Docker Lab 11 — Setting Up a Docker Swarm Cluster

Author: Dr. Sandeep Kumar Sharma

Lab Description

Lab 11 introduces **Docker Swarm**, Docker's native orchestration engine used for building multi-node container clusters. This lab will guide you through setting up a full Swarm cluster, configuring manager and worker nodes, deploying services, and validating distributed workloads.

Topics Covered

- What is Docker Swarm?
- Manager vs Worker nodes
- Creating a Swarm cluster
- Joining nodes to the cluster
- Deploying distributed services
- Inspecting nodes, tasks, and services

Learning Objectives

By the end of this lab, you will be able to: - Initialize a Swarm cluster - Add and manage worker nodes - Deploy replicated services - Inspect cluster state and service distribution

---

Learning Outcomes

After this lab, you will: - Understand Swarm architecture and Raft consensus - Be able to scale services across nodes - Manage multi-host container deployments

---

Section 1 — Prepare 3 Virtual Machines

For this lab, you need: - **Node 1:** swarm-manager - **Node 2:** swarm-worker1 - **Node 3:** swarm-worker2

Install Docker on all 3 nodes.

Section 2 — Initialize Swarm on Manager Node

On **swarm-manager**:

```
docker swarm init --advertise-addr <manager-ip>
```

You will get a worker join command like:

```
docker swarm join --token <token> <manager-ip>:2377
```

Save this.

Section 3 — Join Worker Nodes

On **swarm-worker1** and **swarm-worker2**:

```
docker swarm join --token <token> <manager-ip>:2377
```

Verify from manager:

```
docker node ls
```

You should see:

```
NAME            STATUS    ROLE    AVAILABILITY
swarm-manager   Ready     Manager Active
swarm-worker1   Ready     Worker  Active
swarm-worker2   Ready     Worker  Active
```

Section 4 — Deploy a Replicated Service

On manager:

```
docker service create --name sandeep-svc --replicas 3 nginx
```

Check tasks:

```
docker service ps sandeep-svc
```

Each replica will be scheduled across nodes.

Section 5 — Scale the Service

```
docker service scale sandeep-svc=6
```

View distribution:

```
docker service ps sandeep-svc
```

Section 6 — Inspect Swarm Architecture

Inspect nodes:

```
docker node inspect swarm-manager --pretty
```

Inspect service:

```
docker service inspect sandeep-svc
```

Inspect tasks:

```
docker service ps sandeep-svc
```

Section 7 — Rolling Update

```
docker service update --image nginx:alpine sandeep-svc
```

Docker automatically performs rolling updates.

Rollback if needed:

```
docker service rollback sandeep-svc
```

Section 8 — Cleanup

```
docker service rm sandeep-svc
```

On workers:

```
docker swarm leave
```

On manager:

```
docker swarm leave --force
```

Summary

You have successfully created a Docker Swarm cluster with multiple nodes, deployed replicated services, scaled them, and performed rolling updates.