CHAPTER-2 GET STARTED WITH RED HAT ENTERPRISE LINUX

Open source software

Open source software has source code that anyone can freely use, study, modify, and share.
A Linux distribution is an installable operating system that is constructed from a Linux kernel and that supports 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 the open source, enterprise-ready, commercially supported Linux distribution provided by Red Hat.

A free Red Hat Developer Subscription is a useful method for obtaining learning resources and information, including developer subscriptions to Red Hat Enterprise Linux and other Red Hat products.

Access the Command Line

Introduction to the Bash Shell

A command line is a text-based interface that is used to input instructions to a computer system. The Linux command line is provided by a program called the shell. Many shell program variants have been developed over the years. Every user can use a different shell, but the Red Hat recommends using the default shell for system administration.

The default user shell in Red Hat Enterprise Linux (RHEL) is the GNU Bourne-Again Shell (bash). The shell displays a string when it is waiting for user input, called the shell prompt. When a regular user starts a shell, the prompt includes an ending dollar ($) character:

[user@host ~]**$** (Normal user or standard user login)

A hash (#) character replaces the dollar ($) character when the shell is running as the superuser, root.
[root@host ~]**#** (root or Administrator user login)

Shell Basics

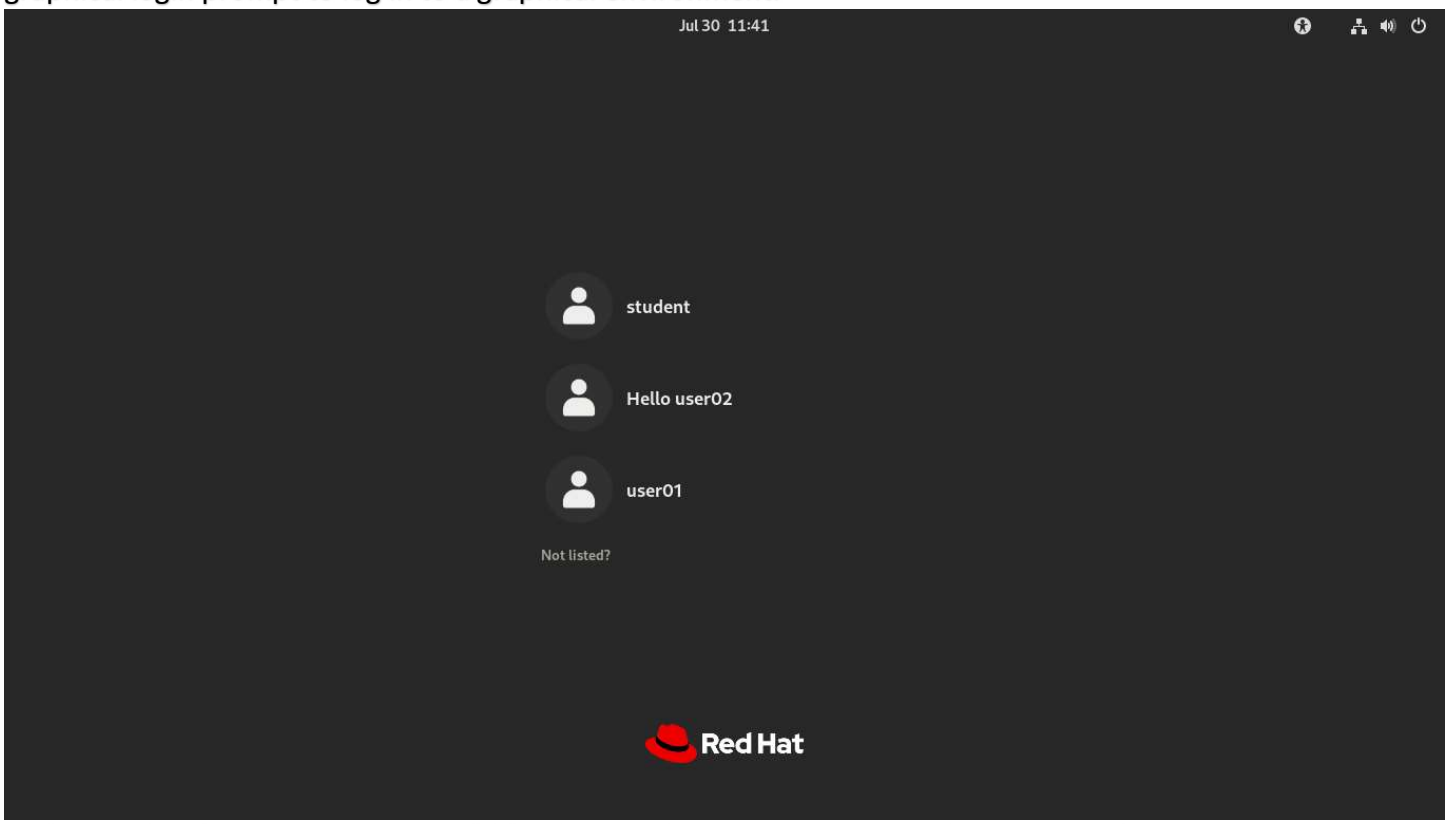
Commands that are entered at the shell prompt have three basic parts:

- Command to run.
- Options to adjust the behavior of the command.
- Arguments, which are typically targets of the command.

Log in to a Local System

A terminal is a text-based interface to enter commands into and print output from a computer system. To run the shell, you must log in to the computer on a terminal.

You can switch between the virtual consoles by pressing Ctrl+Alt and a function key (F1 through F6) at the same time. Most of these virtual consoles run a terminal that provides a text login prompt. If you enter your username and password correctly, then you log in and get a shell prompt.
The computer might provide a graphical login prompt on one of the virtual consoles. You can use the graphical login prompt to log in to a graphical environment.



If the graphical environment is available, then the login screen runs on the first virtual console, which is called tty1. Five additional text login prompts are available on virtual consoles two tty2 through six tty6.

Log in to a Remote System

Linux users and administrators often need to get shell access to a remote system by connecting to it over the network. In a modern computing environment, many headless servers are virtual machines or are running as public or private cloud instances. These systems are not physical and do not have real hardware consoles. They might not even provide access to their (simulated) physical console or serial console.

In Linux, the most common way to get a shell prompt on a remote system is to use Secure Shell (SSH).

Most Linux systems (including Red Hat Enterprise Linux) and macOS provide the OpenSSH command-line program ssh for this purpose.
Example: a user with a shell prompt on the machine host uses ssh to log in to the remote Linux system remotehost as the user remoteuser:

[user@host ~]$ **ssh remoteuser@remotehost**
remoteuser@remotehost's password: **password**
[remoteuser@remotehost ~]$ (remote user login)

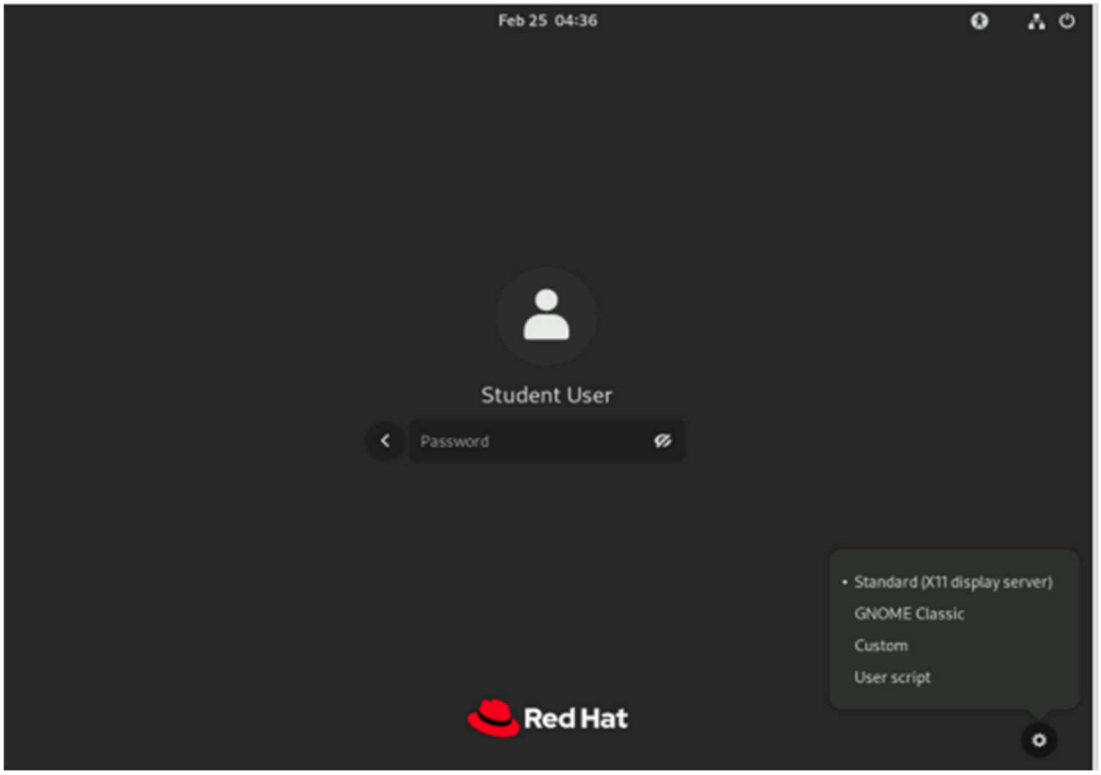
The ssh command encrypts the connection to secure the communication against eavesdropping or hijacking of the passwords and content.

When you are finished with the shell and want to quit, you can choose one of several ways to end the session. You can enter the exit command to terminate the current shell session. Alternatively, finish a session by pressing Ctrl+D.

Access the Command Line with the Desktop

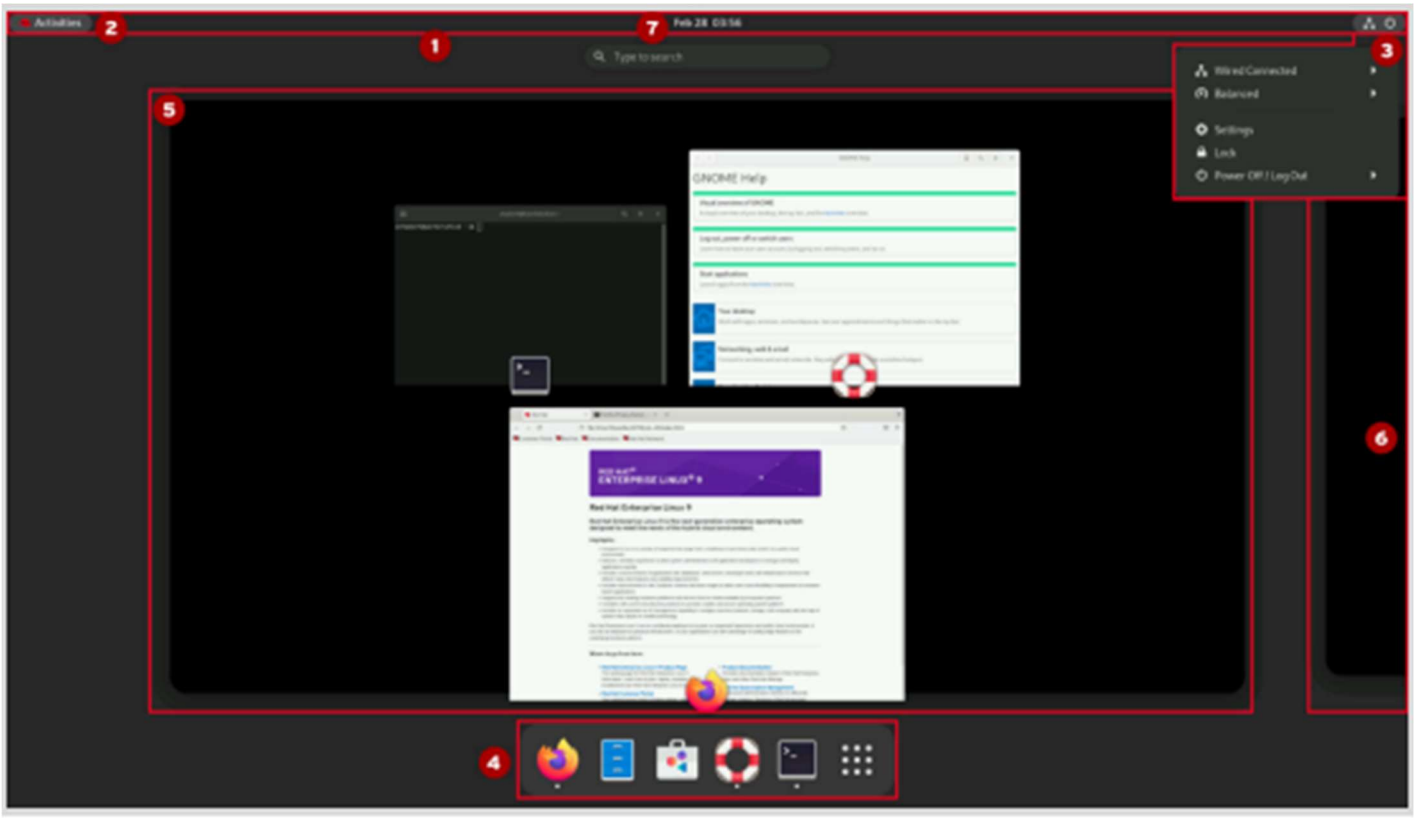
The desktop environment is the graphical user interface on a Linux system. GNOME 40 is the default desktop environment in Red Hat Enterprise Linux 9. It provides an integrated desktop for users and a unified development platform on top of a graphical framework provided by either Wayland (by default) or the legacy X Window System.

The RHEL 9 login screen



Parts of the GNOME Shell

The elements of the GNOME Shell include the following parts, as shown in this screen capture of the GNOME Shell in Activities overview mode:



1. **Top bar:** The bar that runs along the top of the screen. It is displayed in the Activities overview and in workspaces. The top bar provides the Activities button and controls for volume, networking, calendar access, and switching between keyboard input methods (if more than one method is configured).

2. **Activities overview:** This mode helps to organize windows and start applications. Enter the Activities overview by clicking the Activities button at the upper-left corner of the top bar, or by pressing the Super key. Find the Super key (sometimes called the Windows key or Command key) near the lower-left corner of most common keyboards. The three main areas are the dash at the bottom of the screen, the windows overview in the center, and the workspace selector on the right side.

3. **System menu:** The menu in the upper-right corner on the top bar provides control to adjust the brightness of the screen, and to switch on or off the network connections. Under the submenu for the user's name are options to adjust account settings and to log out of the system. The system menu also offers buttons to open the Settings window, lock the screen, or shut down the system.

4. **Dash:** This configurable list of icons shows your favorite applications, running applications, and a Show Applications button to select arbitrary applications. Start applications by clicking an icon or

by using the Show Applications button to find less commonly used applications. The dash is also called the dock.

5. **Windows overview:** The area in the center of the Activities overview that displays thumbnails of active windows in the current workspace, for bringing windows to the foreground on a cluttered workspace, or moving them to another workspace.

6. **Workspace selector:** An area to the right which displays thumbnails of active workspaces and allows workspaces to be selected and windows to be moved from one workspace to another.

7. **Message tray:** With the message tray, you can review notifications from applications or system components. If a notification occurs, the notification typically first appears briefly as a single line at the top of the screen, and a persistent indicator appears in the top bar next to the clock to inform you of recently received notifications. Open the message tray to review these notifications by clicking the clock on the top bar or by pressing Super+M. Close the message tray by clicking the clock on the top bar, or by pressing Esc or Super+M again. The message tray also shows the calendar and information about the events in the calendar.

Execute Commands with the Bash Shell

The GNU Bourne-Again Shell (bash) is a program that interprets commands that the usertypes. Each string that is typed into the shell can have up to three parts: the command, options (which usually begin with a hyphen - or double hyphen -- characters), and arguments. Each word that is typed into the shell is separated from other words with spaces. Commands are the names of programs that are installed on the system. Each command has its options and arguments. When you are ready to execute a command, press the Enter key. Type each command on a separate line.

To type more than one command on a single line, use the semicolon (;) as a command separator.
A semicolon is a member of a class of characters called metacharacters that have a special interpretation for bash. In this case, the output of both commands is displayed before the following shell prompt appears.

```
[user@host ~]$ command1 ; command2
command1 output
command2 output
[user@host ~]$
```

DESCRIPTION	COMMANDS / OPTIONS
To displays the current date and time	**Syntax:** date [OPTION]... [+FORMAT] date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]] **Example:** [user@host ~]$date ↲ **-u** To display the current time in GMT (GMT) or UTC. **-d** To display the date of your choice **Example:** To display the date of your choice [user@host ~]$ date –d tomorrow ↲ **-f** To display dates given in a file **-R** To display date with reference to GMT **-s** To set date and time (Date format :**YYYYMMDD hhmm**) **-r** To display modification time of a file **Example:** To display date by full date, month and Year [user@host ~]$ date +"%A%B%Y" ↲ **Example:** To display Full month name last two digits of the year [user@host ~]$ date +"%B%Y" ↲
To view type of files	**Syntax:** File [options][filename] **-b** brief ***** displays the all files's file type **-c** Cause a checking printout of the parsed form of the magic file **-f** Read the names of the files to be examined from namefile (one per line) before the argument list
To view the Contents of Files	**Syntax:** cat [OPTION] [FILE] To View the Content of a Single File in Linux **Example:** [user@host ~]$ cat fileName ↲ To View the Content of Multiple Files in Linux **Example:** [user@host ~]$ cat file1 file2 ↲ **-n** To cat introduces line numbers, making it convenient to identify and reference specific lines within the file **Example:** [user@host ~]$ cat –n file1 ↲ **>** To create a new file or overwrite an existing file with new content **Example:** [user@host ~]$ cat > newfile_name ↲

	To Copy the Contents of One File to Another File in Linux **Example:** [user@host ~]$ cat file1.txt file2.txt > merged_file.txt ↲ To Append the Contents of One File to the End of Another File **Example:** [user@host ~]$ cat file_name1 >> file_name2↲
To display N line of end of File (By default, these commands display 10 lines of the file)	**Syntax:** tail [OPTION]... [FILE]... **Example:** [user@host ~]$ tail file1↲ **-n** option to specify a different number of lines. **Example:** [user@host ~]$ tail –n 5 file1↲
To counts lines, words, and characters in a file	**Syntax:** wc [OPTION]... [FILE]... **Example:** [user@host ~]$ wc file1↲ **-l** number of lines **-w** number of words **-c** number of characters
To write one command in more than one line, use a backslash character (\), which is referred to as the escape character. The backslash character ignores the meaning of the following character.	**Example:** [user@host ~]$ head -n 3 \ /usr/share/dict/words \ /usr/share/dict/linux.words
To display previous entered command history	**Syntax:** history **-c** clear history

Tab Completion

With tab completion, users can quickly complete commands or file names after typing enough at the prompt to make it unique. If the typed characters are not unique, then pressing the Tab key twice displays all commands that begin with the typed characters.

```
[user@host ~]$ pasTab+Tab❶
passwd       paste        pasuspender
[user@host ~]$ passTab ❷
[user@host ~]$ passwd
Changing password for user user.
Current password:
```

❶ Press Tab twice.

❷ Press Tab once.

Useful Command-line Editing Shortcuts

Short-Cut	Used for
Ctrl + A	Jump to the beginning of the command line
Ctrl + E	Jump to the end of the command line
Ctrl + U	Clear from the cursor to the beginning of the command line
Ctrl + K	Clear from the cursor to the end of the command line
Ctrl + LeftArrow	Jump to the beginning of the previous word on the command line
Ctrl + RightArrow	Jump to the end of the next word on the command line
Ctrl + R	Search the history list of command for a pattern