Experiment -1.2

Student Name: Aditi Pandey **UID:22BDO10031**
Branch: CSE(DevOps) **Section/Group: 22BCD-1/A**
Semester: 5Sem **Date of Performance:5/8/24**
Subject Name: Docker and Kubernates **Subject Code: 22CSH-343**

1. Aim/Overview of the practical: To understand the Container Lifecycle Management with Docker:

- Docker images
- Containers
- Docker repository
- Docker commands
- Docker File

2. Apparatus: Windows 11 PC, VS Code, Chrome Browser

3. Terms used in experiment/practical:

Docker is an open-source platform **based on Linux containers** for developing and running applications inside containers.
Docker is used to deploy many containers simultaneously on a given host.
Containers are very fast and lightweight because they don't need the extra load of a hypervisor as they run directly within the host machine's kernel.

The main concepts involved in Container Lifecycle Management with Docker:

1. Docker Images:

- Docker images are read-only templates used to create containers.
- They include everything needed to run an application—code, runtime, libraries, environment variables, configuration files, and dependencies.

2. Containers:

- Containers are lightweight, standalone, executable packages that contain everything needed to run an application.
- They are created from Docker images and run in isolated environments.

3. Docker Repository:

- A Docker repository (like Docker Hub) is a storage location for Docker images.
- It acts as a version-controlled registry where images are stored and can be pushed or pulled from.

4. Docker File:

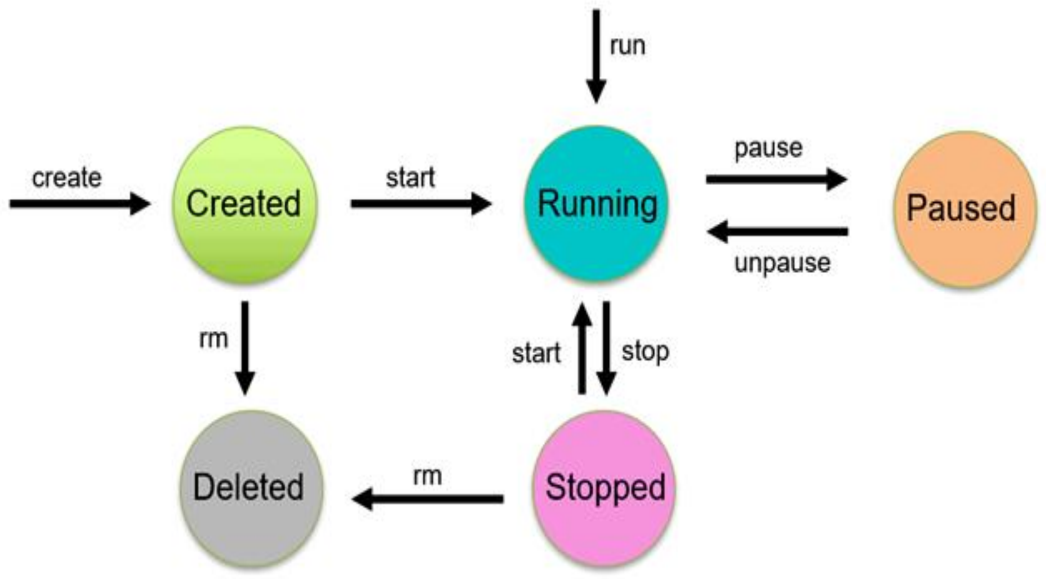
- A Dockerfile is a text document that contains a series of instructions on how to build a Docker image.
- Each instruction in the file builds a layer in the image, adding things like base images, dependencies, and configuration.

4. Docker Commands:

Here are some commonly used Docker commands that manage the lifecycle of images, containers, and repositories:

Container Commands:

- docker run: Creates and starts a container from an image.
- docker ps: Lists running containers.
- docker stop: Stops a running container.
- docker rm: Removes a stopped container.

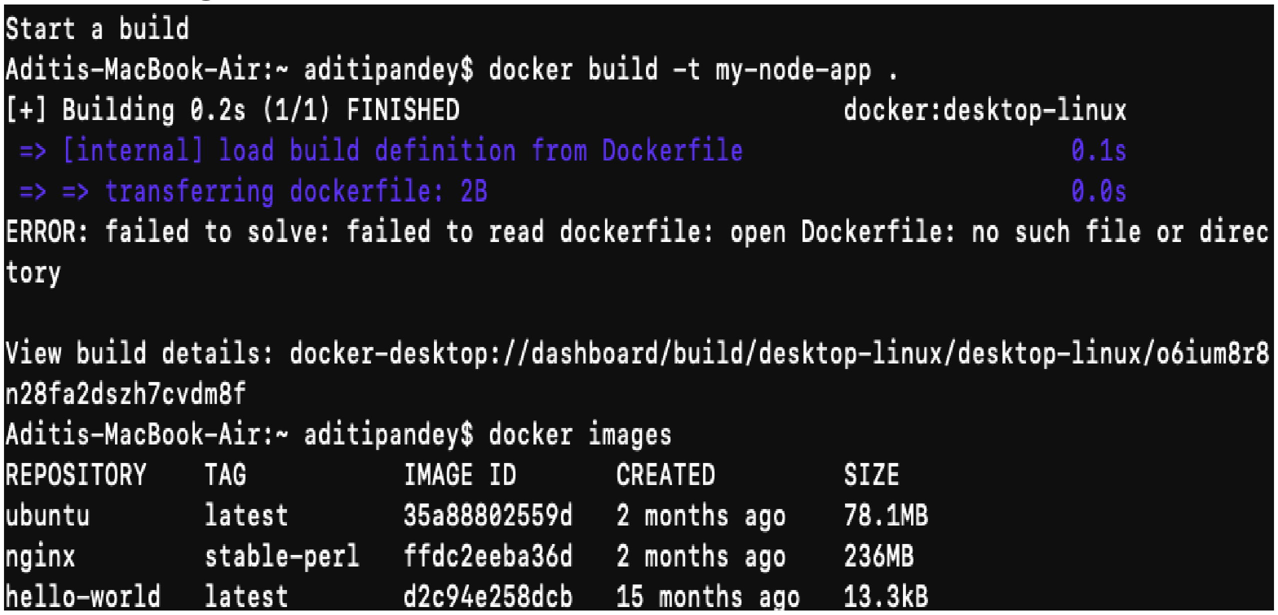


<u>Container Lifecycle Management</u> refers to the process of handling containers throughout their lifecycle, from creation to removal. Here's a simplified version of the stages:

1. **Container Creation:** Build a container from an image (e.g., with Docker), which packages the application and its dependencies.
2. **Container Configuration:** Set up environment variables, volumes, and networking before running the container.
3. **Running:** The container is started and executes the application. It's monitored for performance and issues.
4. **Scaling and Orchestration:** Use tools like Kubernetes to scale containers up or down and distribute traffic across them.
5. **Health and Resilience:** Health checks and automatic restarts keep containers running smoothly.
6. **Pausing and Stopping:** Containers can be paused or stopped temporarily without losing state.
7. **Termination and Removal:** Stop and remove containers when no longer needed to free up resources.

1. Container Creation:

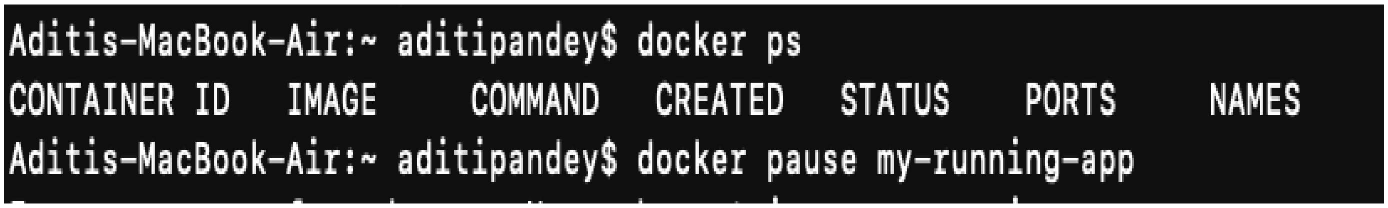
- **Build an Image from Dockerfile:**



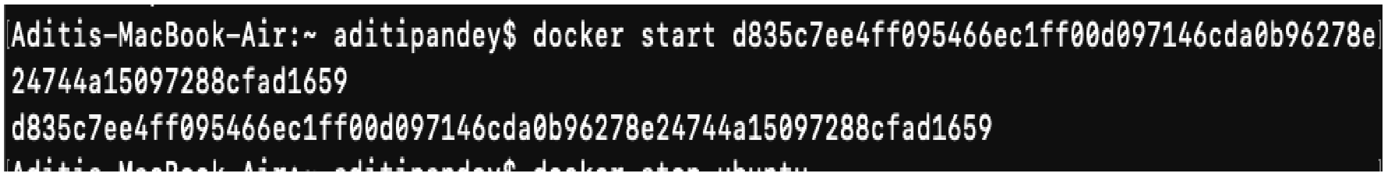
```
Start a build
Aditis-MacBook-Air:~ aditipandey$ docker build -t my-node-app .
[+] Building 0.2s (1/1) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                      0.1s
 => => transferring dockerfile: 2B                                        0.0s
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or direc
tory

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/o6ium8r8
n28fa2dszh7cvdm8f
Aditis-MacBook-Air:~ aditipandey$ docker images
REPOSITORY    TAG           IMAGE ID       CREATED         SIZE
ubuntu        latest        35a88802559d   2 months ago    78.1MB
nginx         stable-perl   ffdc2eeba36d   2 months ago    236MB
hello-world   latest        d2c94e258dcb   15 months ago   13.3kB
```

2. Running/Execution:

- **List Running Containers**



```
Aditis-MacBook-Air:~ aditipandey$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
Aditis-MacBook-Air:~ aditipandey$ docker pause my-running-app
```



```
Aditis-MacBook-Air:~ aditipandey$ docker start d835c7ee4ff095466ec1ff00d097146cda0b96278e
24744a15097288cfad1659
d835c7ee4ff095466ec1ff00d097146cda0b96278e24744a15097288cfad1659
```

3. Pausing and Stopping

Pause a Container by docker pause <container_name>

Learning outcomes (What I have learnt):

1. I have learnt how to structure a webpage using HTML

2. I have learnt how to grant logic in our webpage using JavaScript

3. I have learnt how to design our webpage using CSS

4. I have learnt how I can show an element based on some condition

5. I have learnt about the DOM model in JS

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			