

Experiment No. 3.1

Student Name: Parvinder Singh

UID: 22MCC20043

Branch: MCA - CCD

Section/Group: 22MCD-1/ Grp B

Semester: III

Subject Name: Containerization With Docker

Subject Code: 22CAH-742

1. **Aim/Overview of the practical:** Working with Docker Compose.

2. Code for experiment/practical:

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

Compose works in all environments; production, staging, development, testing, as well as CI workflows. It also has commands for managing the whole lifecycle of your application:

- Start, stop, and rebuild services
- View the status of running services
- Stream the log output of running services
- Run a one-off command on a service

Working with Docker Compose:

- Installation
- Create Docker Compose File
- Use Docker Compose file to create services

A. Installation

With Docker-Desktop installation, Docker-Compose is pre-installed

```
PS C:\Users\Pinda> docker-compose -v
Docker Compose version v2.19.1
```

B. Create Docker Compose file

- Create a directory at a specific location.

```
PS C:\Users\Pinda\Desktop\Docker> mkdir DockerComposeFile

Directory: C:\Users\Pinda\Desktop\Docker

Mode                LastWriteTime         Length Name
----                -
d-----          04-11-2023   15:58             DockerComposeFile
```

- Go to that directory and create a file docker-compose.yml.

```

v DOCKERCOMPOSEFILE
  docker-compose.yml
1  version: '3'
2
3  services:
4    web:
5      image: nginx
6
7      volumes:
8        - ./templates:/etc/nginx/templates
9
10     ports:
11       - "9090:80"
12
13     database:
14       image: mysql

```

- Check the validity of docker-compose file.

```

PS C:\Users\Pinda\Desktop\Docker\DockerComposeFile> docker-compose config
name: dockercomposefile
services:
  database:
    image: mysql
    networks:
      default: null
  web:
    image: nginx
    networks:
      default: null
    ports:
      - mode: ingress
        target: 80
        published: "9090"
        protocol: tcp
    volumes:
      - type: bind
        source: C:\Users\Pinda\Desktop\Docker\DockerComposeFile\templates
        target: /etc/nginx/templates
        bind:
          create_host_path: true
networks:
  default:
    name: dockercomposefile_default

```

C. Use Docker Compose File to create services

- Run docker-compose.yml file.

```

PS C:\Users\Pinda\Desktop\Docker\DockerComposeFile> docker-compose up -d
[+] Running 3/3
  ✓ Network dockercomposefile_default      Created                                0.3s
  ✓ Container dockercomposefile-web-1      Started                                3.3s
  ✓ Container dockercomposefile-database-1 Started                                3.3s

PS C:\Users\Pinda\Desktop\Docker\DockerComposeFile> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                               NAMES
588a71f85df7   nginx    "/docker-entrypoint...." About a minute ago Up About a minute 0.0.0.0:9090->80/tcp                dockercomposefile-web-1

```

- To stop docker-compose

```

PS C:\Users\Pinda\Desktop\Docker\DockerComposeFile> docker-compose down
[+] Running 3/3
  ✓ Container dockercomposefile-web-1      Removed                                0.6s
  ✓ Container dockercomposefile-database-1 Removed                                0.0s
  ✓ Network dockercomposefile_default      Removed                                0.2s

```

3. Learning outcomes (What I have learned):

- To understand the Docker Compose.
- To manage multiple containers using Docker Compose.
- To create services in Docker Compose.