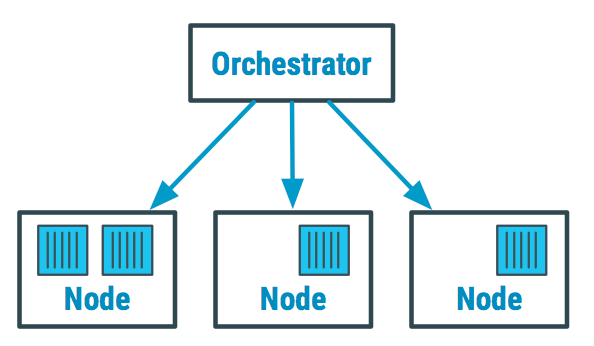
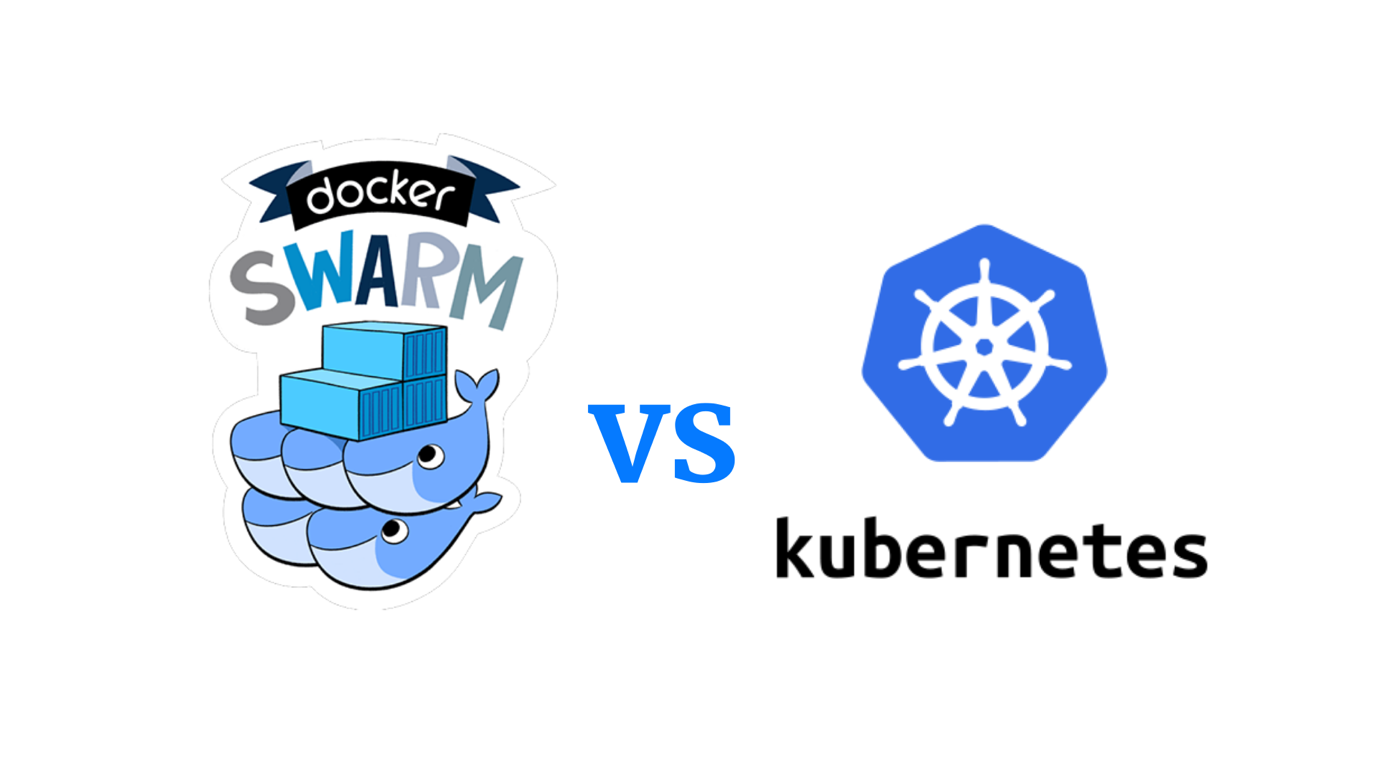
Orchestration



Diagram

Description automatically generated



Graphical user interface

Description automatically generated

Docker Swarm Setup

## Master

docker swarm init

**Important Note:** Copy Swarm join command

# Worker1

docker swarm join --token <token> <ip:port>

# Master

docker node ls

Note: If you forgot to copy the join command,

docker swarm join-token worker

# Worker2

docker swarm join --token <token> <ip:port>

Service

docker service create --name myweb --replicas 3 nginx

docker service ls

docker service create --name myservice --mode global nginx

docker service ps myservice

#Containers runs in all nodes, as mode is GLOBAL

docker service ps myweb

#Containers are created based on Replicas

Scaling Services

docker service scale myweb=5

docker service create --name scale-cont --replicas 3 nginx

docker service update --replicas=4 scale-cont

docker service update --replicas=1 scale-cont

docker service scale myweb=3 scale-cont=3

#Cannot scale service running in global mode

docker service scale myservice=5

Notice that the scaling cannot be used with global services. It can only be done with replicated service.

Labels

docker node ls -f role=manager

docker node ls

docker node update --label-add name=worker1 ip-172-31-0-35

docker node ls --filter node.label=name=worker1

Node Management

# Drain a Node

docker node ls

docker service create --replicas 3 --name redis --update-delay 10s redis:3.0.6

docker service ps redis

docker node update --availability drain <hostname\_Worker1>

docker node inspect --pretty <hostname\_Worker1 >

docker service ps redis

# Activate a Node

docker node update --availability active hostname\_Worker1

Network

Overlay Network that spans multiple Docker hosts each running an Engine

master

docker network create -d overlay nginx-net1

docker service create --name nginx-service1 --publish target=80,published=5051 --replicas=5 --network nginx-net1 nginx

docker service ls

docker network inspect nginx-net1

Run the above command in Worker1 and Worker2 as well

docker service inspect nginx-service1 --pretty

docker service rm nginx-service1

docker network rm nginx-net1

Port forwarding

docker service create --mode global --publish mode=host,target=80,published=8080 --name=nginx nginx:latest

docker service create --name my\_web --replicas 3 --publish published=5050,target=80 nginx

docker service ps my\_web

Volumes

docker volume create volume1

docker volume ls

docker service create -d --replicas=4 --name replicated-service --mount source=volume1,target=/app nginx:latest

docker service ps replicated-service

Check in Worker1 and Worker2 Nodes

docker volume ls

Troubleshooting

docker service logs redis

sudo apt install jq

docker service ps redis

Get an ID of container

docker inspect CONTAINER\_ID | jq '.[].State'

docker inspect CONTAINER\_ID | jq '.[].NetworkSettings'

docker inspect CONTAINER\_ID | jq '.[].NetworkSettings.Networks.ingress.IPAddress'

RESET Docker Swarm

At Worker Nodes,

sudo docker swarm leave

At Master,

docker swarm leave -f

Verify,

docker node ls