Lab Exercise 1

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

```
(base) vanshika@VANSHIKAs-MacBook-Air ~ % docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
```

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

   sudo docker images

```
[(base) vanshika@VANSHIKAs-MacBook-Air ~ % docker images
REPOSITORY                                          TAG
                        IMAGE ID       CREATED       SIZE
diwanvanshika/aco_5                                 latest
                        04ced233c161   2 hours ago   60.9MB
aco_5                                               latest
                        04ced233c161   2 hours ago   60.9MB
myimage                                             0.1
                        04ced233c161   2 hours ago   60.9MB
alpine                                              latest
                        98487ac3d995   2 days ago    7.66MB
nginx                                               latest
                        5628e5ea3c17   12 days ago   192MB
httpd                                               latest
                        1d37c7a1c40f   12 days ago   195MB
redis                                               latest
                        7f7ce731b26f   3 weeks ago   158MB
nginx                                               <none>
                        12ef77b9fab6   7 weeks ago   192MB
nginx                                               <none>
                        91582cfffc2d   2 months ago  192MB
flask-app                                           latest
```

Step 2: Creating a new container

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

   sudo docker run -dt -p 81:81 nginx

```
                              0b49127b0200     7 years ago    004MB
[(base) vanshika@VANSHIKAs-MacBook-Air ~ % docker run -dt -p 81:81 nginx

 5c3a1cc1ab7ec50f52293fde08f7f0239e84ea979f9a264d1915ee8ba5749ea9
```

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

```
[(base) vanshika@VANSHIKAs-MacBook-Air ~ % docker ps
CONTAINER ID   IMAGE   COMMAND                  CREATED         STATUS         PORTS
   NAMES
5c3a1cc1ab7e   nginx   "/docker-entrypoint.…"   37 seconds ago  Up 36 seconds  80/tcp, 0.0.0.0:81->81/tcp
   goofy_nightingale
bf8fab8151d8   redis   "docker-entrypoint.s…"   3 hours ago     Up 3 hours     6379/tcp
   redis
2ebebf97ebd7   httpd   "httpd-foreground"       2 days ago      Up 2 days
   k8s_httpd_apache-deployment-788fbf66b5-cxccd_default_ae619b1f-64c4-44de-9072-eb96b05861f3_0
345a21f7225c   httpd   "httpd-foreground"       2 days ago      Up 2 days
   k8s_httpd_apache-deployment-788fbf66b5-lfcmn_default_08927e4a-2ca3-409c-8467-0d79f8939183_0
```

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_NAM*E

```
[(base) vanshika@VANSHIKAs-MacBook-Air ~ % docker stop  goofy_nightingale  goofy_nightingale
goofy_nightingale
```

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

> *sudo docker ps*

```
[(base) vanshika@VANSHIKAs-MacBook-Air ~ % docker ps
CONTAINER ID   IMAGE   COMMAND                  CREATED         STATUS         PORTS
   NAMES
5c3a1cc1ab7e   nginx   "/docker-entrypoint.…"   37 seconds ago  Up 36 seconds  80/tcp, 0.0.0.0:81->81/tcp
   goofy_nightingale
bf8fab8151d8   redis   "docker-entrypoint.s…"   3 hours ago     Up 3 hours     6379/tcp
   redis
2ebebf97ebd7   httpd   "httpd-foreground"       2 days ago      Up 2 days
```

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 p*s

```
(base) vanshika@VANSHIKAs-MacBook-Air ~ % sudo docker start goofy_nightingale

goofy_nightingale
```

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

   sudo docker run -it --name=Test_1 ubuntu

```
docker: Error response from daemon: Conflict. The container name "/Test_1" is already i
3d59d18bb142d61debbe1f92539578eb50012f271524df42f11b46527d3f6". You have to remove (or
to be able to reuse that name.
See 'docker run --help'.
(base) vanshika@VANSHIKAs-MacBook-Air ~ %  docker run -it --name=Test_2 ubuntu

root@06a57ddae0ac:/#
```