Intro

Ref https://docs.docker.com/engine/swarm/

Docker swarm is a container orchestration system, very similar to kubernetes.

It's not used as often anymore, k8s picked up most of the heat.

Core concepts

Services

Tasks

Containers

Kubernetes vs Docker swarm

Kubernetes	Docker swarm
Very hard to understand	Much easier to understand
Much more prod ready, adopted and used	Not used as often
Supports autoscaling	Have to scale it manually

Need to install/understand kubectl	Works with the docker cli

Architecture

Manager Node

Manager nodes handle cluster management tasks:

- Maintaining cluster state
- Scheduling services

Worker Node

Worker nodes are also instances of Docker Engine whose sole purpose is to execute containers.

Services, tasks, containers

To deploy an application image when Docker Engine is in Swarm mode, you create a service. Frequently a service is the image for a microservice within the context of some larger application (eg - HTTP Server)

- **Service** - A service is the definition of how you want to run your application in the swarm. It specifies the desired state, including the number of replicas, the image to use, the command to run, and other configurations such as networks and volumes.
- **Task -** A task is a single instance of a service running on a node. Each task represents one container and its associated metadata. When you create a service with multiple replicas, Docker Swarm creates a task for each replica.
- Container - A container is a running instance of a Docker image. Each task maps to one container. The swarm orchestrator ensures the tasks (and thus the containers) are distributed across the nodes in the swarm according to the defined service specifications.

Create a 2 node swarm

- Create two EC2 machines, install docker in both of them
- Initialise swarm in the first machine

```
docker swarm init
```

- Make the other server join the master (replace the token, ip from the first command)

```
docker swarm join --token b-1-45q02kic0tij84lhkb5du9esm38ly2g6kf3ssm2tq1
```

- Make sure the 2377 port on the machine is open
- Confirm the nodes status

```
docker node ls
```

Deploying a service

- Deploy the nginx service

```
docker service create --replicas 3 --name helloworld -p 3000:80 nginx
```

- Check the status of the service

```
docker service ls
```

- Go to the machine URL on port 3000 and ensure you see it running

```
your_machine_ip:3000
```

- Try deleting a few pods and see if they come back up
- Delete the service

```
docker service rm helloworld
```