



MODULE 02: ESSENTIAL LINUX COMMANDS

DevOps end to end services & solutions

Dinghye2e

© Copyright <https://dinghy-e2e.com> | Yehuda Shoshani

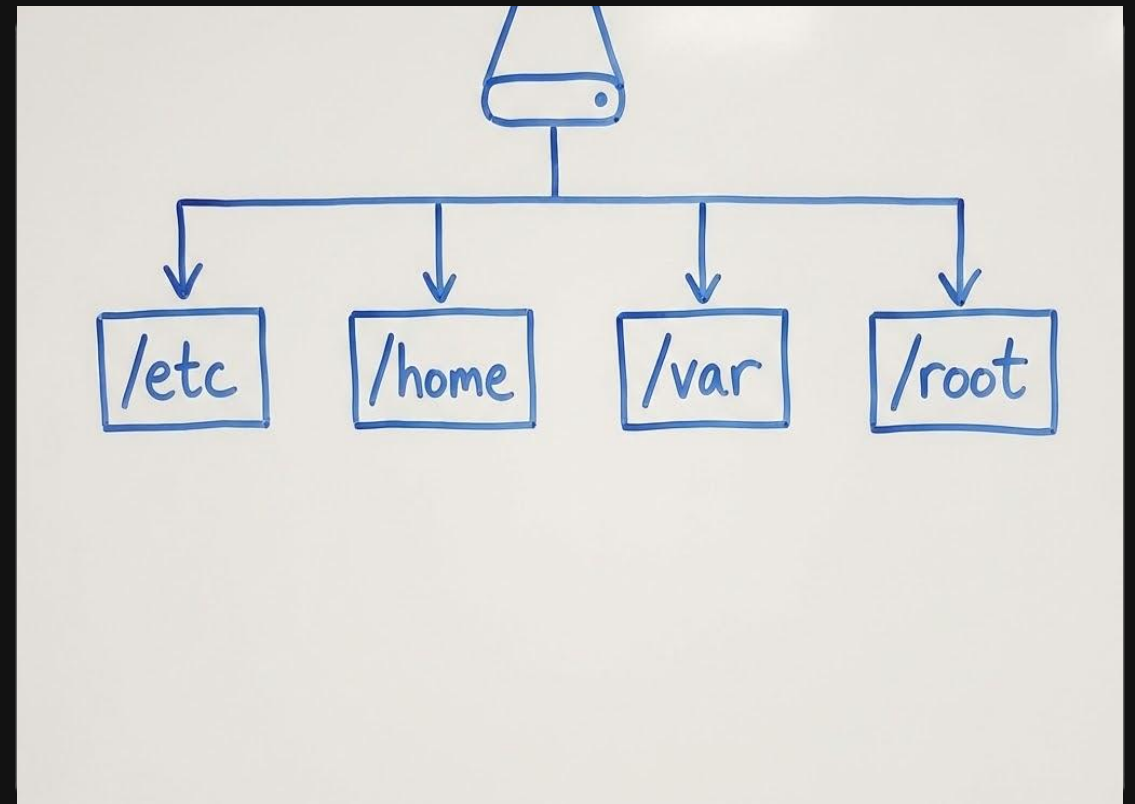
Agenda

- > Structure folders and files
- > Linux Shell (bash, sh, zsh, etc.)
- > Essential Linux Commands (sudo, pwd, cd, ls, etc.)
- > File Operations (mkdir, rmdir, cp, mv, touch)
- > Viewing & Editing Files (cat, echo)
- > Hands-on Solution

Linux Directory Structure

The Linux directory structure is laid out like a tree with the **root (/)** directory being the trunk.

- > **/bin:** Binary or executable programs.
- > **/etc:** System configuration files.
- > **/home:** User home directories.
- > **/var:** Variable files (logs, etc.).



Linux Shell

>_What is it?

A command-line interpreter that provides a user interface for the OS. It executes commands and scripts.

Common Shells

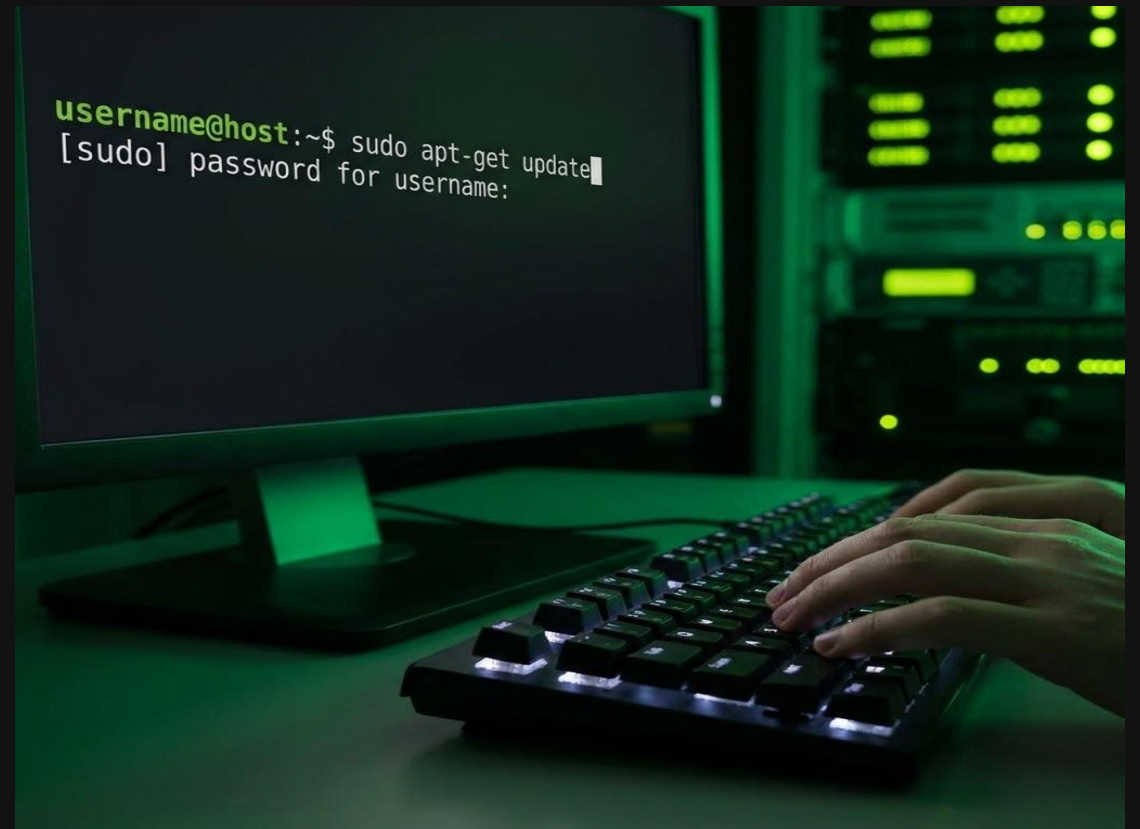
- **bash:** Bourne Again Shell (Most common).
- **sh:** The original Bourne shell.
- **zsh:** Z Shell (Modern, feature-rich).
- **ksh:** Korn Shell.

The “sudo” Command

SuperUser DO

Allows a permitted user to execute a command as the superuser (root) or another user.

```
# Update package list  
sudo apt-get update
```



Navigation & System Information: pwd, cd & date



pwd

Print Working Directory

Displays the full path of the current directory.

```
$ pwd  
/home/user
```



cd

Change Directory

Navigates to a different directory.

```
cd ~ (home)  
cd .. (up one level)  
cd - (previous dir)
```



date

Display Date & Time

Shows the current system date and time.

```
$ date  
Thu Oct 26 10:00:00 UTC 2023  
$ date "+%Y-%m-%d"  
2023-10-26
```

Listing Files: ls

The ls command lists files and directories within the file system.

- > ls: Simple list.
- > ls -l: Long listing format (permissions, owner, size).
- > ls -a: Show hidden files (starting with .).
- > ls -R: Recursive list.

```
$ ls -la
drwxr-xr-x 2 user user 4096 Sep 19 08:23 .
drwxr-xr-x 20 user user 4096 Sep 19 08:20 ..
-rw-r--r-- 1 user user 21 Sep 19 08:23 .bashrc
-rw-r--r-- 1 user user 100 Sep 19 08:23 file.txt
```

File Manipulation



mkdir

Create directories.
`mkdir -p dir1/dir2`



touch

Create empty files or update
timestamps.
`touch newfile.txt`



rm / rmdir

Remove files or
directories.
`rm -rf folder/`
(Use with caution!)

Move & Copy

cp (Copy)

Copy files or directories.

```
# Copy file
cp source.txt dest.txt
# Copy directory recursively
cp -r src_dir/ dest_dir/
```

mv (Move)

Move or rename files and directories.

```
# Rename file
mv old.txt new.txt
# Move file to directory
mv file.txt /tmp/
```

Viewing Content

cat

Reads files sequentially, writing them to standard output.

```
$ cat hello_script.sh
#!/bin/bash
echo "Hello World"
```

echo

Displays a line of text. Often used to write to files.

```
# Write to file (overwrite)
echo "Line 1" > file.txt
# Append to file
echo "Line 2" >> file.txt
```

LAB-01: Essential Linux Commands (Hands-on)

1. Setup & Navigation

- ▢ Display full path
- ▢ Navigate to system logs (/var/log)
- ▢ Return to home (~ / home / my-user) [shortest cmd]

3. Content Manipulation & Verification

- ▢ Create myfile.txt in 'a' with text: "I am a great devops" [one command]
- ▢ Append to myfile.txt: "and I would like to be an expert on Linux" [one command]
- ▢ Print the file content

2. File & Directory Creation

- ▢ Create directories (a, b, c) [one command]
- ▢ Display date & time (date)

4. Advanced Operations & Cleanup

- ▢ Create 3 empty files (file1.txt, file2.txt, file3.txt) in 'a' [one command]
- ▢ Copy 'a' folder to 'b' [recursive cp]
- ▢ List ~/home/my-user/b/a contents (permissions, hidden)
- ▢ Rename 'c' to 'd', create placeholder.txt in 'd'
- ▢ Navigate home, remove 'a, b, d' folders [one command]
- ▢ Display command history & count



Bonus Challenge

Create a folder called 'bonus' under ~/home/

Questions?

Thank you for participating.

Feel free to reach out:

<https://dinghy-e2e.com>