UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Experiment No. 1.4

Student Name: Ambika Sharma **UID: 22MCC20022**
Branch: MCA CC & Devops **Section/Group – 22MCD - 1/A**
Semester: III **Date of Performance: 27/09/23**
Subject Name: Containerization with Docker **Subject Code: - 22CAH-742**

1. Aims/Overview for the practical:

Managing containers with docker CLI

2. Steps for experiment/practical:

Managing containers with the Docker Command-Line Interface (CLI) involves various tasks like running containers, monitoring their status, stopping and removing containers, and inspecting container properties. Here are the steps for managing containers using the Docker CLI:

Step 1. Install Docker: If you haven't already, install Docker on your host machine. You can download and install it from the official Docker website or use a package manager if you're on a Linux distribution.

Step 2. **Pull Docker Images:** If you plan to run containers based on existing Docker images, use the `docker pull` command to download the desired image from a container registry, such as Docker Hub.
For example:

```bash
docker pull nginx
```

```
PS D:\docker> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
a378f10b3218: Pull complete
4dfff0708538: Pull complete
2135e49ace4b: Pull complete
c843f6b280ce: Pull complete
6f35ab6f1400: Pull complete
6c538b49fa4a: Pull complete
d57731fb9008: Pull complete
Digest: sha256:b4af4f8b6470febf45dc10f564551af682a802eda1743055a7dfc8332dffa595
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

Step 3. Run a Container: Start a container from an image using the `docker run` command. For example, to run an Nginx web server container:

```bash
docker run -imagename
```

- `-d`: Run the container in the background.
- `-p 80:80`: Map port 80 on the host to port 80 in the container.
- `--name my-nginx-container`: Assign a custom name to the container.

```
PS D:\docker> docker run nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/10/17 17:30:44 [notice] 1#1: using the "epoll" event method
2023/10/17 17:30:44 [notice] 1#1: nginx/1.25.2
```

Step 4. List Running Containers: To see the list of running containers, use the `docker ps` command:

```bash
docker ps
```

To view all containers (including stopped ones), add the `-a` option:

```bash
docker ps -a
```

```
PS D:\docker> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS D:\docker> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS                       PORTS     NAMES
aa8bc0cc006e   nginx     "/docker-entrypoint.…"   About a minute ago  Exited (0) 26 seconds ago              jovial_davinci
e3faaca102c9   ambika    "python ./helloworld…"   9 hours ago         Exited (0) 9 hours ago                 pensive_noyce
c1194021817d   ambika    "python ./helloworld…"   2 weeks ago         Exited (0) 2 weeks ago                 dazzling_nash
```

Step 5. Inspect Container Details: To get detailed information about a specific container, including its configuration and network settings, use the `docker inspect` command:

```bash
**docker inspect my-nginx-container**
```

```
PS D:\docker> docker inspect nginx
[
    {
        "Id": "sha256:bc649bab30d150c10a84031a7f54c99a8c31069c7bc324a7899d7125d59cc973",
        "RepoTags": [
            "nginx:latest"
        ],
        "RepoDigests": [
            "nginx@sha256:b4af4f8b6470febf45dc10f564551af682a802eda1743055a7dfc8332dffa595"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2023-10-12T03:14:44.515974298Z",
        "Container": "8dfe4db4918db2931d8291b5d55c6ac6099d142e58d5688cee7cddb27e404724",
        "ContainerConfig": {
```

Step 6. View Container Logs: To view the logs of a running container, use the `docker logs` command:

docker logs my-nginx-container

```
PS D:\docker> docker logs 454a1df9227d
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/10/17 18:45:46 [notice] 1#1: using the "epoll" event method
2023/10/17 18:45:46 [notice] 1#1: nginx/1.25.2
2023/10/17 18:45:46 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/10/17 18:45:46 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2023/10/17 18:45:46 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
```

Step 7. Stop a Container: To stop a running container, use the `docker stop` command:

docker stop my-nginx-container

```
PS D:\docker> docker stop aa8bc0cc006e
aa8bc0cc006e
PS D:\docker> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
aa8bc0cc006e   nginx     "/docker-entrypoint.…"   55 minutes ago   Exited (0) 53 minutes ago
e3faaca102c9   ambika    "python ./helloworld…"   10 hours ago     Exited (0) 10 hours ago
c1194021817d   ambika    "python ./helloworld…"   2 weeks ago      Exited (0) 2 weeks ago
```

Step 8. Start a Stopped Container: To start a stopped container, use the `docker start` command:

docker start my-nginx-container

```
PS D:\docker> docker start aa8bc0cc006e
aa8bc0cc006e
PS D:\docker> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
aa8bc0cc006e   nginx     "/docker-entrypoint.…"   56 minutes ago   Up 1 second
e3faaca102c9   ambika    "python ./helloworld…"   10 hours ago     Exited (0) 10 hours ago
c1194021817d   ambika    "python ./helloworld…"   2 weeks ago      Exited (0) 2 weeks ago
PS D:\docker>
```

Step 9. Remove a Container: If you want to remove a container, use the `docker rm` command:

```bash
docker rm my-nginx-container
```

```
PS D:\docker> docker rm aa8bc0cc006e
aa8bc0cc006e
PS D:\docker> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
e3faaca102c9   ambika    "python ./helloworld…"   10 hours ago     Exited (0) 10 hours ago
c1194021817d   ambika    "python ./helloworld…"   2 weeks ago      Exited (0) 2 weeks ago
PS D:\docker>
```

Note that you cannot remove a running container. You must stop it first using `docker stop`.

Step 10. Manage Container Resources: To modify container resources, such as CPU and memory limits, you can use the `docker update` command. For example, to change the CPU share of a running container:

```bash
docker update --cpus 2 my-nginx-container
```

- `--cpus 2`: Limit the container to 2 CPU shares.

```
PS D:\docker> docker update --cpus 2 ef08076d595e
ef08076d595e
PS D:\docker> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
acf314dd28f1   nginx     "/docker-entrypoint.…"   2 minutes ago    Exited (0) About a minute ago
d1cf77439961   nginx     "/docker-entrypoint.…"   2 minutes ago    Exited (0) 2 minutes ago
ef08076d595e   nginx     "/docker-entrypoint.…"   4 minutes ago    Exited (0) 4 minutes ago
e3faaca102c9   ambika    "python ./helloworld…"   10 hours ago     Exited (0) 10 hours ago
c1194021817d   ambika    "python ./helloworld…"   2 weeks ago      Exited (0) 2 weeks ago
```

Step 11. Cleanup: Regularly clean up stopped containers, unused images, and other Docker resources to save disk space and keep your environment tidy. You can use the `docker system prune` command for this purpose.

docker system prune

```
PS D:\docker> docker system prune
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all dangling images
  - all dangling build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
acf314dd28f1a8a2ac21ad31601fe793932fe24ad2c674deb855427bce2df6a6
d1cf774399611a9737d164b11b3a0e219d91060b4e783af59c13ad032ad8e6f7
ef08076d595e56341e580bcfd73766a63de9c178b70ab141fb3a971222a3d2d2
e3faaca102c9a9df80f62721e75955dac0173e6a21e0f45456b741e5079fb3e1
c1194021817df4ae98f2157e812263a7063da79957f954b22a7333319aa70159
```

4. Result/Output/Writing Summary:

```
Are you sure you want to continue? [y/N] y
Deleted Containers:
acf314dd28f1a8a2ac21ad31601fe793932fe24ad2c674deb855427bce2df6a6
d1cf774399611a9737d164b11b3a0e219d91060b4e783af59c13ad032ad8e6f7
ef08076d595e56341e580bcfd73766a63de9c178b70ab141fb3a971222a3d2d2
e3faaca102c9a9df80f62721e75955dac0173e6a21e0f45456b741e5079fb3e1
c1194021817df4ae98f2157e812263a7063da79957f954b22a7333319aa70159

Deleted build cache objects:
snms6p6vy7vele0py644tfi3c
x8qq1des0ydt1ia9qtmtz0s75
uwex3hqrzoja2ewrxwrhvgi0n
j9h2fr8lgvgsmjmwk6cue0ckk
ns3lo2fp2nujpjr9yde390ku5

Total reclaimed space: 3.433kB
PS D:\docker> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS D:\docker>
```

5. Learning outcomes (What I have learned):

a) How to build image using docker CLI
b) How to start, stop and remove container on docker CLI