Docker task - 2

1. Install docker to your system :

```
    #Add Docker's official GPG key:
```

sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

```
#Add the repository to Apt sources:
```

echo
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu
(. /etc/os-release && echo "VERSION_CODENAME") stable" |
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

[This Setup Docker's apt repository in your system]

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

[This command install latest version docker in your system]

2. Creation of ubuntu image pulling it from dockerhub :

```
    sudo docker pull ubuntu:22.04

  output:

  22.04: Pulling from library/ubuntu
Digest: sha256:0e5e4a57c2499249aafc3b40fcd541e9a456aab7296681a3994d631587203f97
Status: Image is up to date for ubuntu:22.04
```

docker.io/library/ubuntu:22.04

```
  [It pulls the ubuntu image from the dockerhub to create an ubuntu in our system,
  image is essential to create an container in our system]
```

```
 sudo docker images

output:

 REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
 ubuntu       22.04     97271d29cb79   5 weeks ago   77.9MB

[This command shows the number images and its details available in our system]
```

3. Creation of container from the image and naming the container with it, Verifying :

```
sudo docker run -d --name ubuntu_container_1 ubuntu:22.04
```

output:

e8b23ed8606d17541bf554ba2906e5403632dcc27b7eab720c06cb845654b0ed

[This command creates an container in detached mode with a name ubuntu_container_1 with an image ubuntu 22.04, the image will be taken from our local machine]

```
sudo docker ps -a
```

output:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e8b23ed8606d	ubuntu:22.04	"/bin/bash"	28 seconds ago	Exited (0) 26 seconds ago		ubuntu_container_1

[This command shows the status of the containers available in our system]

4. Creation of container with a feature of interaction with terminal :

```
sudo docker run -it --name interactive_container ubuntu:22.04
```

output:

root@065572e406c1:/#

[This command creates a container which interacts with terminal, here we directly entered inside the container after creation because we didn't use the command "-d" here.. which

will dettach us from the container after its creation]

Press ctrl+D followed by ctrl+O to dettach from the container after logged in or type "exit" command to dettach

5. Execution of commands for inside container from our local machine :

```
sudo docker run -d --name exec_container ubuntu:22.04 sleep 600
```

output:

05252ea6e7f378b4a246f734f0519a6d697753e1538e5624a379935596b7f652
[This command creates an container which will run in background for 600 seconds]

```
sudo docker exec -it exec_container bash
```

output:

root@05252ea6e7f3:/#
[This command means, docker runs the execution process for "exec_container" container with the allowance of interactive terminal mode. bash shell is opened inside container through execution process]

6. Checking logs of an container :

```
sudo docker logs exec_container
```

output :
If any logs are present then that details will be shown in our local machine terminal, if not it'll output an empty space

sample output :
Starting application...
Error: Missing configuration file
Application failed to start

7. Attach to an container :

```
 sudo docker run -d -it --name attach_container ubuntu:22.04
```

output :
4031e237ffeb28b8adb661f7148796865fee46d993e0aff30a29671a3a1608d5

[It creates a container in detached mood with the allowance of interaction with terminal which runs in the backgroud, that is the container willl be in the "up" state]

```
sudo docker attach attach_container
```

output :
root@4031e237ffeb:/#

[This command makes us to attach or enter into the container "attach_container"]