CU CHANDIGARH UNIVERSITY UNIVERSITY INSTITUTE of COMPUTING *Asia's Fastest Growing University* NAAC GRADE A+ ACCREDITED UNIVERSITY

Experiment No. 3.3

Student Name: Gaurav Kumar
Branch: MCA–CCD
Semester: III
Subject Name: CONTAINERIZATION WITH DOCKER

UID: 22MCC20177
Section/Group: MCD-1/A
Date of Performance: 15th Oct 23
Subject Code: 22CAH-742

1. Aim/Overview of the practical:

a) Cleaning Up Old Containers and Docker Images.

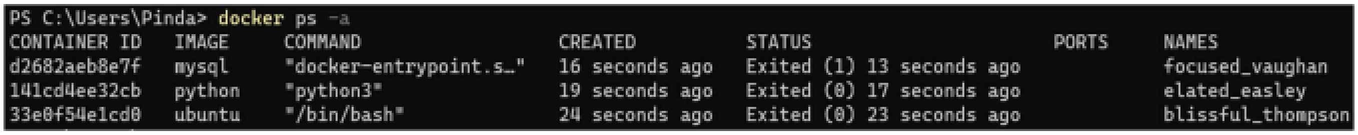
2. Code for practical: (a)

To clean up old containers and Docker images, you can use the following commands:

- docker container prune: Removes all stopped containers.
- docker image prune: Removes all dangling images.
- docker system prune: Removes all unused images, containers, volumes, and networks.

Step 1 : First Cleaning up Old Container. Then First check containers that are available:

```
$docker ps -a.
```

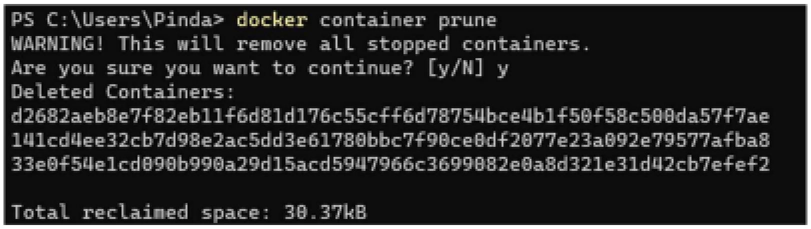


```
PS C:\Users\Pinda> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS     NAMES
d2682aeb8e7f   mysql     "docker-entrypoint.s…"   16 seconds ago   Exited (1) 13 seconds ago             focused_vaughan
141cd4ee32cb   python    "python3"                19 seconds ago   Exited (0) 17 seconds ago             elated_easley
33e0f54e1cd0   ubuntu    "/bin/bash"              24 seconds ago   Exited (0) 23 seconds ago             blissful_thompson
```

Step 2 : Then use docker container prune to clean up old container:

```
$docker container prune
```

- There will be WARNING that This will remove all stopped containers.
- Are you sure you want to continue? [y/N]
- Press Y for yes else N, then Enter.



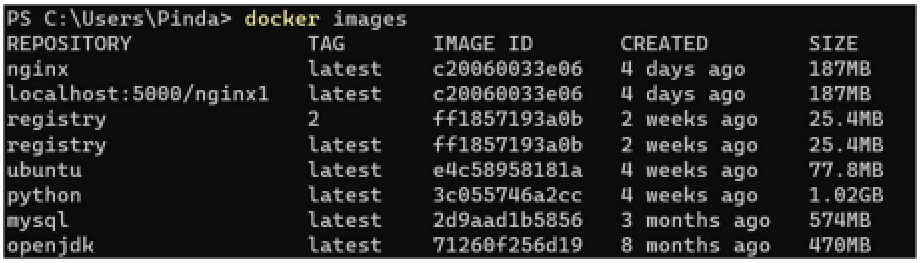
```
PS C:\Users\Pinda> docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
d2682aeb8e7f82eb11f6d81d176c55cff6d78754bce4b1f50f58c500da57f7ae
141cd4ee32cb7d98e2ac5dd3e61780bbc7f90ce0df2077e23a092e79577afba8
33e0f54e1cd090b990a29d15acd5947966c3699082e0a8d321e31d42cb7efef2

Total reclaimed space: 30.37kB
```

Step 3 : **Cleaning up Old Images:**

Step 4 : First check images that are available:

```
$docker images
```

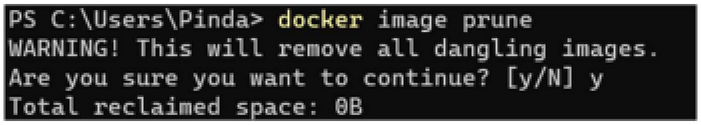


```
PS C:\Users\Pinda> docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
nginx                    latest    c20060033e06   4 days ago      187MB
localhost:5000/nginx1    latest    c20060033e06   4 days ago      187MB
registry                 2         ff1857193a0b   2 weeks ago     25.4MB
registry                 latest    ff1857193a0b   2 weeks ago     25.4MB
ubuntu                   latest    e4c58958181a   4 weeks ago     77.8MB
python                   latest    3c055746a2cc   4 weeks ago     1.02GB
mysql                    latest    2d9aad1b5856   3 months ago    574MB
openjdk                  latest    71260f256d19   8 months ago    470MB
```

Step 5 : Then use docker image prune to clean up old dangling images:

```
$docker image prune
```

- There will be WARNING that This will remove all dangling images.
- Are you sure you want to continue? [y/N]
- Press Y for yes else N, then Enter.

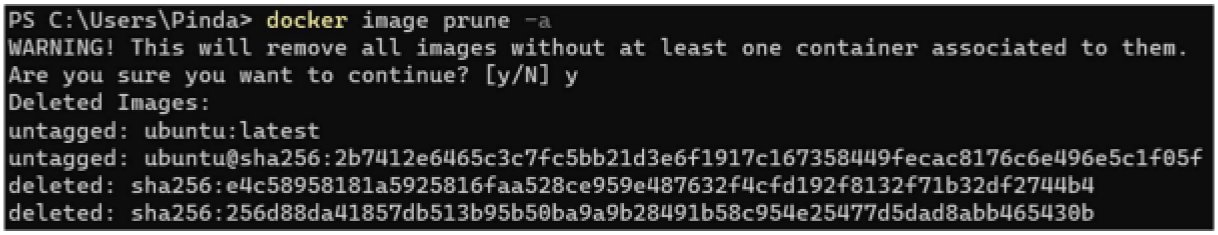


```
PS C:\Users\Pinda> docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

Step 6 : Then use docker image prune to clean up all available images:

```
$docker image prune -a
```

- There will be WARNING that This will remove all images.
- Are you sure you want to continue? [y/N]
- Press Y for yes else N, then Enter.



```
PS C:\Users\Pinda> docker image prune -a
WARNING! This will remove all images without at least one container associated to them.
Are you sure you want to continue? [y/N] y
Deleted Images:
untagged: ubuntu:latest
untagged: ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
deleted: sha256:e4c58958181a5925816faa528ce959e487632f4cfd192f8132f71b32df2744b4
deleted: sha256:256d88da41857db513b95b50ba9a9b28491b58c954e25477d5dad8abb465430b
```