to suppress it all.



## Chapter 4

# Getting Help in Red Hat Enterprise Linux

### Overview

Resolve problems by using local help systems.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Reading Manual Pages	P: Lecture	10
		A: Guided Exercise	10
2	Reading Info Documentation	P: Lecture	10
		A: Guided Exercise	10
	Lab	Review	15
Conclusion			2

Total Time: 60 minutes

## Objectives

---

- Find information in local Linux system manual pages.
- Find information from local documentation in GNU Info.

## Key Takeaways

---

- Man pages are viewed with the **man** command and provide information on components of a Linux system, such as files, commands, and functions.
- By convention, when referring to a man page the name of a page is followed by its section number in parentheses.
- Info documents are viewed with the **pinfo** command and are made up of a collection of hypertext nodes, providing information about software packages as a whole.
- The navigational keystrokes used by **man** and **pinfo** are slightly different.

## Instructor Tips and Suggestions

---

### Reading Manual Pages

- Ensure students grasp what each sections of the **man** command contains. It is not expected that they have knowledge of or experience with the contents of sections other than **(1)**.
- Use the **man -a intro** command to flip through the **intro ( )** page for each section of the Linux manual. Use the **q** key to end each screen and go to the next topic when using the **-a** option.

```
[user@host ~]$ man -a intro
```

- In guided exercise, the **man -k**, and **whereis** commands to do keyword search is enabled by executing the **mandb** command in the start script. The **mandb** command creates or updates the manual page index caches.



#### Warning

Before showing an example of a keyword search using the **man -k** command. Remember to execute the **mandb** command for the first time to create the manual page index caches.



## Reading Info Documentation

- Highlight to students that for some commands or utilities, info documents are of greater use than manual pages.
- During the guided exercise, if students put real effort to read and browse info documents, it is expected that they will comprehend both the amount and the useful type of information found in info nodes.



## Chapter 5

# Creating, Viewing, and Editing Text Files

### Overview

Create, view, and edit text files from command output or in a text editor.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Redirecting Output to a File or Program	P: Lecture	20
		A: Quiz	5
2	Editing Text Files from the Shell Prompt	P: Lecture	10
		A: Guided Exercise	15
3	Changing the Shell Environment	P: Lecture	15
		A: Guided Exercise	10
	Lab	Review Lab	15
Conclusion			2

Total Time: 95 minutes

## Objectives

---

- Save command output or errors to a file with shell redirection, and process command output through multiple command-line programs with pipes.
- Create and edit text files using the **vim** editor.
- Use shell variables to help run commands, and edit Bash startup scripts to set shell and environment variables to modify the behavior of the shell and programs run from the shell.

## Key Takeaways

---

- Running programs, or processes, have three standard communication channels, standard input, standard output, and standard error.
- You can use I/O redirection to read standard input from a file or write the output or errors from a process to a file.
- Pipelines can be used to connect standard output from one process to standard input of another process, and can be used to format output or build complex commands.
- You should know how to use at least one command-line text editor, and Vim is generally installed.
- Shell variables can help you run commands and are unique to a particular shell session.
- Environment variables can help you configure the behavior of the shell or the processes it starts.

## Instructor Tips and Suggestions

---

### Redirecting Output to a File or Program

- Explain the difference between standard input, standard output and standard error. Use the following diagram given in the student guide for this section to assist you in your explanations.

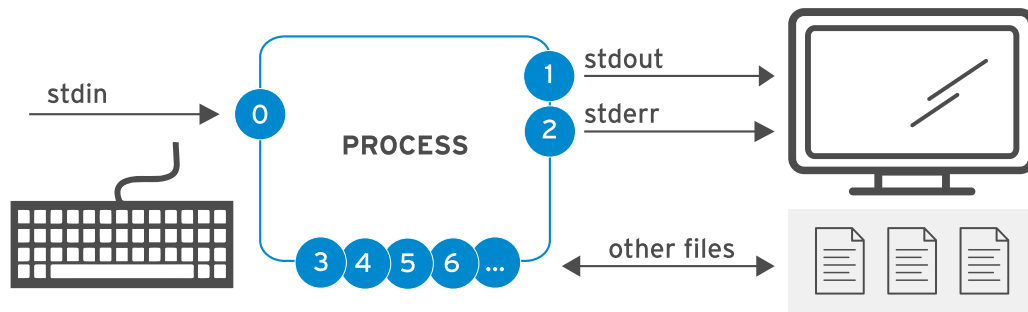


Figure 5.1: Process I/O channels (file descriptors)

- Use the following table given in the student guide for this section to mention about the file descriptors for each of standard input, output and error channels.

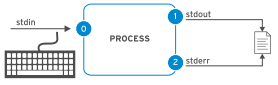
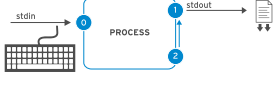
### Channels (File Descriptors)

Number	Channel name	Description	Default connection	Usage
0	<b>stdin</b>	Standard input	Keyboard	read only
1	<b>stdout</b>	Standard output	Terminal	write only
2	<b>stderr</b>	Standard error	Terminal	write only
3+	<b>filename</b>	Other files	none	read and/or write

- Use the following table given in the student guide for this section to explain the output redirection and its operators. Demonstrate the use of major operators. You do not need to demonstrate each and every operator from the table below.

### Output Redirection Operators

Usage	Explanation	Visual aid
<code>&gt; file</code>	redirect <b>stdout</b> to overwrite a file	
<code>&gt;&gt; file</code>	redirect <b>stdout</b> to append to a file	
<code>2&gt; file</code>	redirect <b>stderr</b> to overwrite a file	
<code>2&gt; /dev/null</code>	discard <b>stderr</b> error messages by redirecting to <b>/dev/null</b>	

Usage	Explanation	Visual aid
> file 2>&1	redirect <b>stdout</b> and <b>stderr</b> to overwrite the same file	
&> file		
>> file 2>&1	redirect <b>stdout</b> and <b>stderr</b> to append to the same file	
&>> file		

## Editing Text Files from the Shell Prompt

- Introduce **vim** to the students.
- Show the students how to switch between *insert mode*, *visual mode* and *extended command mode* in **vim**.
- Show the students how to edit a text file, save the changes and exit from the file using **vim**.

## Changing the Shell Environment

Refer to the same section in the student guide and present the topics as given.

## Chapter 6

# Managing Local Users and Groups

### Overview

Create, manage, and delete local users and groups and administer local password policies.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Describing User and Group Concepts	P: Lecture	10
		A: Quiz	5
2	Gaining Superuser Access	P: Lecture	15
		A: Guided Exercise	10
3	Managing Local User Accounts	P: Lecture	10
		A: Guided Exercise	10
4	Managing Local Group Accounts	P: Lecture	5
		A: Guided Exercise	10
5	Managing User Passwords	P: Lecture	15
		A: Guided Exercise	10
	Lab	Review Lab	15
Conclusion			2

Total Time: 120 minutes

## Objectives

---

- Describe the purpose of users and groups on a Linux system.
- Switch to the superuser account to manage a Linux system, and grant other users superuser access using the **sudo** command.
- Create, modify, and delete locally defined user accounts.
- Create, modify, and delete locally defined group accounts.
- Set a password management policy for users, and manually lock and unlock user accounts.

## Key Takeaways

---

- There are three main types of user account: the superuser, system users, and regular users.
- A user must have a primary group and may be a member of one or more supplementary groups.
- The three critical files containing user and group information are **/etc/passwd**, **/etc/group**, and **/etc/shadow**.
- The **su** and **sudo** commands can be used to run commands as the superuser.
- The **useradd**, **usermod**, and **userdel** commands can be used to manage users.
- The **groupadd**, **groupmod**, and **groupdel** commands can be used to manage groups.
- The **chage** command can be used to configure and view password expiration settings for users.

## Instructor Tips and Suggestions

---

### Describing User and Group Concepts

- Start the discussion with the question “What is a User?” and give the definition of the user. Mention the different types of user accounts and their use. Show them how to view the file owner and process owner. Grab a specific line from the **/etc/passwd** file and just mention which field represents what.
- Define group with the simple line “collection of users” and mention its types. Mention that the groups help in addressing multiple users while enforcing access controls. Grab a specific line from the **/etc/group** file and just mention which field represents what. Point out the differences between the primary and supplementary groups.



## Gaining Superuser Access

- Bring to students' attention the unlimited privileges that the superuser has on a system. Show them how to switch to **root** using **su**. Mention about the risks that comes with using the **su** command to switch to **root**. Introduce **sudo** and mention how it helps to overcome risks with the **su** command. Show the students how to use the **sudo** command. Show them a sample **sudo** configuration that is required for a specific user to use **sudo**.

## Managing Local User Accounts

- Show the students how to use the **useradd**, **usermod** and **userdel** commands. Also show them the use of the **passwd** command.

## Managing Local Group Accounts

- Show the students how to use the **groupadd**, **groupmod** and **groupdel** commands. Show the students how to add a group member using the **usermod** command.

## Managing User Passwords

- Mention to the students that the passwords are stored in **/etc/shadow**. Grab a specific line from the **/etc/shadow** file and just mention which field represents what. While explaining the fields of **/etc/shadow**, mention about the three components of the encrypted password. Explain the password aging attributes while explaining the relevant fields of **/etc/shadow**. Show them the use of **chage** command to adjust the password aging attributes. Bring to students' attention the file **/etc/login.defs** for setting the default maximum and minimum password ages. Show the students how to lock and unlock a user account.



## Chapter 7

# Controlling Access to Files

### Overview

Set Linux file-system permissions on files and to interpret the security effects of different permission settings.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Interpreting Linux File System Permissions	P: Lecture	20
		A: Quiz	5
2	Managing File System Permissions from the Command Line	P: Lecture	10
		A: Guided Exercise	10
3	Managing Default Permissions and File Access	P: Lecture	15
		A: Guided Exercise	10
	Lab	Review Lab	15
Conclusion			2

Total Time: 90 minutes

## Objectives

---

- List the file system permissions on files and directories, and interpret the effect of those permissions on access by users and groups.
- Change the permissions and ownership of files using command-line tools.
- Control the default permissions of new files created by users, explain the effect of special permissions, and use special permissions and default permissions to set the group owner of files created in a particular directory.

## Key Takeaways

---

- Files have three categories to which permissions apply. A file is owned by a user, a single group, and other users. The most specific permission applies. User permissions override group permissions and group permissions override other permissions.
- The **ls** command with the **-l** option expands the file listing to include both the file permissions and ownership.
- The **chmod** command changes file permissions from the command line. There are two methods to represent permissions, symbolic (letters) and numeric (digits).
- The **chown** command changes file ownership. The **-R** option recursively changes the ownership of a directory tree.
- The **umask** command without arguments displays the current umask value of the shell. Every process on the system has a umask. The default umask values for Bash are defined in the **/etc/profile** and **/etc/bashrc** files.

## Instructor Tips and Suggestions

---

### Interpreting Linux File System Permissions

- Mention to the students that Linux file permissions are used to control access. Explain the impact of read, write and execute permissions on files and directories. Mention the three categories of entities (user, group and other) that the permissions are applicable to. Consider taking an example and explain the behavioral impact of each read, write and execute permissions on files and directories for user, group and other. You can use the same example given in the student guide.

## Managing File System Permissions from the Command Line

- Show the students how to use the **chmod** to specify the permissions in numeric and symbolic modes. Show them how to change file owner and group owner using the **chown** and **chgrp** commands respectively.

## Managing Default Permissions and File Access

- Mention to the students that the special permissions provide additional access-related features over and above what the regular permissions allow. Explain the impact of SetUID, SetGID and stickybit special permissions on files and directories. Show them how to apply the special permissions using the **chmod** command.
- Take an example and explain the impact of **umask** on the default file permissions to the students. Show the students how to set the **umask** temporarily and permanently.



## Chapter 8

# Monitoring and Managing Linux Processes

### Overview

Evaluate and control processes running on a Red Hat Enterprise Linux system.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Listing Processes	P: Lecture	15
		A: Quiz	5
2	Controlling Jobs	P: Lecture	10
		A: Guided Exercise	15
3	Killing Processes	P: Lecture	15
		A: Guided Exercise	10
4	Monitoring Process Activity	P: Lecture	15
		A: Guided Exercise	15
	Lab	Review	15
Conclusion			2

Total Time: 120 minutes

## Objectives

---

- Get information about programs running on the system so that you can determine status, resource use, and ownership, so you can control them.
- Use Bash job control to manage multiple processes started from the same terminal session.
- Control and terminate processes that are not associated with your shell, and forcibly end user sessions and processes.
- Describe what load average is and determine processes responsible for high resource use on a server.

## Key Takeaways

---

- A process is a running instance of an executable program. Processes are assigned a state, which can be running, sleeping, stopped, or zombie. The **ps** command is used to list processes.
- Each terminal is its own session and can have foreground process and independent background processes. The **jobs** command displays processes within a terminal session.
- A signal is a software interrupt that reports events to an executing program. The **kill**, **killall**, and **killall** commands use signals to control processes.
- Load average is an estimate of how busy the system is. To display load average values, you can use the **top**, **uptime**, or **w** command.

## Instructor Tips and Suggestions

---

### Listing Processes

- Use the "Linux process states" diagram as the visual reference while quickly listing the process state information from the table. Answer questions about any process state, but ensure that students grasp three main concepts:

A running process is one that actively has work to do and is not waiting for resources to arrive.

Processes that have asked for resources will wait in a sleeping state. This includes waiting for disk, network and user input.

A process can be suspended and restarted; administrators can initiate this "frozen" state. Clarify that "stopped" is this suspended state, and is not the same thing as "terminated" or "killed" which are both fatal.



- For instructor knowledge only; be clear about the distinction between interruptible and uninterruptible sleeping tasks. It is common for requests to disk drives and other devices to be made uninterruptible, since allowing system administrators to abort such activity might leave a device in an unexpected or unresponsive condition. The kill-able state is an additional setting that extends the uninterruptible state. This explains why some uninterruptible processes may be killed, but will still display as the **D** state in process listings.
- When introducing the **ps** command, ensure that students comprehend the three conflicting but supported syntax forms. While all are acceptable, system administrators tend to select favorites in output choices.
- Process states are used extensively in the next sections to control processes. Return to this section and diagrams as necessary while explaining signals and controlling processes in the following sections.

## Controlling Jobs

- You can choose to demonstrate the commands listed in the student guide. Send a job to the background then bring it back to the foreground. Use the **ps** command to list in another window to show the corresponding process state.
- Explain that any command mistakenly started without an ampersand can still be sent to run in the background by first suspending it, then restarting it in the background with job control.

## Killing Processes

- While discussing signals can easily lead to other related but distracting topics (process core dumps, modem signaling, remote disconnect and reconnect or the comparison of process restart to process reload). Try to avoid these additional details; focus here on explaining signals and manipulate processes.

## Monitoring Process Activity

- You can execute a process and run **top** in a separate window. Methodically demonstrate each keystroke in the "Fundamental keystrokes in top" table. Reinforce that each duplicates commands and tasks that students have preciously accomplished, for example; **ps**, **kill**, **pgrep**.



## Chapter 9

# Controlling Services and Daemons

### Overview

Control and monitor network services and system daemons using Systemd.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Identifying Automatically Started System Processes	P: Lecture	15
		A: Guided Exercise	10
2	Controlling System Services	P: Lecture	15
		A: Guided Exercise	10
	Lab	Review	15
Conclusion			2

Total Time: 75 minutes

## Objectives

---

- List system daemons and network services started by the **systemd** service and socket units.
- Control system daemons and network services, using **systemctl**.

## Key Takeaways

---

- **systemd** provides a method for activating system resources, server daemons, and other processes, both at boot time and on a running system.
- Use the **systemctl** to start, stop, reload, enable, and disable services.
- Use the **systemctl status** command to determine the status of system daemons and network services started by **systemd**.
- The **systemctl list-dependencies** command lists all service units upon which a specific service unit depends.
- **systemd** can mask a service unit so that it does not run even to satisfy dependencies.

## Instructor Tips and Suggestions

---

### Identifying Automatically Started System Processes

- Point out to students that **service** is assumed when the unit type is not specified. The **systemctl start httpd** command is same as **systemctl start httpd.service**.
- Use **-l** option to view the full log when using the **systemctl status** command.
- Use the **systemctl -t service** command to list loaded services. The **systemctl --failed** command list the services in failed service.

### Controlling System Services

- Introduce to the students that they can use the **systemctl enable --now UNITNAME** command to start and enable services in one command.
- Even though not covered in student guide, you can introduce the **systemctl -H [hostname] restart httpd** command to control services on remote hosts.

**Note**

In addition to the references in the student guide. The following links may be useful for understanding **systemd**. Much of this information is beyond the scope of this course and are listed here for instructors *only*.

- The upstream site, systemd System and Service Manager [<https://www.freedesktop.org/wiki/Software/systemd/>], has the usual documentation, mailing lists, but also videos, external links, and information on the correct spelling of systemd.
- The Red Hat Customer Portal has some introductory videos and knowledge base articles. Additional videos may be available from the Red Hat Summit archives. View one of the latest video from Red Hat Summit 2018, Demystifying systemd [<https://www.redhat.com/en/about/videos/summit-2018-demystifying-systemd>].



## Chapter 10

# Configuring and Securing SSH

### Overview

Configure secure command-line service on remote systems, using OpenSSH.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Accessing the Remote Command Line with SSH	P: Lecture	15
		A: Guided Exercise	15
2	Configuring SSH Key-based Authentication	P: Lecture	15
		A: Guided Exercise	15
3	Customizing OpenSSH Service Configuration	P: Lecture	15
		A: Guided Exercise	15
	Lab	Review Lab	15
Conclusion			2

Total Time: 110 minutes

## Objectives

---

- Log in to a remote system and run commands using **ssh**.
- Configure key-based authentication for a user account to log in to remote systems securely without a password.
- Restrict direct logins as root and disable password-based authentication for the OpenSSH service.

## Key Takeaways

---

- The **ssh** command allows users to access remote systems securely using the SSH protocol.
- A client system stores remote servers' identities in `~/.ssh/known_hosts` and `/etc/ssh/ssh_known_hosts`.
- SSH supports both password-based and key-based authentication.
- The **ssh-keygen** command generates an SSH key pair for authentication. The **ssh-copy-id** command exports the public key to remote systems.
- The **sshd** service implements the SSH protocol on Red Hat Enterprise Linux systems.
- It is a recommended practice to configure **sshd** to disable remote logins as **root** and to require public key authentication rather than password-based authentication.

## Instructor Tips and Suggestions

---

### Accessing the Remote Command Line with SSH

- Tell the students that the SSH protocol allows connecting to remote systems. Mention to the students that OpenSSH is the implementation of the SSH protocol in Red Hat Enterprise Linux systems.
- Show the students how to use the **ssh** command to connect to remote systems. Show them how to connect to the remote system as a specific user different than the currently logged-in one. Show the students how to run a command on the remote system using the **ssh** command.
- Explain the use of SSH host keys to the students. Show them the location of the SSH host keys in the SSH server and the SSH client.



## Configuring SSH Key-based Authentication

- Explain to the students that the SSH key-based authentication enables users to log in to the remote systems without entering passwords.
- Show the students how to use the **ssh-keygen** command to generate the SSH keypair.
- Show the students how to use the **ssh-copy-id** command to share the public key of SSH keypair with the remote system. Also, show them how to use the public key of SSH keypair to log into the remote system using **ssh**.
- Show the students how to use **ssh-agent** for non-interactive authentication using passphrase-protected SSH keys.

## Customizing OpenSSH Service Configuration

- Explain the use of **PermitRootLogin** and **PasswordAuthentication** parameters of **/etc/ssh/sshd\_config** to the students. Mention about the **sshd\_config(5)** man page that can help learn other parameters of **/etc/ssh/sshd\_config**.
- Show the students how to use the **systemctl** command to bring the changes in **/etc/ssh/sshd\_config** to effect.



### Note

There used to be a note here telling folks why to use **systemctl reload sshd** rather than **systemctl restart sshd** which made incorrect assumptions about how **sshd** operates.

It is easy to make the assumption that a **systemctl restart sshd** actually kills and restarts all the **sshd** processes. It turns out that this is not quite what happens.

When someone connects to the main **sshd** service, the master process forks child processes to handle the connection. It creates a privilege separated **root** process for the connection, which then forks a child process of its own running as the user. When **sshd** is restarted, then it kills the master process but does not disrupt the child processes (orphaning the child processes for the connection, so the parent of the privsep **root** process becomes PID 1).

The original reason for this design was explicitly so restarting **sshd** does not break all active SSH connections to the host. Not all network services are this smart, though, so the point stands even though the example is wrong. If we want to make this point somewhere, a new example of a less tolerant network service will be needed.

Running **systemctl sshd stop** also does not break existing connections; those **sshd** child processes keep running even though new SSH connections cannot be established because the master **sshd** process has been stopped! This might be very surprising to students.



## Chapter 11

# Analyzing and Storing Logs

### Overview

Locate and accurately interpret logs of system events for troubleshooting purposes.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Describing System Log Architecture	P: Lecture	10
		A: Quiz	5
2	Reviewing Syslog Files	P: Lecture	20
		A: Guided Exercise	10
3	Reviewing System Journal Entries	P: Lecture	20
		A: Guided Exercise	10
4	Preserving the System Journal	P: Lecture	10
		A: Guided Exercise	10
5	Maintaining Accurate Time	P: Lecture	25
		A: Guided Exercise	5
	Lab	Review Lab	20
Conclusion			2

Total Time: 150 minutes

## Objectives

- Describe the basic logging architecture used by Red Hat Enterprise Linux to record events.
- Interpret events in relevant syslog files to troubleshoot problems or review system status.
- Find and interpret entries in the system journal to troubleshoot problems or review system status.
- Configure the system journal to preserve the record of events when a server is rebooted.
- Maintain accurate time synchronization using NTP and configure the time zone to ensure correct time stamps for events recorded by the system journal and logs.

## Key Takeaways

- The **systemd-journald** and **rsyslog** services capture and write log messages to the appropriate files.
- The **/var/log** directory contains log files.
- Periodic rotation of log files prevent them from filling up the file system space.
- The **systemd** journals are temporary and do not persist across reboot.
- The **chronyd** service helps to synchronize time settings with a time source.
- The time zone of the server can be updated based on its location.

## Instructor Tips and Suggestions

### Describing System Log Architecture

- Start the discussion by mentioning that the log messages makes the life of an administrator easy while troubleshooting an issue. Mention that **rsyslog** and **systemd-journald** are two distinct services that handle logging in Red Hat Enterprise Linux 8 systems. Mention the sources that the **systemd-journald** service collects the log messages from. Tell the students that the **systemd-journald** service stores the log messages in a structured and binary format.



#### Important

Do not yet get into showing the **journalctl** command and its options. If the students ask, just mention that we are covering this in a later section of the chapter.

- Explain how the **rsyslog** service uses the facility and priority to store the log messages in the files under the **/var/log** directory.
- Use the given table in the student guide to introduce the system-specific log files. Additionally, mention that there are service-specific log files such as **/var/log/httpd/access\_log** and **/var/log/httpd/error\_log**.

## Reviewing Syslog Files

- Use the existing SELECTOR-based rules in the **/etc/rsyslog.conf** file to explain how the **rsyslog** service determines the destination file of a log message.
- Explain the necessity of log rotation and introduce the **logrotate** program that helps in log rotation.
- Grab a sample line from **/var/log/secure** and explain the different fields of the syslog entry.
- Show them the use of **tail -f** to monitor logs in real time.
- Show them the use of **logger** to manually generate log messages for testing the **rsyslog** service configuration.

## Reviewing System Journal Entries

- Introduce **journalctl** and mention about its user-friendliness with its wide scope of options to construct granular filter for filtering Systemd events of specific interest.
- Show the students few examples of **journalctl** command.
- Explain major fields from the verbose output of **journalctl**.

## Preserving the System Journal

- Show the students how to configure persistent system journals.
- Explain the use of **-b** option with the **journalctl** command.

## Maintaining Accurate Time

- Show the students how to use **timedatectl** to view and set the system time. Show them how to use **timedatectl** to view and set the time zone. Show the students how to use **timedatectl** to activate and deactivate time synchronization.
- Show the students how to use **tzselect**.
- Introduce the **chronyd** service to the students. Discuss the parameters **stratum**, **server** and **peer** from **/etc/chrony.conf**.
- Show the students how to configure **chronyd** to synchronize system time with an NTP time source.
- Show the students the use of the **chronyc** command. Mention about the wildcard characters such as the asterisk (\*) and the question mark (?) that represents the state of synchronization with the time source.



## Chapter 12

# Managing Networking

### Overview

Configure network interfaces and settings on Red Hat Enterprise Linux servers.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Describing Networking Concepts	P: Lecture	35
		A: Quiz	5
2	Validating Network Configuration	P: Lecture	10
		A: Guided Exercise	10
3	Configuring Networking from the Command Line	P: Lecture	15
		A: Guided Exercise	10
4	Editing Network Configuration Files	P: Lecture	10
		A: Guided Exercise	10
5	Configuring Host Names and Name Resolution	P: Lecture	10
		A: Guided Exercise	10
	Lab	Review Lab	10
Conclusion			2

Total Time: 140 minutes

## Objectives

---

- Describe fundamental concepts of network addressing and routing for a server.
- Test and inspect current network configuration with command-line utilities.
- Manage network settings and devices using **nmc li**.
- Modify network settings by editing configuration files.
- Configure a server's static host name and its name resolution, and test the results.

## Key Takeaways

---

- The TCP/IP network model is a simplified, four-layered set of abstractions that describes how different protocols interoperate in order for computers to send traffic from one machine to another over the Internet.
- IPv4 is the primary network protocol used on the Internet today. IPv6 is intended as an eventual replacement for the IPv4 network protocol. By default, Red Hat Enterprise Linux operates in dual-stack mode, using both protocols in parallel.
- NetworkManager is a daemon that monitors and manages network configuration.
- The **nmc li** command is a command-line tool for configuring network settings with NetworkManager.
- The system's static host name is stored in the **/etc/hostname** file. The **hostnamectl** command is used to modify or view the status of the system's host name and related settings. The **hostname** command displays or temporarily modifies the system's host name.

## Instructor Tips and Suggestions

---

### Describing Networking Concepts

- The discussion on network layers helps to provide context for later discussion and a basic understanding of network communication, it is not critical for students to understand the layers in detail. The key points are that network communication is layered, each layer depends on the layer below it, and that there are multiple implementations available at each layer. It is easy to get carried-away on this topic, try not to spend too much time on it.



## Validating Network Configuration

- Communicate to students the importance of the commands in this section. Whether users are configuring or troubleshooting networks these commands are the likely starting point.

## Configuring Networking from the Command Line

- Show students that they do not have to memorize the sequence of **nmcli** options to access a particular report or configuration option. Use the **tabtab** key strokes from beginning to end to display the available options.

```
[student@servera ~]$ nmcli tabtab
agent connection device general help monitor networking radio
[student@servera ~]$ nmcli device tabtab
connect delete disconnect help lldp modify monitor reapply set show status wifi
[student@servera ~]$ nmcli device show tabtab
ens3 help lo
[student@servera ~]$ nmcli device show ens3
...output omitted...
```

Additionally, use **nmcli help** option at any point to view details of a particular option:

```
[student@servera ~]$ nmcli help
Usage: nmcli [OPTIONS] OBJECT { COMMAND | help }
...output omitted...
[student@servera ~]$ nmcli device help
Usage: nmcli device { COMMAND | help }
...output omitted...
```

## Editing Network Configuration Files

- Make it clear to students the importance of running the **nmcli con reload** command if they choose to directly edit network configuration files.

## Configuring Host Names and Name Resolution

- For students familiar with earlier versions of Red Hat Enterprise Linux you should point out the following admonition:



### Important

In Red Hat Enterprise Linux 7 and later, the static host name is stored in **/etc/hostname**. Red Hat Enterprise Linux 6 and earlier stores the host name as a variable in the **/etc/sysconfig/network** file.



## Chapter 13

# Archiving and Transferring Files

### Overview

Archive and copy files from one system to another.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Managing Compressed tar Archives	P: Lecture	20
		A: Guided Exercise	10
2	Transferring Files Between Systems Securely	P: Lecture	20
		A: Guided Exercise	10
3	Synchronizing Files Between Systems Securely	P: Lecture	20
		A: Guided Exercise	10
	Lab	Review	20
Conclusion			2

Total Time: 120 minutes

## Objectives

---

- Archive files and directories into a compressed file using `tar`, and extract the contents of an existing tar archive.
- Transfer files to or from a remote system securely using SSH.
- Synchronize the contents of a local file or directory with a copy on a remote server.

## Key Takeaways

---

- The **tar** command creates an archive file from a set of files and directories, extracts files from the archive, and lists the contents of an archive.
- The **tar** command provides a set of different compression methods reduce archive size.
- Besides providing a secure remote shell, the **SSH** service also provides the **scp** and **sftp** commands as secure ways to transfer files from and to a remote system running the **SSH** server.
- The **rsync** command securely and efficiently synchronizes files between two directories, either one of which can be on a remote system.

## Instructor Tips and Suggestions

---

### Managing Compressed tar Archives

- Highlight to students that when archiving files by absolute path names, the leading `/` of the path is removed from the file name by default.
- By default, `tar` preserves file permissions and ownership when creating the archive. To preserve the permissions of an archived file, the **p** option when extracting an archive. Point out that to preserve the ownership of an archived file, the **--same-owner** option can be used by the superuser.

### Transferring Files Between Systems Securely

- Communicate to the students that both the **scp**, and **sftp** supports globbing. Here are some of the examples:

```
[user@host ~]$ scp remotehost:'/tmp/*.txt' .  
...output omitted...  
[user@host ~]$ sftp root@remotehost  
...output omitted...  
sftp> ls /etc/*.conf  
...output omitted...
```

## Synchronizing Files Between Systems Securely

- Ensure at least the following options of the **rsync** are discussed: **-a**, **-v**, and **-n**.
- Highlight that while executing the **rsync** command, a trailing slash at the end of the source directory allows only to synchronize the content of a directory without creating the subdirectory in the target directory.



## Chapter 14

# Installing and Updating Software Packages

### Overview

Download, install, update, and manage software packages from Red Hat and Yum package repositories.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Registering Systems for Red Hat Support	P: Lecture	10
		A: Quiz	5
2	Explaining and Investigating RPM Software Packages	P: Lecture	20
		A: Guided Exercise	10
3	Installing and Updating Software Packages with Yum	P: Lecture	15
		A: Guided Exercise	10
4	Enabling Yum Software Repositories	P: Lecture	10
		A: Guided Exercise	10
5	Managing Package Module Streams	P: Lecture	10
		A: Guided Exercise	10
	Lab	Review Lab	15
Conclusion			2

Total Time: 130 minutes

## Objectives

---

- Register a system to your Red Hat account and assign it entitlements for software updates and support services using Red Hat Subscription Management.
- Explain how software is provided as RPM packages, and investigate the packages installed on the system with Yum and RPM.
- Find, install, and update software packages using the **yum** command.
- Enable and disable use of Red Hat or third-party Yum repositories by a server.
- Explain how modules allow installation of specific versions of software, list, enable, and switch module streams, and install and update packages from a module.

## Key Takeaways

---

- Red Hat Subscription Management provides tools to entitle machines to product subscriptions, get updates to software packages, and track information about support contracts and subscriptions used by the systems.
- Software is provided as RPM packages, which make it easy to install, upgrade, and uninstall software from the system.
- The **rpm** command can be used to query a local database to provide information about the contents of installed packages and install downloaded package files.
- **yum** is a powerful command-line tool that can be used to install, update, remove, and query software packages.
- Red Hat Enterprise Linux 8 uses Application Streams to provide a single repository to host multiple versions of an application's packages and its dependencies.

## Instructor Tips and Suggestions

---

### Registering Systems for Red Hat Support

- Even with an Internet connection and a Red Hat login, it is recommended to keep the lecture short. Students should be able to figure out the GUI screens once they are working with their systems. Point out that subscription choices may differ depending on the type of subscription.



## Explaining and Investigating RPM Software Packages

- The structure of RPM file names is sometimes confusing to novice Red Hat Enterprise Linux users. Therefore, instructors may want to spend a little more time on the "Software Packages and RPM" section in anticipation of student questions.
- Keep this section as simple as possible. Consider showing examples of common **rpm** command options, such as for gathering information **-i** and listing files **-l** in a package.

## Installing and Updating Software Packages with Yum

- Although the examples in this section meet the objectives, instructors may choose to provide additional examples for searching, finding, installing and updating software using the **yum** command.

## Enabling Yum Software Repositories

- Clarify that Extra Packages for Enterprise Linux (EPEL) repositories provide software that is compatible with Red Hat Enterprise Linux but not supported by Red Hat.

## Managing Package Module Streams

- Modularity is a new technology in Red Hat Enterprise Linux 8, therefore instructors should be prepared to highlight the "Introduction to Modularity" section and subsections.



## Chapter 15

# Accessing Linux File Systems

### Overview

Access, inspect, and use existing file systems on storage attached to a Linux server.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Identifying File Systems and Devices	P: Lecture	15
		A: Quiz	5
2	Mounting and Unmounting File Systems	P: Lecture	20
		A: Guided Exercise	10
3	Locating Files on the System	P: Lecture	25
		A: Guided Exercise	10
	Lab	Review	15
Conclusion			2

Total Time: 105 minutes

## Objectives

---

- Explain what a block device is, interpret the file names of storage devices, and identify the storage device used by the file system for a particular directory or file.
- Access file systems by attaching them to a directory in the file system hierarchy.
- Search for files on mounted file systems using the **find** and **locate** commands.

## Key Takeaways

---

- Storage devices are represented by a special file type called block device.
- The **df** command reports total disk space, used disk space, and free disk space on all mounted regular file systems.
- The **mount** command allows the **root** user to manually mount a file system.
- All processes need to stop accessing the mount point in order to successfully unmount the device.
- The removable storage devices are mounted in the **/run/media** directory when using the graphical environment.
- The **find** command performs a real-time search in the local file systems to find files based on search criteria.

## Instructor Tips and Suggestions

---

### Identifying File Systems and Devices

- Ensure students understand which devices files are available in a system and how much storage space is used and available on a file system.
- Ensure students understand the difference between **-h** and **-H** options used with the **df** command.

### Mounting and Unmounting File Systems

- Highlight to the students that the **lsblk** command with the **--fs** does not show the full device path. Thus using the **-fp** in short version or **--fs --path** options in long version makes them easy to understand and recognize the device information.
- Demonstrate with your USB stick where the graphical environment mounts removable media.

## Locating Files on the System

- Highlight that **updatedb**, is used by system administrators to create databases for **locate** to use.
- Highlight that the **find** command, can take multiple directories as arguments. Also, highlight the fact that the **-path** option arguments ending in **/** will match nothing.



### Note

Additional information for instructors, when using the **find** command use the **-prune** option to ignore a whole directory tree.



## Chapter 16

# Analyzing Servers and Getting Support

### Overview

Investigate and resolve issues in the web-based management interface, getting support from Red Hat to help solve problems.

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Analyzing and Managing Remote Servers Objectives Describing the Insights Architecture Detecting Configuration Issues Using the Advisor Service	P: Lecture	10
		A: Guided Exercise	10
2	Getting Help From Red Hat Customer Portal Installing Insights Clients Analyzing Compliance Using the Compliance Service	P: Lecture	10
		A: Guided Exercise	10
3	Detecting and Resolving Issues with Red Hat Insights Comparing Systems Using the Drift Service	P: Lecture	10
		A: Quiz	5
	Lab	Review Lab	10
Conclusion			2

Total Time: 70 minutes

## Objectives

---

- Activate the Web Console management interface to remotely manage and monitor the performance of a Red Hat Enterprise Linux server.
- Describe key resources available through the Red Hat Customer Portal, and find information from Red Hat documentation and the Knowledgebase.
- Analyze servers for issues, remediate or resolve them, and confirm the solution with Red Hat Insights.

## Key Takeaways

---

- Web Console is a web-based management interface to your server based on the open source Cockpit service.
- Web Console provides graphs of system performance, graphical tools to manage system configuration and inspect logs, and an interactive terminal interfaces.
- Red Hat Customer Portal provides you with access to documentation, downloads, optimization tools, support case management, and subscription and entitlement management for your Red Hat products.
- **redhat-support-tool** is a command-line tool to query Knowledgebase and work with support cases from the server's command line.
- Red Hat Insights is a SaaS-based predictive analytics tool to help you identify and remediate threats to your systems' security, performance, availability, and stability.

## Instructor Tips and Suggestions

---

### Analyzing and Managing Remote Servers Objectives Describing the Insights Architecture Detecting Configuration Issues Using the Advisor Service

- Students may question the Web Console message displayed to the console when they log in to a remote server. Notice that the message changes depending on the state of the Cockpit service.

Prior to starting and enabling Web Console the message displays how to start and enable it.



```
[student@workstation ~]$ ssh student@servera
Activate the web console with: systemctl enable --now cockpit.socket
```

After Web Console is started the message displays how to access it.

```
[student@workstation ~]$ ssh student@servera
Web console: https://servera.lab.example.com:9090/ or https://172.25.250.10:9090/
```

## Getting Help From Red Hat Customer PortalInstalling Insights ClientsAnalyzing Compliance Using the Compliance Service

- An effective skill set for troubleshooting is the ability to gather accurate information fast. Therefore, you may want to place a slightly higher focus on the "Using Red Hat Support Tool to Search Knowledgebase" and "Accessing Knowledgebase Articles by Document ID" sections.

## Detecting and Resolving Issues with Red Hat InsightsComparing Systems Using the Drift Service

- Instructors should thoroughly review the Insights information provided in the student guide prior to delivering this section. Insights is another of the topics that can consume time, therefore, try to anticipate student questions and keep the lecture concise.



## Chapter 17

# Comprehensive Review

### Overview

Review tasks from *Red Hat System Administration I*

### Schedule

#### ILT/VT Schedule

Section	Title	Presentation & Engagement Methods	Time (minutes)
Introduction			3
1	Managing Files from the Command Line	A: Review Lab	30
2	Managing Users and Groups, Permissions and Processes	A: Review Lab	20
3	Configuring and Managing a Server	A: Review Lab	30
4	Managing Networks	A: Review Lab	20
5	Mounting Filesystems and Finding Files	A: Review Lab	15
Conclusion			2

Total Time: 120 minutes

# Objectives

---

- Review tasks from *Red Hat System Administration I*

## Instructor Tips and Suggestions

---

### Managing Files from the Command Line

Before the students start this review lab, mention that managing and editing files is among the fundamental skills that the Linux system administrator needs. Be it modifying a plain text file or adjusting the configuration settings of a daemon in the system, the administrator should be comfortable in performing those file operations from the command line. This review lab causes the students to revisit those fundamental file operations skills.

### Managing Users and Groups, Permissions and Processes

Before the students start this review lab, mention that controlling file access is an important part of Linux system administration and so is controlling process. Without an appropriate understanding of the access control mechanism that Linux enforces on users, groups and others, handling of a Linux operating system even at the user level could get frustrating. For example, if an individual does not know how to enable or disable access rights on a file that he or she owns, the individual may end up either giving more permissions on the file than actually required or vice versa. Also, if a user does not know how to control a process that he or she started, he would not know how to communicate with the process if he wants to terminate or suspend the process, for example. This review lab causes the students to revisit those user, group, permission and process management skills.

### Configuring and Managing a Server

Before the students start this review lab, mention that the Linux system administrators perform server patching, updating time synchronization settings, backing up and restoring files as their day-to-day activities. The system administrator usually connects from a remote system to the servers. So, securing the remote connectivity to those servers is also a system administration task. This review lab causes the students to revisit those system administration skills.

### Managing Networks

Before the students start this review lab, mention that learning how to configure network interfaces is not only valuable in production infrastructure but also in home infrastructure. For example, if you intend to use a certain IP address statically that your home router has emitted to your personal computer, you can configure the network interface of your personal computer with that static IP address. This review lab causes the students to revisit the administration skills required to configure network interfaces.

## Mounting Filesystems and Finding Files

This review lab causes the students to revisit the user-level skills of accessing a locally attached file system and finding files.

