DATE:11-Mar-2025

Docker Swarm Setup and Management Notes

1️⃣ Install Docker on All Nodes

Run on **all nodes (manager & workers):**

sudo apt update -y

sudo apt install -y docker.io

sudo systemctl enable docker

sudo systemctl start docker

Verify installation:

docker --version

2️⃣ Initialize Swarm on the Manager Node

Run on the **manager node**:

docker swarm init --advertise-addr <MANAGER_PRIVATE_IP>

Output includes a **join command** for worker nodes:

docker swarm join --token <TOKEN> <MANAGER_PRIVATE_IP>:2377

3️⃣ Add Worker Nodes to Swarm

Run on **each worker node**:

docker swarm join --token <TOKEN> <MANAGER_PRIVATE_IP>:2377

Verify nodes from the manager:

docker node ls

4️⃣ Create and Manage Services

Create a Service:

docker service create --name <SERVICE_NAME> --replicas=<NUMBER> -p <HOST_PORT>:80 <IMAGE_NAME>

Example:

docker service create --name myweb --replicas=3 -p 32000:80 nginx:1.17

List Running Services:

docker service ls

Check Service Status:

docker service ps <SERVICE_NAME>

5️⃣ Scaling Services

Increase/Decrease Replicas:

docker service scale <SERVICE_NAME>=<NUMBER>

Example:

docker service scale myweb=5

- **Scaling up:** New containers are created.
- **Scaling down:** Unused containers are removed.

Stop All Containers in a Service (Scale to Zero)

docker service scale myweb=0

6️⃣ Removing Services

docker service rm <SERVICE_NAME>

Example:

docker service rm myweb

7️⃣ Handling Common Errors

Port Conflict Issue:

Error:

rpc error: code = InvalidArgument desc = port '<PORT>' is already in use

Solution:

- Remove the existing service before re-creating it:
- docker service rm myweb
- Use a different port:
- docker service create --name myweb -p 33000:80 nginx:1.17

Service Not Found Error:

Error:

no such service: <SERVICE_NAME>

Solution:

- Check running services:
- docker service ls
- Ensure you are using the correct service name.

Cannot Remove Running Containers (Swarm Mode):

Error:

cannot remove container: container is running

Solution:

- Stop the container first:
- docker stop <CONTAINER_ID>
- docker rm <CONTAINER_ID>
- Alternatively, scale service to 0 first:
- docker service scale myweb=0

8️⃣ Node Availability Management

Check Nodes:

docker node ls

Drain a Node (Prevent Scheduling New Containers)

docker node update --availability drain <NODE_NAME>

Example:

docker node update --availability drain master

- Containers running on **master** will be rescheduled on other available nodes.

Reactivate a Drained Node:

docker node update --availability active <NODE_NAME>

Example:

docker node update --availability active master

9️⃣ Creating a Large-Scale Service (Example: 10 Replicas)

docker service create --name testcontainer --replicas=10 nginx

✅ **Result:** 10 containers distributed across available nodes.

🔹 **Summary of Key Commands**

Command	Description
docker swarm init	Initialize Swarm mode on a manager node
docker swarm join	Add a worker node to the Swarm
docker node ls	List all nodes in the Swarm
docker service create	Create a new service
docker service ls	List all running services
docker service ps <service>	Show details of a service

Command	Description
docker service scale <service>=<n>	Scale a service to N replicas
docker service rm <service>	Remove a service
docker node update --availability drain <node>	Prevent scheduling containers on a node
docker node update --availability active <node>	Reactivate a drained node