Docker Lab 13 — Swarm Load Balancing, Rolling Updates & Rollbacks

Author: Dr. Sandeep Kumar Sharma

Lab Description

Lab 13 focuses on three major production-grade features of Docker Swarm: 1. **Load Balancing** 2. **Rolling Updates** 3. **Rollbacks**

Using Swarm, you can deploy services that automatically balance traffic across multiple replicas, update versions with zero downtime, and instantly roll back if errors occur.

This lab demonstrates these features step-by-step using an Nginx-based service.

Topics Covered

- Swarm internal load balancer (Ingress + VIP model)
- Service discovery inside Swarm overlay network
- Zero-downtime rolling updates
- Health checks & update configurations
- Performing rollbacks

Learning Objectives

By the end of this lab, you will be able to: - Understand Swarm's built-in load balancing architecture - Configure rolling updates with parallelism & delays - Roll back failed deployments automatically

Learning Outcomes

After this lab, you will: - Deploy resilient, auto-balancing Swarm services - Perform safe version upgrades - Use rollbacks to recover from failed releases

Section 1 — Prerequisites

A Docker Swarm cluster from Lab 11: - 1 Manager - 2 Workers

Section 2 — Deploy Load Balanced Service

Create a service with 5 replicas:

```
docker service create
  --name sandeep-web
  --publish 8085:80
  --replicas 5
  nginx:1.21
```

Check distribution:

```
docker service ps sandeep-web
```

Swarm's load balancer automatically spreads requests across replicas.

Section 3 — Test Load Balancing

Run curl multiple times:

```
for i in {1..10}; do curl -s http://<manager-ip>:8085 | grep title; done
```

Replica responses will vary.

Section 4 — Configure Rolling Update Parameters

Set rolling update config:

```
docker service update
  --update-parallelism 2
  --update-delay 5s
  --update-monitor 10s
  --update-failure-action rollback
  sandeep-web
```

Section 5 — Perform Rolling Update

Update image:

```
docker service update --image nginx:1.23 sandeep-web
```

Monitor update:

```
docker service ps sandeep-web
```

Section 6 — Simulate Failed Update

Try updating to a faulty image:

```
docker service update --image nginx:doesnotexist sandeep-web
```

Swarm detects failure and triggers rollback because of the earlier configuration.

Check status:

```
docker service ps sandeep-web
```

It should be restored to the previous working version.

Section 7 — Manual Rollback

If required:

```
docker service rollback sandeep-web
```

Section 8 — Cleanup

```
docker service rm sandeep-web
```

Summary

You deployed a load-balanced, multi-replica Swarm service, configured rolling updates, tested zero-downtime upgrades, and validated rollback behavior. This is essential for real-world production deployments.