DUMPSBOSS.

Red Hat Linux Essentials

RedHat RH033

Version Demo

Total Demo Questions: 21

Total Premium Questions: 218

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

Which of the following commands can be used to display the contents of a text file one screen at a time?

Each correct answer represents a complete solution. Choose all that apply.

A. less filename

B. more filename

C. rm filename

D. mkdir filename

ANSWER: A B

Explanation:

The `more` and `less` commands are pagers that display text a screen at a time. Both can be used with a file name, such as `more messages.log` or `less messages.log`, to read a long file without printing all of its contents to the terminal at once.

`less` provides additional navigation features, including backward movement through the file, but both commands meet the requirement. See the less(1) manual page and the more(1) manual page.

QUESTION NO: 2

Which of the following protocols are used for mail delivery?

Each correct answer represents a complete solution. Choose two.

A. IMAPS

B. SMTP

C. ESMTP

D. POP3S

ANSWER: B C

Explanation:

SMTP and ESMTP are used to submit and relay email messages. SMTP, the Simple Mail Transfer Protocol, is the core Internet standard for transferring mail from a mail client to a sending mail server and between mail servers as a message is routed to its destination. ESMTP, Extended SMTP, is the widely deployed extension mechanism for SMTP. An SMTP client identifies ESMTP support with the `EHLO` command, after which the server can advertise capabilities such as message-size limits, authentication support, and delivery-status notifications. In practical Linux mail-server configurations, modern SMTP software normally uses ESMTP extensions while retaining SMTP's basic message-transfer role.

Therefore, both SMTP and ESMTP correctly describe protocols used for mail delivery or transfer. This distinction is important: "delivery" in this context means transporting a message to a mail server or relaying it between servers, rather than allowing an end-user mail client to retrieve messages already stored in a mailbox. The SMTP protocol specification is maintained by the IETF in RFC 5321, which defines SMTP and its extension negotiation framework. Red Hat's mail-server documentation also describes SMTP as the protocol used to transfer email messages: Red Hat Enterprise Linux Deployment Guide: Email Applications.

QUESTION NO: 3

Which of the following key-combinations are used to view a file in two windows in a single vim screen?

Each correct answer represents a complete solution. Choose two.

A. Ctrl-w, v

B. Ctrl-w, s

C. :wq

D. :q!

ANSWER: A B

Explanation:

`Ctrl-w, v` and `Ctrl-w, s` are Vim window commands that divide the current Vim editing area so content can be viewed in two windows within one Vim screen. In normal mode, `Ctrl-w` is the prefix for window-management commands. Following it with `v` creates a vertical split, placing the windows side by side. This is useful when comparing separate portions of a file or working with two buffers next to each other. Following the window prefix with `s` creates a horizontal split, placing one window above the other. Both commands leave the original buffer available and create an additional window; the user can then move between windows with other `Ctrl-w` commands and can load a different file into either window if needed. Vim's built-in help documents these commands as `:vsplit` for a vertical split and `:split` for a horizontal split, with the corresponding normal-mode mappings `CTRL-W v` and `CTRL-W s`. See the official Vim documentation for window commands and the Red Hat System Administration I curriculum.

QUESTION NO: 4

Which of the following statements are true about file permissions?

Each correct answer represents a complete solution. Choose three.

A. The execute permission means that the file can be run if it is a program or script.

B. The write permission means that the file can be edited and saved.

C. The read permission means that only the file name can be read not the contents of the file.

D. The read permission means that the contents of the file can be seen with a command such as cat or less.

ANSWER: A B D

Explanation:

On Linux systems, permissions are evaluated separately for reading, writing, and executing a regular file. "The execute permission means that the file can be run if it is a program or script." is correct because the execute bit authorizes execution of a program file. This is the permission normally required when launching an executable binary or invoking a script as a command. "The write permission means that the file can be edited and saved." is correct because write permission on a regular file permits changing its contents; in practical editing workflows, the ability to save changes can additionally depend on permissions for the containing directory when an editor replaces the file. "The read permission means that the contents of the file can be seen with a command such as cat or less." is correct because read permission authorizes a process to open the file and retrieve its data, including through common viewing commands such as `cat` and `less`. These permissions apply independently and are assigned for the file owner, group, and other users. Red Hat's documentation describes the use of `r`, `w`, and `x` permission bits and their role in controlling file access. See Red Hat Enterprise Linux System Administrator's Guide: Managing File Permissions and Red Hat documentation: Managing file permissions.

QUESTION NO: 5

John is a newly appointed Network technician at McNeil Inc. The company has a Linux-based

Network. John is currently working on the /tmp directory. The /tmp directory has a file named file3.txt. John wants to copy this file in a subdirectory named backup of his home directory /home/John. Which of the following commands should John use to accomplish his task?

Each correct answer represents a complete solution. Choose three.

A. cp file3.txt /home/John/backup

B. cp file3.txt /home/John/backup/

C. cp file3.txt backup

D. cp file3.txt ~/backup

ANSWER: A B D

Explanation:

The commands `cp file3.txt /home/John/backup` and `cp file3.txt /home/John/backup/` are complete solutions because the current working directory is `/tmp`, so the relative source name `file3.txt` identifies `/tmp/file3.txt`. Both destination paths explicitly identify the intended `backup` directory beneath John's home directory. A trailing slash on the destination is valid and makes the directory intent explicit; in either form, `cp` places the source file into that existing directory while retaining its name, resulting in `/home/John/backup/file3.txt`.

The command `cp file3.txt ~/backup` is also complete when John runs it. The shell expands the tilde character at the beginning of a pathname to the current user's home directory. For John, `~` therefore expands to `/home/John`, making `~/backup` equivalent to `/home/John/backup`. This command consequently copies the file from the current directory into the requested destination. The destination directory must already exist for these commands to copy the file into it. The `cp` utility's source-and-destination behavior is documented in the GNU Coreutils cp documentation; shell tilde expansion is described in the Bash Reference Manual.

QUESTION NO: 6

You work as a Network Administrator for Rick International. The company has a Linuxbased network. A process is running on a server. It is using a lot of processing power. You want the process to use less resources. Which of the following actions will you perform to accomplish the task?

A. Kill the process and rerun it by using the nice command with the -n 1 switch.

B. Kill the process and rerun it by using the nice command with the -n 19 switch.

C. Kill the process and rerun it by using the fg command.

D. Kill the process and rerun it by using the bg command.

ANSWER: B

Explanation:

Kill the process and rerun it by using the nice command with the -n 19 switch. is correct because the `nice` value influences the CPU scheduling priority assigned to a process. A higher nice value means that the process is more "nice" to other processes: it receives a lower scheduling priority and is therefore less favored for CPU time when the system is busy. The normal Linux nice range is from `-20`, which represents the highest priority, to `19`, which represents the lowest priority. Starting the workload with `nice -n 19 command` gives it the lowest standard CPU scheduling priority, reducing its competition for processor time relative to normally prioritized processes. For a process that is already running, an administrator can alternatively adjust its priority with `renice`; however, among the listed actions, stopping it and restarting it with `nice -n 19` accomplishes the requested outcome. Nice levels affect CPU scheduler priority rather than imposing a fixed CPU percentage limit, so the effect is most relevant when CPU contention exists. Red Hat documents process-priority adjustment through `nice` and `renice` in its Managing processes guidance. The command's exact behavior and syntax are also described in the nice manual page.

QUESTION NO: 7

You want to change the group ownership of the file `project.txt` to the group named `developers`. Which of the following commands should you use?

A. chmod developers project.txt

B. chown developers project.txt

C. chgrp developers project.txt

D. groupmod developers project.txt

ANSWER: C

Explanation:

The `chgrp developers project.txt` command changes the group ownership of `project.txt` to `developers`. The user running the command must normally own the file and be a member of the target group, or must have administrative privileges. Changing a file group can be used to make the file accessible to the members of a project group when the appropriate group permissions are also present. See the chgrp(1) manual page.

QUESTION NO: 8

Which of the following commands can be used by a user to display the list of contents existing in his current user account?

Each correct answer represents a complete solution. Choose all that apply.

A. echo *

B. ls ~

C. ls *

D. pwd

ANSWER: A B

Explanation:

The commands `echo *` and `ls ~` can display the contents associated with the user's account when the intended location is the user's home directory. In a shell, the tilde character in `ls ~` is expanded to the current user's home-directory path before `ls` runs. The command therefore lists the entries stored directly in that home directory, regardless of the shell's current working directory. This is a common and explicit way to inspect a user's account files.

The command `echo *` relies on shell pathname expansion. When it is run while the current directory is the user's home directory, the shell expands the asterisk into the names of the non-hidden entries in that directory, and `echo` prints those names. Thus, it provides a simple listing of the visible home-directory contents in the usual login-shell context. Hidden names, whose names begin with a period, are not matched by a bare asterisk. Red Hat's command-line documentation describes home-directory shorthand and shell expansion concepts, while the GNU Core Utilities manual documents `ls` behavior: Red Hat command-line documentation and GNU ls manual.

QUESTION NO: 9

Fill in the blank with the appropriate permission to complete the statement below.

The_____command is used to start xsession using the xinit utility.

A. startx

ANSWER: A

Explanation:

The correct command is `startx`. It is a user-friendly shell-script front end to `xinit`, the lower-level utility that initializes an X server and launches an initial X client session. Running `startx` from a text console starts the X environment using the user's startup configuration, commonly `~/.xinitrc` when present, or a system-provided default session configuration otherwise. This makes it the conventional command for manually starting an X session without a graphical display manager.

The command also passes suitable client and server arguments through to `xinit`. For example, session-related arguments can select or configure the client program, while server arguments follow `--`. The `startx` manual page explicitly describes it as a front end for `xinit` that starts an X server and runs a user's initial client program. Therefore, filling the blank with `startx` accurately identifies the command used to start an X session through the `xinit` utility. See the startx manual page and the xinit manual page.

QUESTION NO: 10

You work as the Network Administrator for McNeil Inc. The company has a Linux-based network.

You are a root user on the Red Hat operating system. You want view all files on the system that have the .xml filename extension. Which of the following commands should you use to accomplish the task?

A. locate .xml

B. locate -name /.xml

C. locate "*.xml"

D. cat .xml

ANSWER: C

Explanation:

`locate "*.xml"` is the appropriate command because `locate` searches its pathname database and the `*.xml` pattern selects pathnames whose names end with the `.xml` extension. The quotation marks are important in an interactive shell: they preserve the asterisk as part of the pattern passed to `locate`, rather than allowing the shell to expand it using files in the current directory before `locate` runs. As a result, the command can search the indexed filesystem pathnames for XML filenames regardless of the administrator's current working directory.

The results depend on the locate database, which is normally maintained by an indexing job. Therefore, files created after the most recent database update might not appear until the database is refreshed, such as by running `updatedb` with appropriate administrative privileges. Being root is useful for updating and maintaining the database, but the filename-matching syntax itself is the key part of this task. Red Hat documents the use of `locate` for quickly finding files from its database in its file-searching guidance. The detailed matching and database behavior are also described in the locate manual page.

QUESTION NO: 11

You work as a Network Administrator for Perfect Solutions Inc. The company has a Linuxbased network. Rick, a Sales Manager, has a permission issue on a file. You want to know the names of groups of which Rick is a member. You want to store the information in a text file named

RickGroup.txt. Which of the following commands will you use to accomplish the task?

A. ls rick /etc/group > RickGroup.txt

B. ls /etc/group rick > RickGroup.txt

C. grep /etc/group rick > RickGroup.txt

D. grep rick /etc/group > RickGroup.txt

ANSWER: D

Explanation:

`grep rick /etc/group > RickGroup.txt` is the appropriate command in the context of checking local group membership recorded in the system group database. The `/etc/group` file contains one entry per local group, including the group name and its explicitly listed members. Running `grep` with `rick` as the search pattern selects the entries that contain Rick's username. The shell redirection operator, `>`, writes the matching output to `RickGroup.txt`, creating the file if needed or replacing its existing contents. The resulting text file therefore provides the relevant local group entries, with each matching line beginning with the group name.

On Red Hat Enterprise Linux, local user and group account information is maintained through standard account databases, including the group database. Administrators can inspect these records when investigating access and permission issues. For a broader, name-service-aware view in modern environments, tools such as `id rick` can include groups supplied by configured identity services as well; however, the command shown correctly performs the requested search of `/etc/group` and saves it to the specified file. See Red Hat's Managing user accounts and groups documentation and the GNU grep manual.

QUESTION NO: 12

You work as a Network Administrator for Tech Perfect Inc. The company has a Linux-based network.

You run the following command on the terminal: find -name " *.conf " -exec cp {} {} .orig \ ;

What will this command do?

A. It will extract the content of configuration files from the current directory.

B. It will search configuration files from the current directory and a .orig extension.

C. It will back up configuration files from the current directory and a .orig extension.

D. It will fix other-writable files in your home directory.

E. It searches the current directory tree for names matching the literal pattern " *.conf " and runs cp with each match twice plus .orig; it does not create .orig backup files.

ANSWER: E

Explanation:

It searches the current directory tree for names matching the literal pattern `*.conf` and runs `cp` with each match twice plus `.orig`; it does not create `.orig` backup files. This is the accurate interpretation of the command exactly as written. With no starting path supplied, `find` uses the current directory. The argument supplied to `-name` is quoted, so the shell does not expand the wildcard before `find` runs. However, the spaces inside the quotes are part of the pattern, meaning it searches for file names that include a leading and trailing space around the `*.conf` pattern rather than ordinary configuration-file names. For every matching path, the `-exec` action substitutes that path for each `{}`, producing a command in the form `cp matched-file matched-file .orig`. The final argument is treated by `cp` as a destination directory when there are multiple source operands; it is not an extension appended to a file name. If a directory named `.orig` does not exist, the copy action fails. A conventional backup expression would need the destination formed as one operand, such as `{}.orig`, rather than separate `{}` and `.orig` operands. See the GNU find manual and the GNU cp invocation documentation.

QUESTION NO: 13

John works as a Network Administrator for Perfect Solutions Inc. The company has a Linux-based network. John is working as a root user on the Linux operating system. He wants to change the modified date and time of the file private.txt to 11 Nov 2009 02:59:58 am. Which of the following commands will John use to accomplish his task?

Each correct answer represents a complete solution. Choose all that apply.

A. touch -t 200911110259.58 private.txt

B. rm private.txt #11 Nov 2009 02:59:58 am

C. touch private.txt #11 Nov 2009 02:59:58 am

D. touch -d "11 Nov 2009 02:59:58 am" private.txt

ANSWER: A D

Explanation:

`touch -t 200911110259.58 private.txt` correctly sets the timestamps of `private.txt` using the numeric timestamp format accepted by `touch`: `[[CC]YY]MMDDhhmm[.ss]`. Here, `2009` is the year, `11` is November, `11` is the day of the month, `0259` is 2:59 AM, and `.58` supplies the seconds. Because no access-time-only or modification-time-only selector is specified, `touch` updates the file modification time as required (and normally its access time as well).

`touch -d "11 Nov 2009 02:59:58 am" private.txt` is also a complete solution. GNU `touch` accepts a human-readable date string with `-d`, parses the stated date and time, and applies that time to the file timestamps. This form is useful where readability matters, while the numeric form is useful for scripts and formats that require predictable input. The file must already exist unless creation is intended; the scenario states that the target is the existing file `private.txt`. See the GNU Core Utilities documentation for touch invocation and its description of date input formats.

QUESTION NO: 14

You want to run the command `backup.sh` with a lower CPU scheduling priority than normal. Which of the following commands should you use?

A. nice -n 10 backup.sh

B. renice -n 10 backup.sh

C. nohup -n 10 backup.sh

D. kill -n 10 backup.sh

ANSWER: A

Explanation:

The command `nice -n 10 backup.sh` starts `backup.sh` with a niceness value of 10. Higher niceness values lower a process's CPU scheduling priority relative to processes with the default niceness of 0. This can be useful for long-running maintenance tasks that should have less impact on interactive users or important services. See the nice(1) manual page.

QUESTION NO: 15

Which of the following commands look through the currently running processes, and list the process IDs that match the selection criteria to stdout?

Each correct answer represents a complete solution. Choose all that apply.

A. top

B. pkill

C. sort

D. pgrep

E. pidof

ANSWER: D E

Explanation:

`pgrep` is specifically designed to search the process table using criteria such as a process name, user ID, process group, terminal, parent process, and other process attributes. For every running process that satisfies the supplied criteria, it writes the matching process ID to standard output. With no formatting option specified, the normal output is one PID per line, making it suitable for command substitution, scripts, and pipelines. Multiple selection criteria can be combined; the process must meet the applicable criteria for its PID to be reported.

`pidof` also provides a complete solution when the selection criterion is the program name. It searches for running instances of the named program and prints their process IDs to standard output, normally as a space-separated list. This is useful when the objective is to find the PID or PIDs belonging to a known executable. Both commands inspect currently running processes and report identifiers rather than presenting an interactive process display or transforming arbitrary input text. See the pgrep(1) manual page and the pidof(8) manual page for supported matching behavior and output details.

QUESTION NO: 16

Which of the following commands copies files between hosts on a network and uses the same authentication as ssh, and will ask for passwords or passphrases if they are needed for authentication?

A. scp

B. rcp

C. cp

D. mv

ANSWER: A

Explanation:

The `scp` command is designed to securely copy files between a local system and a remote host, or between remote hosts. It uses the Secure Shell (SSH) transport and authentication framework, so its authentication behavior matches an SSH connection. For example, a user can authenticate with an SSH key, an SSH agent, a password, or a passphrase needed to unlock a private key. When interactive authentication is necessary and permitted by the SSH server, `scp` displays the appropriate prompt.

Remote file specifications use SSH-style syntax, such as `user@host:/path/to/file`. A typical transfer might be `scp report.txt admin@server.example.com:/home/admin/`; this copies the local file to the destination through an encrypted SSH connection. The command therefore provides confidentiality for the transferred data in transit and relies on the same host verification and user-authentication configuration used by `ssh`.

The OpenSSH manual documents that `scp` copies files between hosts on a network using SFTP or SSH, and that it accepts SSH options. Red Hat's documentation likewise describes using `scp` for secure file transfer with SSH authentication.

OpenSSH scp manual
Red Hat Enterprise Linux: Using secure communications between two systems

QUESTION NO: 17

Which of the following statements are true about open source software?

Each correct answer represents a complete solution. Choose two.

A. In OSS access to source code by third parties commonly requires the party to sign a nondiscloser agreement.

B. It is software whose license does not allow for the distribution of the software source's source code.

C. Open source software and its source code must be freely distributable.

D. In OSS all users must be able to modify the source code and create derived works.

ANSWER: C D

Explanation:

Open source software and its source code must be freely distributable because an open source license must permit free redistribution. Under the Open Source Definition, a license cannot require royalties or other fees merely for selling or giving away the software, including when it is included with a larger software distribution. The definition also requires that the program include source code and that distribution of source code be allowed in a form suitable for modification. These requirements ensure that recipients can obtain, share, and continue to distribute the software and its corresponding source code under the license terms.

In OSS all users must be able to modify the source code and create derived works because permission to alter the program and distribute derived works is a core open source licensing requirement. The license must allow modifications and derivative works, and it must allow those works to be distributed under the same license terms as the original software. This protects the practical freedoms associated with source availability: users are not limited to inspecting code, but can adapt it for their needs and share their improvements. These principles are defined by the Open Source Initiative's Open Source Definition and align with Red Hat's overview of what open source is.

QUESTION NO: 18

You work as a Network Administrator for McNeil Inc. The company has a Linux-based network. You are a root user on the Red Hat operating system. You want to know how many words are present in the file named file1.txt. You type the following command on the terminal: wc file1.txt.

The command displays the following output:

40 245 1800 file1.txt

How many words are present in the file file1.txt?

A. 40

B. 285

C. 245

D. 1800

ANSWER: C

Explanation:

The correct answer is **245**. When invoked without options, the `wc` command reports three counts for each input file, in this order: newline count, word count, and byte count, followed by the file name. Therefore, for the displayed output `40 245 1800 file1.txt`, the value `40` is the number of lines (more precisely, newline characters), `245` is the number of words, and `1800` is the number of bytes in `file1.txt`. A word for `wc` purposes is a nonzero-length sequence of characters delimited by whitespace, so this result represents 245 whitespace-separated words in the file. This command is available on Red Hat Enterprise Linux as part of the GNU core utilities and is a standard method for obtaining basic file-content counts. If only the word total is needed, the equivalent focused command is `wc -w file1.txt`, which outputs the word count without also requesting line and byte counts. See the GNU Coreutils wc invocation documentation and the Red Hat Enterprise Linux documentation for standard command-line utility usage in RHEL environments.

QUESTION NO: 19 - (SIMULATION)

Which of the following commands are used to put (paste) data from a buffer into a document in vim?

Each correct answer represents a complete solution. Choose two.

A :set

B. p

C. P

D. ps

ANSWER: Check the explanation for the answer

Explanation:

Correct answers: B. `p` and C. `P`

`p` puts (pastes) the buffered text after the cursor, or below the current line for linewise text.
`P` puts (pastes) the buffered text before the cursor, or above the current line for linewise text.

These commands are used in Vim normal mode after text has been copied or deleted into a register/buffer.

QUESTION NO: 20

You work as the Network Administrator for McNeil Inc. The company has a Linux-based

Network.

You are a root user on the Red Hat operating system. You see the following permissions on a file:

-rwxr-----

What operations can be performed on the file?

Each correct answer represents a complete solution. Choose two.

A. The file can be read by its owner.

B. The file cannot be read by its owner.

C. The file can be read by any user.

D. The file can be read by its owner and member of its owning group.

ANSWER: A D

Explanation:

The file can be read by its owner and the file can be read by its owner and member of its owning group are both correct descriptions of the displayed permission mode. The initial hyphen identifies the object as a regular file. The following three characters, `rwx`, are the permissions for the file owner: read, write, and execute. Because this owner permission set includes `r`, the owner can read the file's contents.

The next three characters, `r--`, are the permissions for the file's owning group. They grant read permission to users who are members of that group, without granting write or execute permission through the group class. Thus, a statement that includes both the owner's ability to read and the owning group members' ability to read accurately summarizes the applicable read access. The final three hyphens show that the "other users" class has no permissions from this mode.

Although the prompt identifies the administrator as `root`, the permission string itself is interpreted according to the standard owner, group, and other permission classes. Privileged processes can generally bypass normal discretionary access checks, but that does not alter what the displayed mode grants to the owner and owning group. See the chmod manual page and Red Hat's file permission documentation.

QUESTION NO: 21

You work as the Network Administrator for McNeil Inc. The company has a Linux- based network.

You are working as a root user on the Red Hat operating system. Joe is a user in your company. You want to remove Joe ' s tmp files that are 3 days older. Which of the following command syntaxes should you use to accomplish the task?

A. find /tmp -cmin +3 -user Joe -ok rm {} \;

B. find /tmp -ctime +3 user Joe -ok rm {} \;

C. find /tmp -ctime +3 -user Joe -ok rm {} \;

D. find /tmp -cmin +3 user Joe -ok rm {} \;

ANSWER: C

Explanation:

The command `find /tmp -ctime +3 -user Joe -ok rm {} \;` searches beneath `/tmp` for files owned by the user `Joe` whose inode status-change time is older than three 24-hour periods, then interactively offers to remove each matching file. The `-user Joe` test restricts the search to files owned by that account. The `-ctime +3` test uses age measured in 24-hour units; a leading plus sign selects files with a status-change age greater than the stated value. This matches the question's required day-based age criterion and its use of `ctime`.

The `-ok` action runs the specified command separately for each result and asks the administrator for confirmation before executing it. This provides a useful safety measure for a deletion task, particularly in a shared temporary directory. Within the action, `{}` is replaced by the pathname of the current file. The terminating semicolon marks the end of the action for `find`; escaping it as `\;` prevents the shell from treating the semicolon as its own command separator. Red Hat documentation describes `find` tests and actions, while the GNU Findutils manual documents the time tests and interactive `-ok` action: Red Hat documentation and GNU Findutils manual.