1. File System Management

1. Create a directory named `project_files`.
2. Navigate to the directory using `cd`.
3. Create an empty file named `readme.txt`.
4. Copy `readme.txt` to a new file called `info.txt`.
5. Move `info.txt` to your home directory.
6. Delete `readme.txt`.
7. Display the current directory path using `pwd`.
8. View the list of files including hidden ones using `ls -la`.
9. Check disk usage with `df -h`.
10. Check file space used by the `project_files` directory using `du -sh project_files`.

---

2. User and Group Administration

1. Create users `user1`, `user2`, and `user3`.
2. Create a group named `devteam`.
3. Add `user1` and `user2` to `devteam`.
4. Create a folder `/home/devteam`.
5. Create 2–3 files into this folder.
6. Change group ownership of `/home/devteam` to `devteam`.
7. Set permissions so only group members can read/write: `chmod 770`.
8. Log in as `user1` and create a file inside the folder.

9. Log in as `user3` (not in group) and try accessing the folder (should be denied).
10. Delete the user `user3`

---

3. Package Management with Yum

1. Search for the `nano` text editor using `yum search nano`.
2. Install `nano`.
3. Remove `nano`.
4. List all installed packages.
5. Get details about the `httpd` package.
6. Install Apache (`httpd`).
7. Start the Apache service: `systemctl start httpd`.
8. Enable it at boot: `systemctl enable httpd`.
9. Visit `http://localhost` to verify.
10. Stop the Apache service.

---

4. System Services and Systems

1. View all active services using `systemctl list-units --type=service`.
2. Start the `httpd` service.
3. Enable `httpd` at boot.
4. Stop the `httpd` service.
5. View the status of `httpd` .
6. Restart the `httpd` service.

7. Disable `httpd`.
8. Use `journalctl -u httpd` to view logs.
9. Reload a running service with `systemctl reload httpd` .
10. Check which services failed using `systemctl --failed`.

---

5. System Performance Monitoring + System Info

1. Run `top` to monitor system processes.
2. Use `htop` for a user-friendly view (install if needed).
3. Use `free -h` to view RAM and SWAP.
4. Run `uptime` and extract just the load average using:
   `uptime | awk '{print $10, $11, $12}'`
5. Get architecture info with `uname -m`.
6. Display kernel version using `uname -r`.
7. Show OS type with `uname -o`.
8. View logged-in users with `who`.
9. Show disk usage with `df -h`.
10. Use `du` to check any folder size.

---

6. Process Management

1. Run `ps` and `ps aux` to list all currently running processes.
2. Create a shell script (`test.sh`) that runs an infinite loop:

```
#!/bin/bash
```

```
while true; do

echo "Running..."

sleep 5

done
```

3. Start the script in the background:
   `sh test.sh &`

4. View background jobs using:
   `jobs`

5. Bring the script to the foreground:
   `fg %1`

6. Suspend the running script with `Ctrl + Z`, then send it back to background:
   `bg %1`

7. Kill the background process using its PID:
   `kill <PID>`

8. Kill the script by name using:
   `pkill test.sh`

9. Find the script's PID using:
   `pgrep -f test.sh`

10. Run the script in the background and make it immune to terminal closure:
    `nohup sh test.sh &`

---

7. Text Processing (Grep, Awk, Sed, Find)

1. Use `grep "root" /etc/passwd` to find lines with root.

2. Use `awk -F: '{print $1}' /etc/passwd` to list usernames.

3. Find all `.conf` files in `/etc` using `find /etc -name "*.conf"`.

4. Replace "Apache" with "Nginx" in a test file using `sed`.
5. Use `grep "^user"` to find lines starting with "user".
6. Delete line 3 of a file using `sed '3d' file.txt`.
7. Use `find /home/ec2-user -type f -mtime -1` to locate recent files.
8. Extract usernames and shells from `/etc/passwd` using `awk -F: '{print $1, $7}'`.
9. Use `grep -i "error" /var/log/messages` to find error entries.
10. Count `.txt` files in shared folder:
    ```
    find /home/ec2-user -name "*.txt" | wc -l
    ```

---

8. Backup and Restore

1. Create a directory `backup_test`.
2. Add a few files to the directory.
3. Create a backup using `tar -cvf backup.tar backup_test`.
4. Compress the backup using gzip.
5. Extract the backup using `tar -xvzf`.
6. Copy the backup to another location.
7. Create a daily backup script.
8. Schedule the script with crontab.
9. View current `crontab` jobs.
10. Test restoring a single file from the archive.
