**Step 1: Initialize Docker Swarm**

1. **On the manager node**:

**docker swarm init**

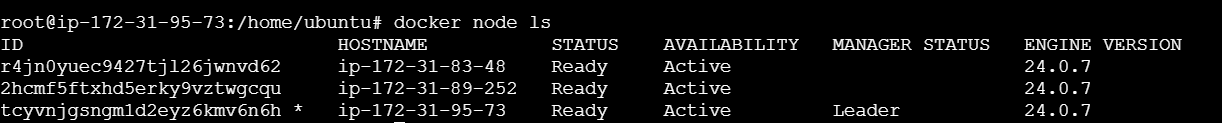
This will output a **join token** for worker nodes.

1. **Join worker nodes** (on the other two EC2 instances): Use the token from the manager node to join the swarm:

**docker swarm join --token SWMTKN-1-2vhxmrys2htn2zj962v4i0ayhw23qi7t2yk3boqbow90n9dzah-9en3y1e58vre7pqoxo3668nfv 172.31.95.73:2377**

1. Verify the nodes have joined by running this on the manager node:

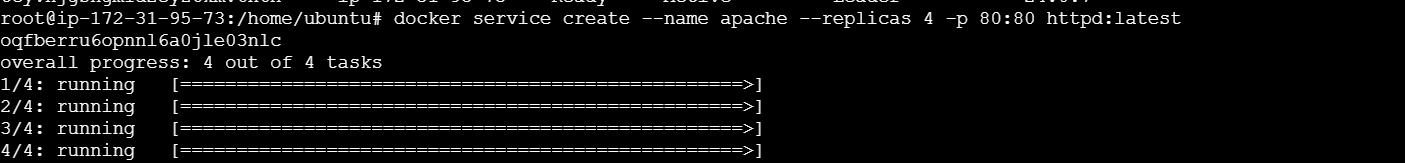
**docker node ls**

****

**Step 2: Deploy Apache Container with 4 Replicas**

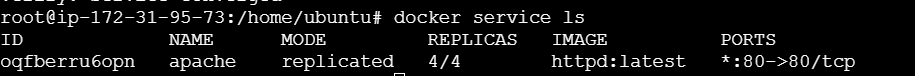
1. **Create a service on the manager node** to deploy the Apache container:

**docker service create --name apache --replicas 4 -p 80:80 httpd:latest**



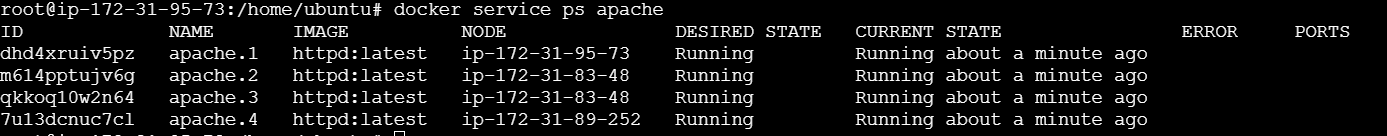
1. Verify the service is running:

**docker service ls**



1. Verify that the Apache containers are distributed across the nodes:

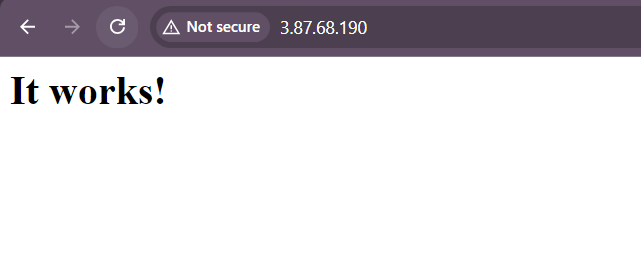
**docker service ps apache**



**Step 3: Access the Apache Service**

1. **Get the public IP** of any of your EC2 instances and open it in a browser.

**http://3.87.68.190/**



That's it! You now have a 3-node Docker Swarm cluster with an Apache container running with 4 replicas on AWS EC2.

**2. Create an Overlay Network**

1. **Create an overlay network**:

**docker network create --driver overlay my\_overlay\_network**

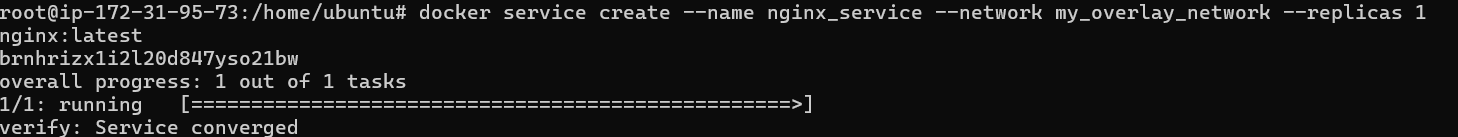
****

**3. Deploy Containers on the Overlay Network**

We'll deploy two simple containers: one running a basic nginx server and the other running a busybox container for testing.

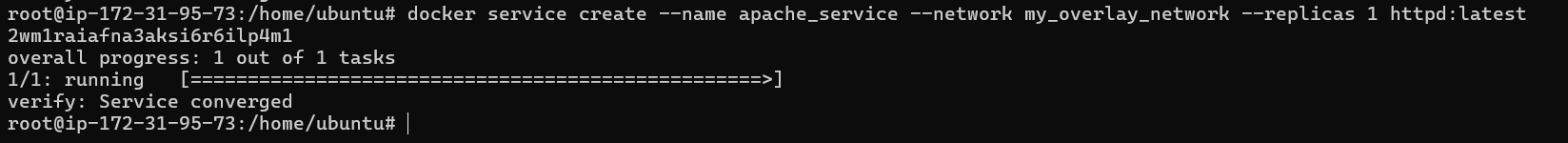
1. **Deploy the nginx container**:

**docker service create --name nginx\_service --network my\_overlay\_network --replicas 1 nginx:latest**



1. **Deploy the apache container**:

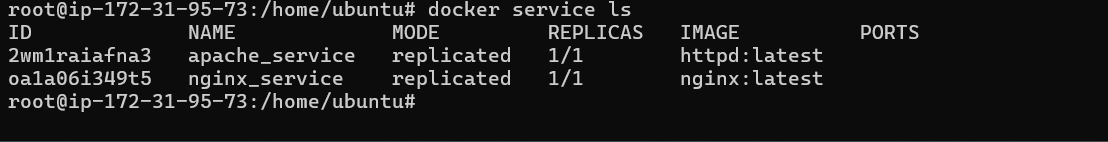
**docker service create --name apache\_service --network my\_overlay\_network --replicas 1 httpd:latest**



**4. Verify the Services**

Check that the services are up and running:

**docker service ls**

****

1. **Ping the apache container** from the nginx container:

