Day2 Assignment

Task 1: linux command line basics

Step 1: To create a directory as shown below in image the comammand used is: **mkdir -p devops-lab/{projects/{app1,app2},configs,logs}.**

Step 2: Navigating to /devops-lab/projects/app1 and creating a file called README.md

Step 3: Copying file README.MD from app1 directory to app2 directory.

```
┌──(kali㉿kali)-[~]
└─$ cp devops-lab/projects/app1/README.md devops-lab/projects/app2/

┌──(kali㉿kali)-[~]
└─$ cd devops-lab/projects/app2

┌──(kali㉿kali)-[~/devops-lab/projects/app2]
└─$ ls
README.md
```

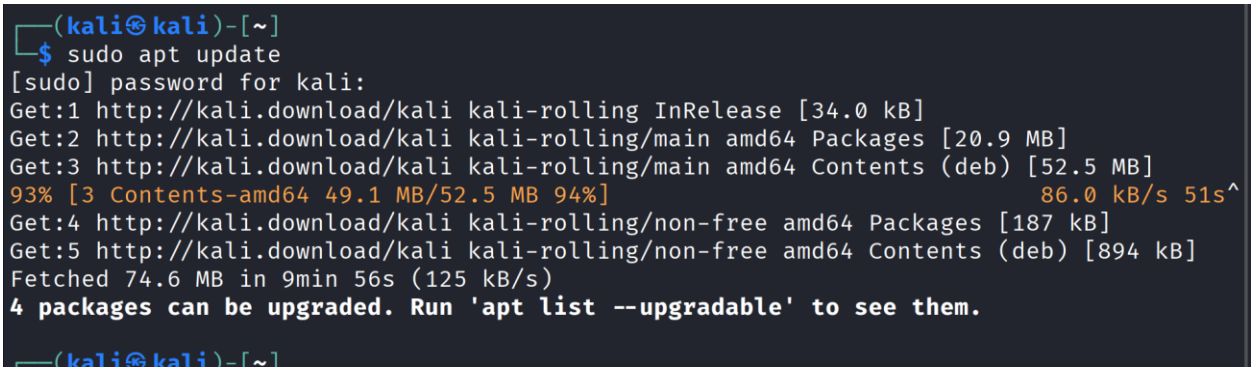
Step 4: Listing all files in devops-lab recursively showing hidden file.

```
┌──(kali㉿kali)-[~/devops-lab]
└─$ ls -laR
.:
total 20
drwxrwxr-x  5 kali kali 4096 Nov 17 23:02 .
drwx------ 19 kali kali 4096 Nov 18 16:48 ..
drwxrwxr-x  2 kali kali 4096 Nov 17 23:02 configs
drwxrwxr-x  2 kali kali 4096 Nov 17 23:02 logs
drwxrwxr-x  4 kali kali 4096 Nov 17 23:02 projects

./configs:
total 8
drwxrwxr-x 2 kali kali 4096 Nov 17 23:02 .
drwxrwxr-x 5 kali kali 4096 Nov 17 23:02 ..

./logs:
total 8
drwxrwxr-x 2 kali kali 4096 Nov 17 23:02 .
drwxrwxr-x 5 kali kali 4096 Nov 17 23:02 ..

./projects:
total 16
drwxrwxr-x 4 kali kali 4096 Nov 17 23:02 .
drwxrwxr-x 5 kali kali 4096 Nov 17 23:02 ..
drwxrwxr-x 2 kali kali 4096 Nov 17 23:13 app1
drwxrwxr-x 2 kali kali 4096 Nov 17 23:16 app2

./projects/app1:
total 8
drwxrwxr-x 2 kali kali 4096 Nov 17 23:13 .
drwxrwxr-x 4 kali kali 4096 Nov 17 23:02 ..
-rw-rw-r-- 1 kali kali    0 Nov 17 23:19 README.md

./projects/app2:
total 8
drwxrwxr-x 2 kali kali 4096 Nov 17 23:16 .
drwxrwxr-x 4 kali kali 4096 Nov 17 23:02 ..
-rw-rw-r-- 1 kali kali    0 Nov 17 23:16 README.md
```

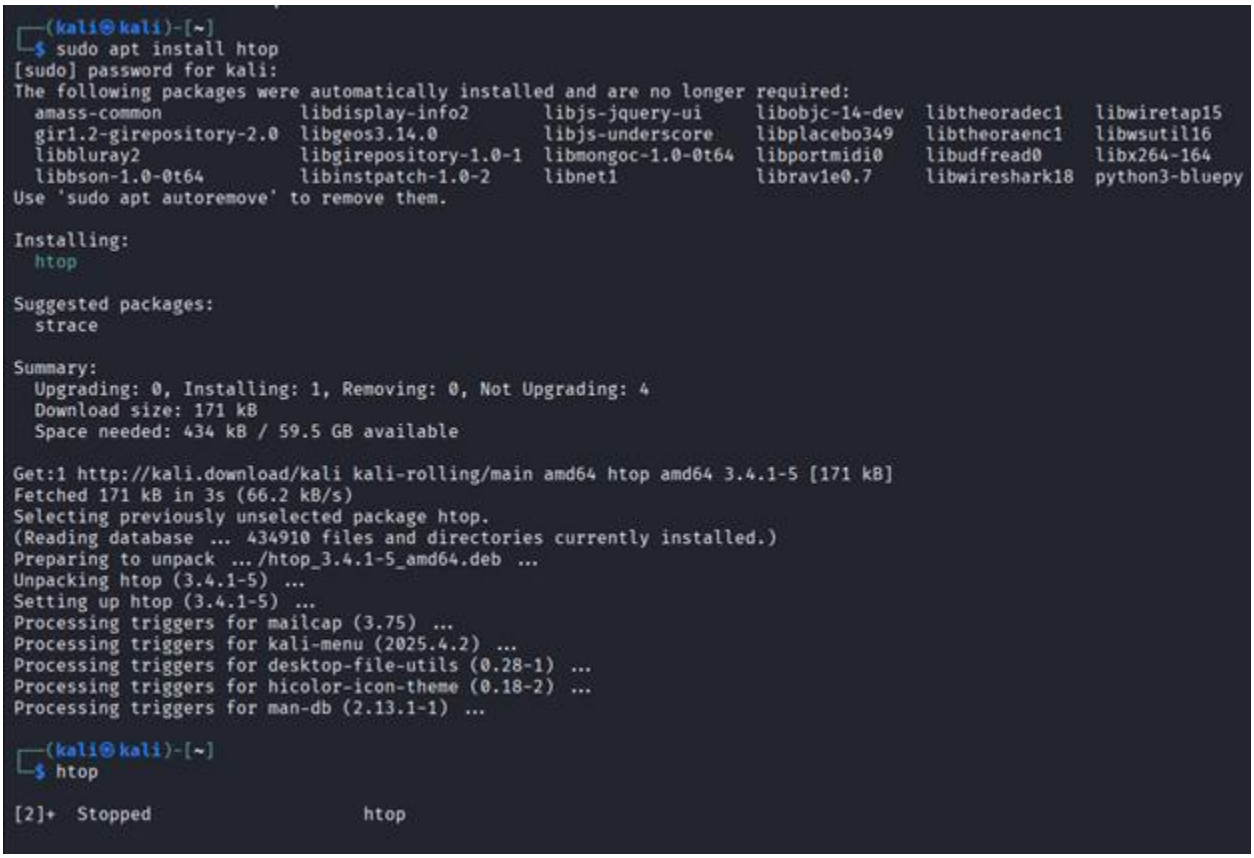
Where, **-a** → show hidden files (files starting with **.**) & **-R** → recursive (go inside all subdirectories)

Task 2: Package Management

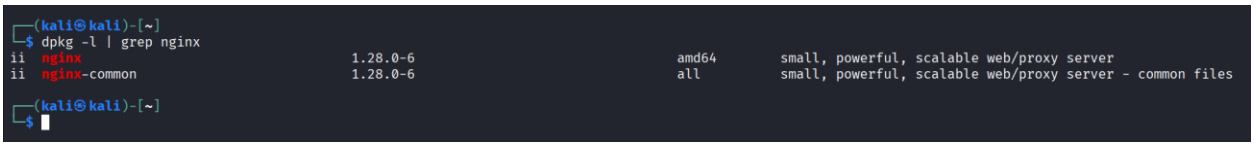
Step 1: Updating Package indexing.



Step 2: Installing htop (system monitoring tool)



Step 3: Checking if nginx service exists.



Where,

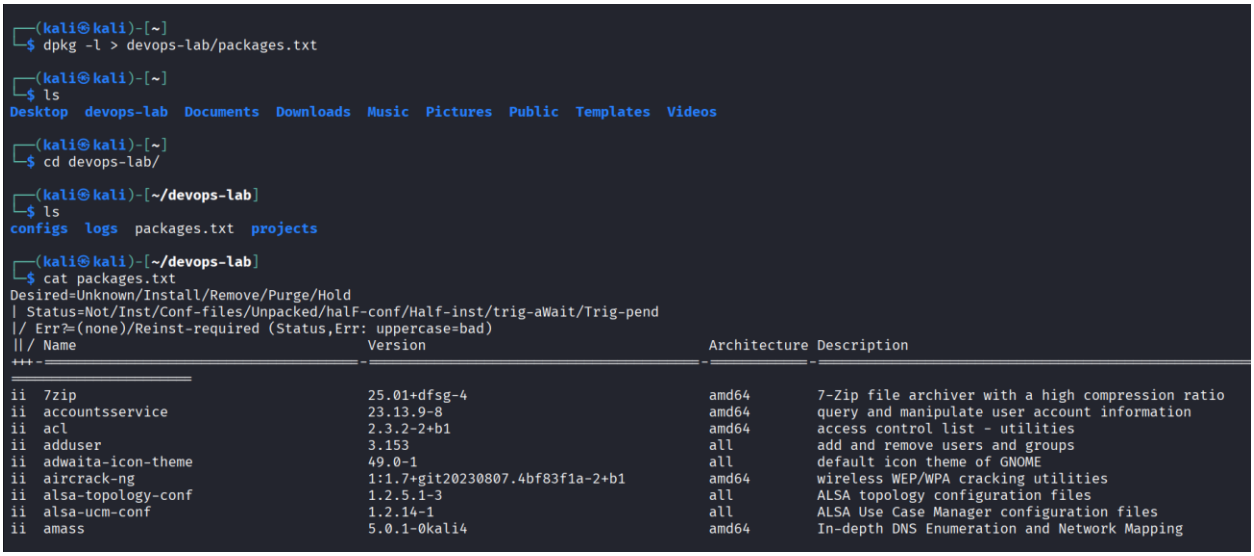
dpkg → Debian package manager (used in Ubuntu, Kali, Debian, etc.)

-l → list all installed packages on your system

| → Take all the text output and filter it through the next command."

grep → searches text for a pattern, Here, the pattern is nginx

Step 4: Listing all currently installed packages and saving the output to devops-lab/packages.txt.



Task 3: DORA thresholds

Given,

Total deployments (month): **40**

Deployments causing incidents: **6**

Working days: **30**

Average lead time (commit → production): **3 hours**

Incidents (detected → resolved):

1. 10:00 → 11:30 → **1.5 h = 90 min**
2. 14:00 → 14:45 → **0.75 h = 45 min**
3. 09:00 → 11:00 → **2.0 h = 120 min**
4. 16:00 → 20:00 → **4.0 h = 240 min**
5. 13:00 → 13:30 → **0.5 h = 30 min**
6. 11:00 → 15:00 → **4.0 h = 240 min**

Sum of incident recovery times = 90 + 45 + 120 + 240 + 30 + 240 = **765 minutes**

Number of incidents = 6

1) Deployment frequency (per day)

Formula: total deployments / working days

40 deployments ÷ 20 days

= 2 deployments per working day

2) Lead Time for Changes

Given directly: **3 hours (average)**

Answer: 3 hours (commit → production)

3) Change Failure Rate (%)

Formula:

(Deployments causing incidents / total deployments) × 100

= (6 ÷ 40) × 100

= 0.15 × 100

= 15%

4) Mean Time To Recovery (MTTR)

Formula: **average of incident recovery durations**

Total minutes = 765
MTTR (minutes) = 765 ÷ 6 = 127.5 minutes

Convert to hours/minutes:
= 127.5 minutes

= 2 hours + 7 minutes + 30 seconds

= 2.125 hours

5) Classification (Elite / High / Medium / Low) — based on common DORA bands

Deployment frequency

Elite: multiple deploys per day

This team: **2/day → Elite**

Lead time for changes

Elite: minutes → hours (very short)

This team: **3 hours → Elite**

Change failure rate

Elite: **0–15%** (≤15% as top tier)

This team: **15% (Elite/High MTTR)**

Elite: < 1 hour

High: < 1 day (hours)

This team: **~2.13 hours → High**