Practical-8
Study of Docker swarm and Deployment of ML project in swarm network

The objective of this lab is to understand Docker Swarm, a container orchestration tool, and deploy a machine learning project in a Swarm network using three Linux host systems.

Prerequisites:
- Three Linux host systems (e.g., Ubuntu)
- Docker installed on each host (follow the installation steps in the previous response)
- Basic understanding of Docker and machine learning concepts

Lab Setup:
- Assign unique hostnames and IP addresses to each Linux host.
- Ensure that the hosts can communicate with each other over the network.

Lab Steps:

Step 1: Initialize Docker Swarm

```
PS D:\Desktop\stream> docker swarm init
```
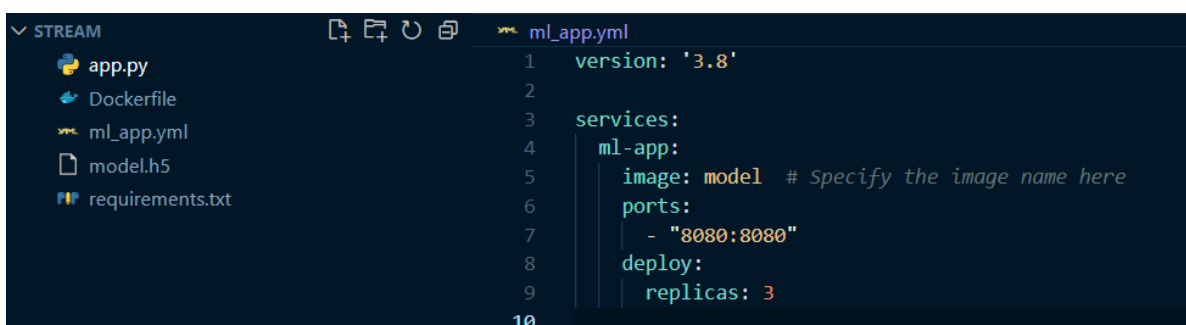
Step 2: Join Worker Nodes

```
PS D:\Desktop\stream> docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-01318qryysycsm974bfp3a5zs0jq39gy3grfzgweslo59wsxn2-6sgdmezjy22wefsr8sv413w9a 192.168.65.4:2377
```

```
PS D:\Desktop\stream> docker info --format '{{.Swarm.NodeAddr}}'
192.168.65.4
```

Step 3: Deploy a ML Model as a Service
Create a Docker Compose file (e.g., `ml_app.yml`) with the following content

```
✓ STREAM                              ᵐᵒ ml_app.yml
   🐍 app.py                      1     version: '3.8'
   🐳 Dockerfile                  2
   ᵐᵒ ml_app.yml                  3     services:
   📄 model.h5                    4       ml-app:
   🔧 requirements.txt            5         image: model  # Specify the image name here
                                  6         ports:
                                  7           - "8080:8080"
                                  8         deploy:
                                  9           replicas: 3
                                 10
```

Step 4: Deploy the Service On the manager node, deploy the ML project as a service

```
PS D:\Desktop\stream> docker stack deploy -c ml_app.yml ml_stack
Creating network ml_stack_default
Creating service ml_stack_ml-app
```

Step 5: Verify the Deployment Check the status of the deployed service:

```
PS D:\Desktop\stream> docker service ls
ID              NAME              MODE        REPLICAS   IMAGE          PORTS
i6ci69j0ud96    ml_stack_ml-app   replicated  0/3        model:latest   *:8080->8080/tcp
```

Step 6: Access the ML Project You can access the ML project on any of the worker nodes using their IP addresses and port 8080 (the port we exposed in the Compose file).

Using https://192.168.1.2:8080 we can access the model

Step 7: Scaling Experiment with scaling the service up or down to see how Docker Swarm manages the replicas.

```
PS D:\Desktop\stream> docker service scale ml_stack_ml-app=5
ml_stack_ml-app scaled to 5
overall progress: 0 out of 5 tasks
1/5: preparing [=================================>                ]
2/5: preparing [=================================>                ]
3/5: No such image: model:latest
4/5: preparing [=================================>                ]
5/5: preparing [=================================>                ]
```

Step 8: Removing the Stack When done, you can remove the stack

```
PS D:\Desktop\stream> docker stack rm ml_stack
Removing service ml_stack_ml-app
Removing network ml_stack_default
```

Step 9: Leave Swarm On worker nodes, leave the Swarm when you're finished

```
PS D:\Desktop\stream> docker swarm leave --force
Node left the swarm.
```

Step 10: Shut Down Shut down all host systems.

```
PS D:\Desktop\stream> sudo shutdown -h now
```