

Experiment-3

Student Name: Abhishek Sainani

UID: 23MCC20080

Branch: MCA (CC & DevOps)

Section/Group: MCD 1-A

Semester: 3rd

Date of Performance: 13-08-2024

**Subject Name: Containerization
With Docker**

Subject Code: 23CAH-732

1) **Aim of the practical:** To deploy Docker images as stateless containers.

2) **Task to be done:**

- Deploy a Docker Image as stateless container and verify if it is running.

3) **Steps for the practical:**

- **Pull an image:** docker pull nginx

```
C:\Users\Abhishek Sainani>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
e4fff0779e6d: Pull complete
2a0cb278fd9f: Pull complete
7045d6c32ae2: Pull complete
03de31afb035: Pull complete
0f17be8dcff2: Pull complete
14b7e5e8f394: Pull complete
23fa5a7b99a6: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview nginx
```

- **Command To list all images:** docker images

```
C:\Users\Abhishek Sainani>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    5ef79149e0ec   7 days ago    188MB
```

- **Command to create and start a new container from pulled nginx image:**
docker run -d -p 80:80 nginx

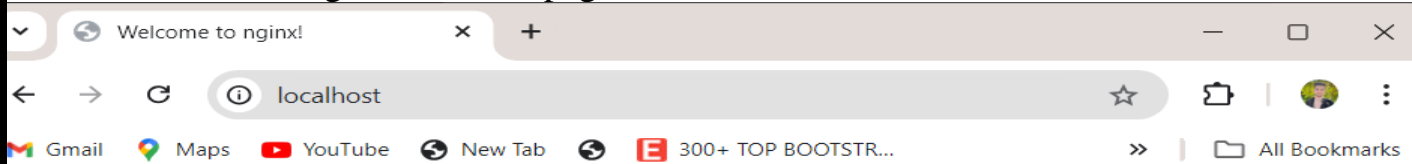
```
C:\Users\Abhishek Sainani>docker run -d -p 80:80 nginx
043117b764dbc368ace1025f65e467e55157f0a4e2362ff3941c2e7221e1beb9
```

- **Command to List All Containers:** `docker ps -a`
- **Command to List All currently Running Containers:** `docker ps`

```
C:\Users\Abhishek Sainani>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
043117b764db   nginx     "/docker-entrypoint..." 37 seconds ago Up 36 seconds  0.0.0.0:80->80/tcp       nifty_mayer

C:\Users\Abhishek Sainani>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
043117b764db   nginx     "/docker-entrypoint..." About a minute ago Up About a minute  0.0.0.0:80->80/tcp       nifty_mayer
```

- Now to verify deployment run `http://localhost` in web browser. We will see the default Nginx welcome page.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- **Commands to stop running container:** `docker stop containerName`
- **And command to verify that container is stopped or not:** `docker ps`

```
C:\Users\Abhishek Sainani>docker stop nginx
Error response from daemon: No such container: nginx

C:\Users\Abhishek Sainani>docker stop nifty_mayer
nifty_mayer

C:\Users\Abhishek Sainani>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
1f07c91c1491   nginx:latest "/docker-entrypoint..." 8 minutes ago Up 8 minutes  80/tcp                  exciting_heisenberg
```

- **Command to remove the container:** `docker rm containerName`
- **And command to verify that container is removed or not:** `docker ps -a`

```
C:\Users\Abhishek Sainani>docker rm nifty_mayer
nifty_mayer

C:\Users\Abhishek Sainani>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
1f07c91c1491   nginx:latest  "/docker-entrypoint..."  9 minutes ago  Up 9 minutes  80/tcp       exciting_heisenberg

C:\Users\Abhishek Sainani>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
1f07c91c1491   nginx:latest  "/docker-entrypoint..."  9 minutes ago  Up 9 minutes  80/tcp       exciting_heisenberg
```

- **Command to rename the container:** docker rename containerId newname
- **And command to verify that container is renamed or not:** docker ps -a

```
C:\Users\Abhishek Sainani>docker rename 1f07c91c1491 Elastic

C:\Users\Abhishek Sainani>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
1f07c91c1491   nginx:latest  "/docker-entrypoint..."  12 minutes ago  Up 12 minutes  80/tcp       Elastic
```

- **Finally run command to remove the image:** docker rmi imageId

```
C:\Users\Abhishek Sainani>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    5ef79149e0ec   7 days ago     188MB

C:\Users\Abhishek Sainani>docker rmi 5ef79149e0ec
Error response from daemon: conflict: unable to delete 5ef79149e0ec (must be forced) - image is being used by stopped container 1f07c91c1491

C:\Users\Abhishek Sainani>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
1f07c91c1491   nginx:latest  "/docker-entrypoint..."  19 minutes ago  Exited (0) 2 minutes ago  Elastic

C:\Users\Abhishek Sainani>docker rm 1f07c91c1491
1f07c91c1491

C:\Users\Abhishek Sainani>docker rmi 5ef79149e0ec
Untagged: nginx:latest
Deleted: sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add
Deleted: sha256:5ef79149e0ec84a7a9f9284c3f91aa3c20608f8391f5445eabe92ef07dbda03c
Deleted: sha256:aa557aaf0b93b5b9af247beb198be89632444af2d52b37f2b67bdf2227194625
Deleted: sha256:85dee7d6fa4b70eaa6b43b5d5e5255b6794fac35d2aa6a80f8590dad223c1a6
Deleted: sha256:9996f8429bfc2f1e10f561e17bdc0c66eb82e30088725fbd634a774933ba066
Deleted: sha256:d861b1c34411369feaaaf4921018baa607e89f1c393b8ed950f676dd3349b9ff
Deleted: sha256:01b9716819d1d6863f886e1adcd8bb2bf936dcac616e1750e873bfd8e1667014
Deleted: sha256:8d8c7099ecd8343e6700cd1d348211ab159c0eda2a9d318c5cf600a8eb2588f5e
Deleted: sha256:9853575bc4f955c5892dd64187538a6cd02dba6968eba9201854876a7a257034

C:\Users\Abhishek Sainani>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
```

4) learning Outcomes:

1. Understood what Docker images and containers are and how they work.
2. Understanding of what stateless containers are and why they are useful.
3. Built the ability to pull, run, and clean up Docker images, optimizing system resources.
4. Develop skills to manage Docker containers from creation to removal, ensuring smooth application deployment and maintenance.

