Lesson 04 Demo 02

Running Replicated and Global Services

Objective: To demonstrate Docker's service deployment versatility by creating replicated and global services, allowing users to manage containerized applications efficiently within a swarm cluster

Tools required: Docker and Ubuntu OS

Prerequisites: None

Steps to be followed:

1. Create a replicated service
2. Create a global service
3. List all Docker services
4. Check the status of global services
5. Check the status of replicated services

Step 1: Create a replicated service

1.1 Run the following command to create a replicated service using the nginx image:
sudo docker service create --name myweb --replicas 3 nginx

```
labsuser@ip-172-31-29-216:~$ sudo docker service create --name myweb --replicas 3 nginx
63n9xqt5k5ufcmdfczw9ft9ij
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
labsuser@ip-172-31-29-216:~$
```

Step 2: Create a global service

2.1 Create a global service with the Alpine image using the following command:

**sudo docker service create **
**--name myservice **
--mode global alpine top

```
labsuser@ip-172-31-29-216:~$ sudo docker service create \
> --name myservice \
> --mode global alpine top
e3fbcgtamdz1e9g4fotoih6yl
overall progress: 3 out of 3 tasks
qh2u7kcfnepf: running   [==================================================>]
2ok547v07m3s: running   [==================================================>]
e5a7syx7911y: running   [==================================================>]
verify: Service converged
labsuser@ip-172-31-29-216:~$
```

Step 3: List all Docker services

3.1 Use the command to list all Docker services:

sudo docker service ls

```
labsuser@ip-172-31-29-216:~$ sudo docker service ls
ID             NAME        MODE         REPLICAS   IMAGE          PORTS
e3fbcgtamdz1   myservice   global       3/3        alpine:latest
63n9xqt5k5uf   myweb       replicated   3/3        nginx:latest
labsuser@ip-172-31-29-216:~$
```

Step 4: Check the status of global services

4.1 Verify the status of the global service **myservice** by running the following command:
sudo docker service ps myservice

```
labsuser@ip-172-31-29-216:~$ sudo docker service ps myservice
ID             NAME                                    IMAGE          NODE               DESIRED STATE   CURRENT STATE
qiaaizdynrb2   myservice.2ok547v07m3sy4bnhruvtke70      alpine:latest  ip-172-31-26-147   Running         Running 2 minutes ago
qsv7ez52tpvj   myservice.e5a7syx7911yluh83wceditdd      alpine:latest  ip-172-31-29-216   Running         Running 2 minutes ago
nd6x4nfr9a2c   myservice.qh2u7kcfnepf2e1zp69j10kbv      alpine:latest  ip-172-31-30-210   Running         Running 2 minutes ago
labsuser@ip-172-31-29-216:~$
```

> **Note:** The global service is running its instances on all the nodes of the swarm cluster.

Step 5: Check the status of replicated services

5.1 Verify the status of the replicated service **myweb** by executing the following command:
sudo docker service ps myweb

```
labsuser@ip-172-31-29-216:~$ sudo docker service ps myweb
ID             NAME      IMAGE          NODE               DESIRED STATE   CURRENT STATE
3leywxiv5vam   myweb.1   nginx:latest   ip-172-31-30-210   Running         Running 9 minutes ago
9fdd56cx6o19   myweb.2   nginx:latest   ip-172-31-26-147   Running         Running 9 minutes ago
7zzb2pifo93a   myweb.3   nginx:latest   ip-172-31-29-216   Running         Running 9 minutes ago
labsuser@ip-172-31-29-216:~$
```

> **Note:** The replicated service is running the service replicas on all the nodes of the swarm cluster.

By following these steps, you have successfully demonstrated Docker's versatility in deploying replicated and global services within a swarm cluster. This understanding allows for efficient management and scalability of containerized applications across distributed environments.