__/__/__

Session 1 - Core Command

1) Directory navigation & file handling

- What is a directory?

→ Folder, starts with root

Commands & mention

- pwd - print working directory
  - ↳ shows path
- ls - list files
- ls -l - lists files with details
- cd - change directory
  - ↳ ~ → home directory
  - (we can switch at any point)
- mkdir - make directory
  - ↳ either using path or when your standing there
- touch - create files
- mv filename path folder (use pwd)
  - ↳ move file there

We can also rename

mv filename newfile

↳ changes name

- cp filename path

↳ copies a link

2) Hardlink - Makes another file/folder to access the same memory location so deleting this won't erase that.

3) Softlink - Link of that folder so if the original removed, no other one also can't work

- Softlink • ln -s file/folder target
- Hardlink. ln file/folder target

8) File Permission and Ownership

- Permission can be given to 3 - user / owner
  group
  others
- 3 permissions can be given r - read
  w - write
  x - execute

r w x		
0 0 0	- 0	Nothing
0 0 1	- 1	Only execute
0 1 0	- 2	Only write
0 1 1	- 3	write and execute
1 0 0	- 4	only read
1 0 1	- 5	Read and execute
1 1 0	- 6	Read and write
1 1 1	- 7	All can be done

Commands

- chmod permission filename
  ↳ change the permission
  - → file
  d → directory

	owner	group	other
1	3	3	3
dir			

- sudo - superuser permission
- sudo chown user filename
  (we need permission ownership goes)
- ACL - Access Control List
  (advanced permissions)
  ↳ give access specifically to one user without changing that files owner or group
- sudo setfacl -m u:user:r filename
  set ACL rules modify give this user to read

__/__/__

(gives ~~of~~ user permission to read)

- getfacl file
  ↳ prints all the ACL permissions on that file

3) System status and resource information
- ~~pwd~~ who - shows all the logged-in users
- uptime - Tells us how long system has been running and other details like that
- free -h - Shows the memory usage
- df -h - Shows disk space
- top - running processes are displayed
- uname -r - Tells us version of kernel and other details

Section 2 - User, Group and Permission Management

4) User and Group Administeration
- User - an account to log in and use your system
- Each user has its own - Username
  - Directory
  - Default group
  - User ID
  - Password (use set it)

- Group is a collection of users
- We use this to give permission to multiple users together (ACL can be used tho)

Commands
- sudo groupadd groupname
  needs (sudo) creates (group)

- sudo usermod -aG ~~data~~ groupname username
  - modify users setting
  - append to group
  (without a it just erases and overwrites)

- sudo mkdir /path
  ↳ makes shared folder in a path
- sudo chown :group path
  ↳ we change ownership from (since nothing before :, current user to group and then directory
  (Changes group owner of database to datateam)
- sudo chmod permission path
  ↳ changes permissions

- SGID - this ensures file made in directory automatically gets the group of that directory
- sudo chmod g+s path
  group + set SGID bit

Bonus
To check user, groupID
- id username → gives
- groups username → we see all groups
(we can use ls -l to see file's group)

5) File Security and Auditing
- auditd
  ↳ Linux Auditing System
  (records security events)
  ↳ Access
    Modification
    failures

- we can - detect unauthorized access and investigate who did it
- sudo auditctl -w path -p war -k vaultwatch
  - -w: live auditing, watch
  - -p: write, attribute change, read
  - -k: tag for the watching
- so we log every time someone - Reads / Writes into / a / Deletes / Changes owner / permissions
- ausearch -k vaultwatch
  - ↳ we search and we can see (all suspicious activities can be found out!)

Bonus

- ausearch -k vaultwatch | grep opened
  - ↳ shows opened files
- ausearch -k vaultwatch | grep denied
  - ↳ who accessed it [SECURITY]

(same for others, & put it in a .log file and then do grep)

6) Network Diagnostics

- ping - checks if a host can be reached, sends packets and checks if it gets a reply, here & save it to netcheck.txt
- ip a - we can see all network interfaces
- ip route - shows our router (default gateway)
- cat /etc/resolv.conf
  - ↳ shows the DNS servers we given DNS ↔ IP
- ss -tuln - to see open ports (Bonus we also saved tcp ones using grep into text file)
  - TCP / UDP / listening / numeric output

7) File checker script

- We open a script file using nano file_inspector.sh → goes into shell
- chmod +x fileinspector.sh
  change file permission → add permission x (execute)
- then we run the script
  ./file_inspector.
- ACC to the SHELL script
  We take the rest of commands
  read -p "Enter file" f → file to variable
  to read ~~pdf~~ to print file before taking
- cat inspector_log.txt
  ↳ print it
- Then if else fi
  (then) ↳ end
- $(date) → gets date input and prints in echo
- #!/bin/bash
  ↳ start every shell script
  (tells OS to run script)

☆ Read also supports -n - character limit
-t - timeout (t seconds before stop taking i/p)
-s - hidden password

- [-f "$filename"]
  ↳ variable for file
  -f - if file exists
  -d - if directory exists
  -e - either exists
  -L - link

☆ if [ ]; then

8) System monitoring script

* We can just create and run scripts using
  nano a.sh → open shell script
  chmod +x a.sh → modify execution
  ./a → run
  cat a.sh → print it

* ~~crontab didn't~~

- In the shell script
  We add it to it and print according to commands (both explained above) and print

- crontab -e
  ↳ editor to schedule & run task in an interval
  and we schedule every 30 minutes by adding a line
  */30 * * * * path

* time stamp "+%Y-%m-%d %H:%M:%S"
  ↳ is the format

9) Package operations

- APT - Advanced Package Tool (Debian)
  (Package manager)
  ↳ install, update, remove, manage
- sudo apt install update
  ↳ downloads latest package from repository
  (ONLY UPDATES & LIST OF AVAILABLE)
- sudo apt upgrade -y
  (newer version of currently installed)
  (INSTALLS NEW)
- sudo apt install curl
  ↳ Downloads package & install
- dpkg --list >> pkg_list.txt
  lists all installation and puts

Bonus

- dpkg-query -W --showformat '${Installed-Size} ${Package}\n'
  detailed info about package
- sort -nr
  ↳ numerically in reverse
- head -5
  ↳ top 5 entries

★ installed size vs download size
↳ Compressed when downloaded
↳ uncompressed when installed

10) Process and Service Control

- Process - Running instance of a program
  Every process has a PID, state
  ↳ running, sleeping, zombie
  ↳ finished but still entered on
- ps aux - lists all processes
  ↳ a u x
  shows process from all users; u = user oriented format; x = include background processes
- ps aux --sort=-%cpu | head → default 10
  ↳ top CPU consuming processes
- top - live monitor of processes
- ~~kill~~ kill
  ↳ kills a ~~one~~ process
  (unresponsive, stuck, zombie)
  (Didn't implement was scared)

__/__/__

- Service - Long running background process handling remote connection, web & database server and to schedule tasks. (systemd managed)
  ↳ automatically started on booting

- systemctl status ssh
  - used for controlling & checking systemd
  - current status
  - secure shell (checking remote network)
- ⇒ sudo systemctl start ssh - start a service
- sudo systemctl restart ssh - restart a service
  - stop - stop a service
  - enable - enable service on boo
  - disable

Bonus

- we use nano to make a shell script
- ! systemctl is-active --quiet "$SERVICE";
  not active service then suppress it
  (zero exit status)

☆ We use crontab to update 10 minutes once

- systemd - starts service
  manages processes
- KILL- 15 - gentle stop
  9 - force stop
  2 - interrupt
  1 - reload

__/__/____

11) Web Server Setup
- We install apache using sudo then start and enable using following commands
- sudo apt install apache2 -y
- sudo systemctl start apache2 - start
- sudo systemctl enable apache2 - enable
- systemctl status apache2 - then check status
- now visit website and find it in default web page location /var/www/html/
- now we replace default index page
echo "welcome" | sudo tee ____
writes as root ↳ to html
- Web Server - Hosts APIs and webpages and listens to HTTP
- apache2 - helps HTTP server and service management files
- ufw - uncomplicated firewall
↳ allow to open take ports blocked by default
~~then enable~~
ufw enable - enables it
ufw status - gives status & rules list

12) Automated Backup Script
- We create shell daily_backup.sh and usual shell steps
- tar -czf file.tar.gz path → Compress dir to
↓ ↳
ziptool C create
z gzip extention
f - file

__/__/__

find ~/Backups -type f -mtime +10 -delete
search for files — only files — older than 10 days — ↓ remove

— Search files more than 10 days and remove it

13] Text Utilities Pratice

- We first make directories and files
- Grep - Global Regular Expression print
  ↳ Searches text in files
- grep hello *.txt
  ↳ searches hello in all files
  (others are same)
- Awk - ~~text processing~~ to take column wise (based on spaces)
- $ awk '{ print $2 }' data.txt
  ↳ prints second word
- Sed - stream editor
  (edits text)
- ☆ Sed -i 's/hello/hi/g' *.txt
  edit in place — replace hello with hi globally in all