Experiment-2.2

Student Name:Aditi Pandey **UID:22BDO10031**
Branch: CSE(DevOps) **Section/Group:22BCD1/A**
Semester:5 Sem **Date of**
Performance:23sept2024Subject Name:Docker and Kuberenats Subject Code: 22CSH-343

1.Aim/Overview of the practical:

To Setup i. Container to WWW Communication,
ii. Container to Local Host Machine Communication,
iii. Container to Container Communication,
iv. Creating a Container & Communicating to the Web (WWW),
v. Container to Host Communication Work,
vi. Container to Container Communication using Docker Desktop

2. Apparatus: PC, Docker Engine, DockerHub, Ubuntu Linux

3. Steps for experiment/practical:

Docker Networking:

1. It allows you to create a Network of Docker Containers managed by a master node called the manager.

2. Containers inside the Docker Network can talk to each other by sharing packets of information.

3. The Docker network is a virtual network created by Docker to enable communication between Docker containers.

4. If two containers are running on the same network or host, they can communicate with each other without the need for ports to be exposed to the host machine.

Docker Network Commands:-

1.Create a Network:
sudo docker network create --driver <network_driver> <network_name>

2 . List all available Docker networks

sudo docker network ls

```
Aditis-MacBook-Air:~ aditipandey$ docker network create --driver bridge demo
6583d76747eb541ace19c7c8efc0b70e68d85c1ac0b17f00a27ccb93ce4d79dd
Aditis-MacBook-Air:~ aditipandey$ docker network ls
NETWORK ID     NAME           DRIVER    SCOPE
2b5629bb27a8   bridge         bridge    local
6583d76747eb   demo           bridge    local
11cadca67783   demo-network   bridge    local
68300b774225   host           host      local
76247a2fa1fa   none           null      local
```

3. Connect a Container to a Network:

docker network connect <network_name> <container_name_or_id>

```
Aditis-MacBook-Air:~ aditipandey$ docker network connect demo 2c773f456ed82eb53031838ed0155438ce2657e8c267974e68d7fd3312c2349f
Aditis-MacBook-Air:~ aditipandey$ sudo docker network rm demo
Password:
demo
Aditis-MacBook-Air:~ aditipandey$ docker network ls
NETWORK ID     NAME           DRIVER    SCOPE
2b5629bb27a8   bridge         bridge    local
11cadca67783   demo-network   bridge    local
68300b774225   host           host      local
76247a2fa1fa   none           null      local
```

4.Inspect a Network: Get detailed information about a specific network:

docker network inspect <network_name_or_id>

```
Aditis-MacBook-Air:~ aditipandey$ docker network inspect demo
[
    {
        "Name": "demo",
        "Id": "261d78792d5178ce11f5087de2293bb1389677398cb625874036c1508dae0f66",
        "Created": "2024-10-15T13:57:02.003860921Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
```

5.Disconnect a container from a network:
docker network disconnect <network_name> <container_name_or_id>
6. Remove a custom network (it must not be in use):
docker network rm <network_name_or_id>

```
Aditis-MacBook-Air:~ aditipandey$ docker network disconnect demo 2c773f456ed82eb53031838ed0155438ce2657e8c267974e68d7fd3312c2349f
Aditis-MacBook-Air:~ aditipandey$ sudo docker network rm demo
Password:
Sorry, try again.
Password:
demo
Aditis-MacBook-Air:~ aditipandey$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
185cda24afaf   bridge    bridge    local
68300b774225   host      host      local
76247a2fa1fa   none      null      local
```

7. Remove Unused Networks: Clean up networks that are not associated with any containers:
docker network prune

```
Aditis-MacBook-Air:~ aditipandey$ docker network prune
WARNING! This will remove all custom networks not used by at least one container.
Are you sure you want to continue? [y/N] y
Aditis-MacBook-Air:~ aditipandey$
```

● Container to WWW Communication:

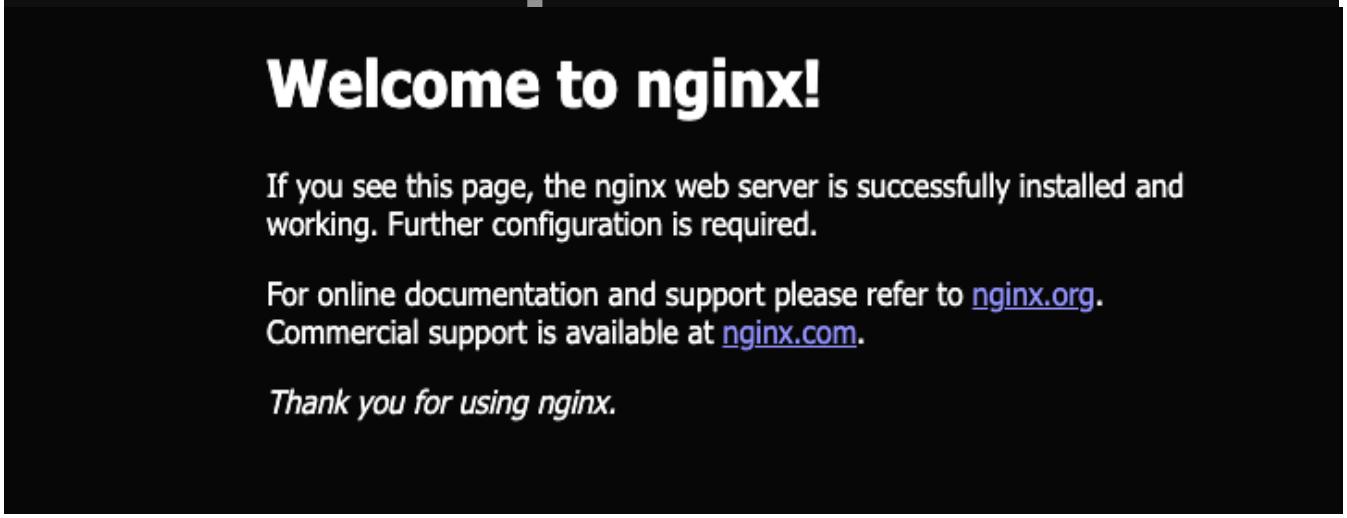
1. Pull an Nginx image from Docker Hub to create a web server container.
2. Create a Docker container using the Nginx image
3. Access the default webpage running inside the container using https://localhost:80

```
Aditis-MacBook-Air:~ aditipandey$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
302e3ee49805: Already exists
d07412f52e9d: Pull complete
9ab66c386e9c: Pull complete
4b563e5e980a: Pull complete
55af3c8febf2: Pull complete
5b8e768fb22d: Pull complete
85177e2c6f39: Pull complete
Digest: sha256:d2eb56950b84efe34f966a2b92efb1a1a2ea53e7e93b94cdf45a27cf3cd47fc0
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
```

1. Create a Docker container using the Nginx image.

```
Aditis-MacBook-Air:~ aditipandey$ docker run -dit --name default-web  -p 80:80 nginx
d0772a727e5e0fce082c0f0a4346630ccb4d9840136d26a2c03a3e2a11e79219
```



Container to Host Communication Work:

1. Identify the service running on the host machine that you want to communicate with from the container. For this, a web-server running on nginx on the host machine.
2. Create a Docker container and configure it to communicate with the host service.

```
root@1bceb290d1c3:/# curl http://localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Learning outcomes (What I have learnt):

1. I have learnt the concept of containerization.
2. I have learnt to configure Docker to work with different environments.
3. I have learnt how to build docker images using Dockerfile.
4. I have learnt the purpose of Dockerfile and its advantages.
5. I have learnt how Dockerfile can help in creating CI/CD pipelines.

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.			