Lab Exercise 1

Name = Harsh Sharma
SAP = 500097351

Performing CRUD Operation on Containers

Objective: Performing CRUD Operation on Containers

Tools required: Docker Configuration

Pre-requisites: Ubuntu Configuration, Docker

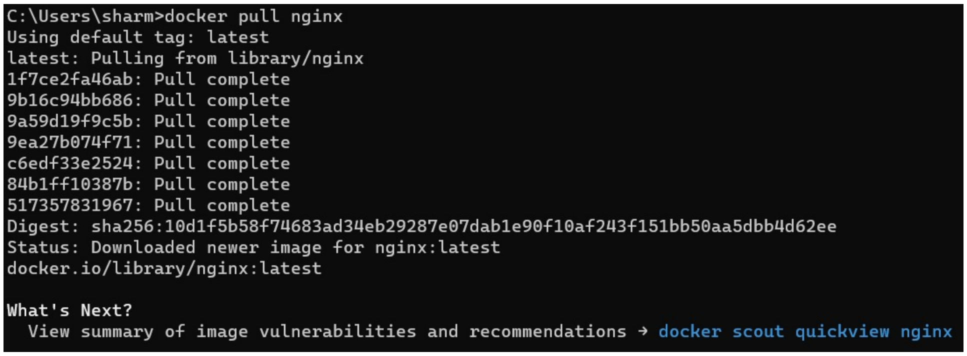
Steps to be followed:

1. Pulling a Docker image
2. Creating a new container
3. Stopping the container
4. Listing all the containers
5. Deleting the container
6. Removing the image

Step 1: Pulling a Docker image

1.1 Open the terminal and pull an image using the command:

sudo docker pull nginx

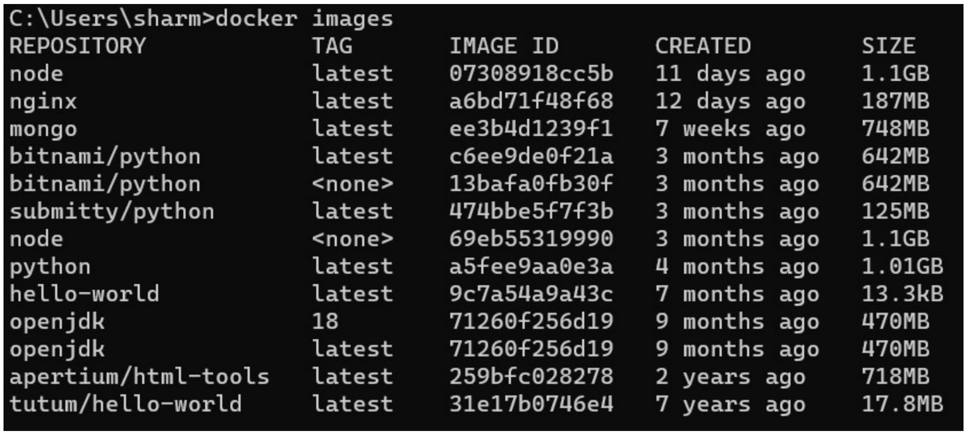


```
C:\Users\sharm>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
1f7ce2fa46ab: Pull complete
9b16c94bb686: Pull complete
9a59d19f9c5b: Pull complete
9ea27b074f71: Pull complete
c6edf33e2524: Pull complete
84b1ff10387b: Pull complete
517357831967: Pull complete
Digest: sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview nginx
```

1.2 List all the docker images to check the newly pulled *nginx* image:

sudo docker images

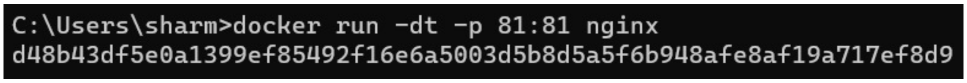


```
C:\Users\sharm>docker images
REPOSITORY             TAG       IMAGE ID       CREATED         SIZE
node                   latest    07308918cc5b   11 days ago     1.1GB
nginx                  latest    a6bd71f48f68   12 days ago     187MB
mongo                  latest    ee3b4d1239f1   7 weeks ago     748MB
bitnami/python         latest    c6ee9de0f21a   3 months ago    642MB
bitnami/python         <none>    13bafa0fb30f   3 months ago    642MB
submitty/python        latest    474bbe5f7f3b   3 months ago    125MB
node                   <none>    69eb55319990   3 months ago    1.1GB
python                 latest    a5fee9aa0e3a   4 months ago    1.01GB
hello-world            latest    9c7a54a9a43c   7 months ago    13.3kB
openjdk                18        71260f256d19   9 months ago    470MB
openjdk                latest    71260f256d19   9 months ago    470MB
apertium/html-tools    latest    259bfc028278   2 years ago     718MB
tutum/hello-world      latest    31e17b0746e4   7 years ago     17.8MB
```

Step 2: Creating a new container

2.1 Create a new container from the *nginx* image:

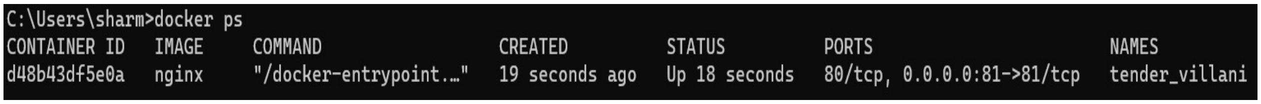
sudo docker run -dt -p 81:81 nginx



```
C:\Users\sharm>docker run -dt -p 81:81 nginx
d48b43df5e0a1399ef85492f16e6a5003d5b8d5a5f6b948afe8af19a717ef8d9
```

2.2 List all the running containers to check the newly created container. You can find various details like port of container, it's time of creation and ID.

sudo docker ps



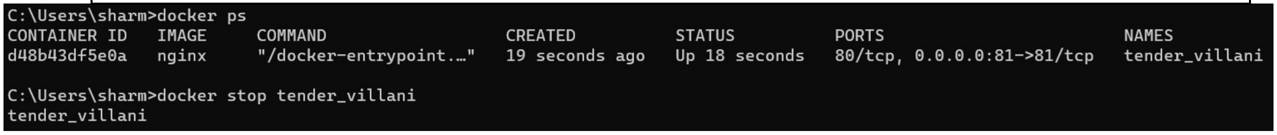
```
C:\Users\sharm>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                           NAMES
d48b43df5e0a   nginx     "/docker-entrypoint.…"   19 seconds ago   Up 18 seconds   80/tcp, 0.0.0.0:81->81/tcp      tender_villani
```

Step 3: Stopping the container

3.1 Use the following command to stop the running container. ***(****You can also us the container ID to stop the container: sudo docker stop CONTAINER_ID****)***

sudo docker stop CONTAINER_NAME

Note: Replace CONTAINER_NAME with the name of the newly created container. In this case CONTAINER_NAME is stoic_darwin. The container name may differ from the one shown in the image below.

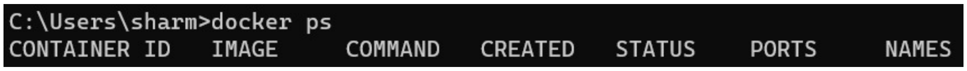


```
C:\Users\sharm>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                          NAMES
d48b43df5e0a   nginx     "/docker-entrypoint.…"   19 seconds ago   Up 18 seconds   80/tcp, 0.0.0.0:81->81/tcp     tender_villani

C:\Users\sharm>docker stop tender_villani
tender_villani
```

3.2 Use the following command to list all the running containers and verify if the container has stopped running:

sudo docker ps



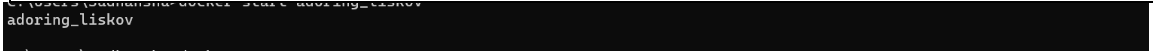
```
C:\Users\sharm>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

3.3 You can start the container again and check the running containers. ***(***You can also us the container ID to start the container: *sudo docker start CONTAINER_ID****)***

sudo docker start CONTAINER_NAME

sudo docker ps

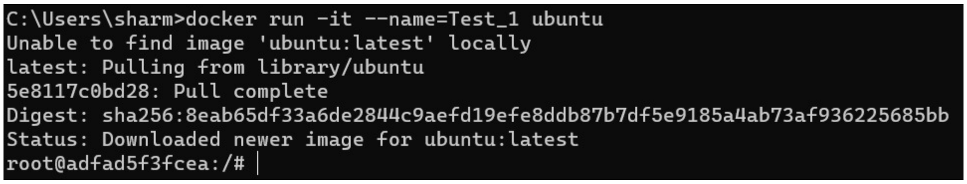
Note: Replace CONTAINER_NAME with the name of the newly created container. In this case CONTAINER_NAME is stoic_darwin. The container name may differ from the one shown in the image below.



```
C:\Users\Sudhanshu>docker start adoring_liskov
adoring_liskov
```

3.4 To start the container in interactive mode, use the –i and –t options.

sudo docker run -it --name=Test_1 ubuntu

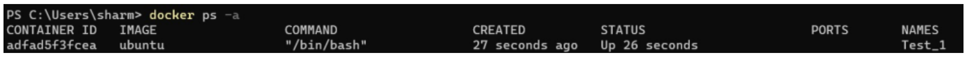


```
C:\Users\sharm>docker run -it --name=Test_1 ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
5e8117c0bd28: Pull complete
Digest: sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936225685bb
Status: Downloaded newer image for ubuntu:latest
root@adfad5f3fcea:/# |
```

Step 4: Listing all the containers

4.1 Use the below command to list all the containers started and the once which are stopped:

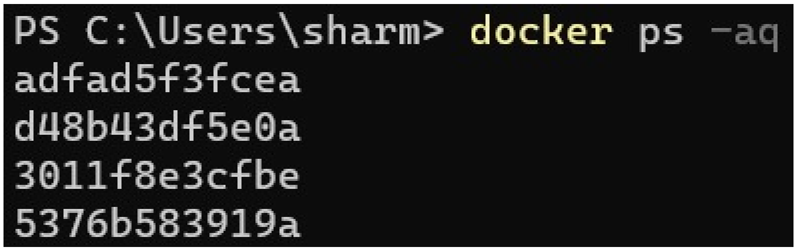
sudo docker ps -a



```
PS C:\Users\sharm> docker ps -a
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS                     PORTS      NAMES
adfad5f3fcea   ubuntu                 "/bin/bash"              27 seconds ago   Up 26 seconds                         Test_1
```

4.2 To list the containers by their ID, use the below command

sudo docker ps -aq

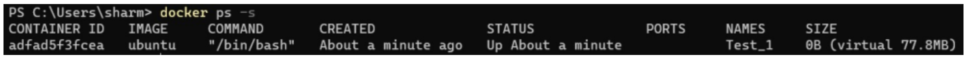


```
PS C:\Users\sharm> docker ps -aq
adfad5f3fcea
d48b43df5e0a
3011f8e3cfbe
5376b583919a
```

You can see the containers with ID are listed.

4.3 To list the total file size of each container, use the below command:

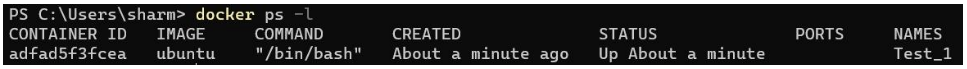
sudo docker ps -s



```
PS C:\Users\sharm> docker ps -s
CONTAINER ID   IMAGE     COMMAND       CREATED              STATUS              PORTS     NAMES     SIZE
adfad5f3fcea   ubuntu    "/bin/bash"   About a minute ago   Up About a minute             Test_1    0B (virtual 77.8MB)
```

4.4 To list the latest created containers, use the following command:

sudo docker ps -l



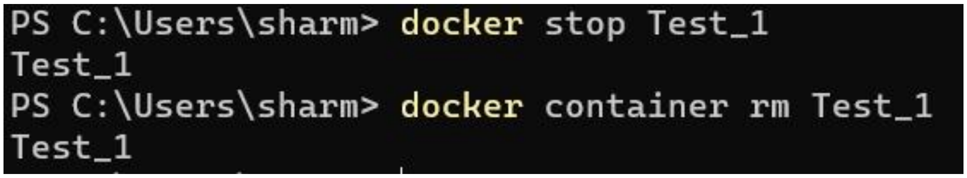
```
PS C:\Users\sharm> docker ps -l
CONTAINER ID   IMAGE     COMMAND       CREATED              STATUS              PORTS     NAMES
adfad5f3fcea   ubuntu    "/bin/bash"   About a minute ago   Up About a minute             Test_1
```

Step 5: Deleting the container

5.1 Stop the running container and remove it using the following commands:

sudo docker stop CONTAINER_NAME

sudo docker container rm CONTAINER_NAME



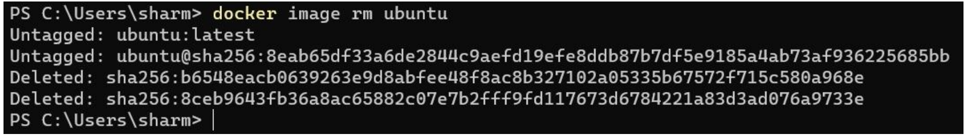
```
PS C:\Users\sharm> docker stop Test_1
Test_1
PS C:\Users\sharm> docker container rm Test_1
Test_1
```

Step 6: Removing the image

6.1 Remove the image using the command:

Note: Replace CONTAINER_NAME with the name of the newly created container. In this case CONTAINER_NAME is sweet_brown. The container name may differ from the one shown in the image below.

sudo docker image rm nginx



```
PS C:\Users\sharm> docker image rm ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936225685bb
Deleted: sha256:b6548eacb0639263e9d8abfee48f8ac8b327102a05335b67572f715c580a968e
Deleted: sha256:8ceb9643fb36a8ac65882c07e7b2fff9fd117673d6784221a83d3ad076a9733e
PS C:\Users\sharm> |
```