Lesson 04 Demo 04

Inspecting a Service on Swarm

Objective: To inspect a service on a Docker swarm for monitoring its task distribution across nodes and extracting comprehensive service details in various readable formats

Tools required: Docker and Ubuntu OS

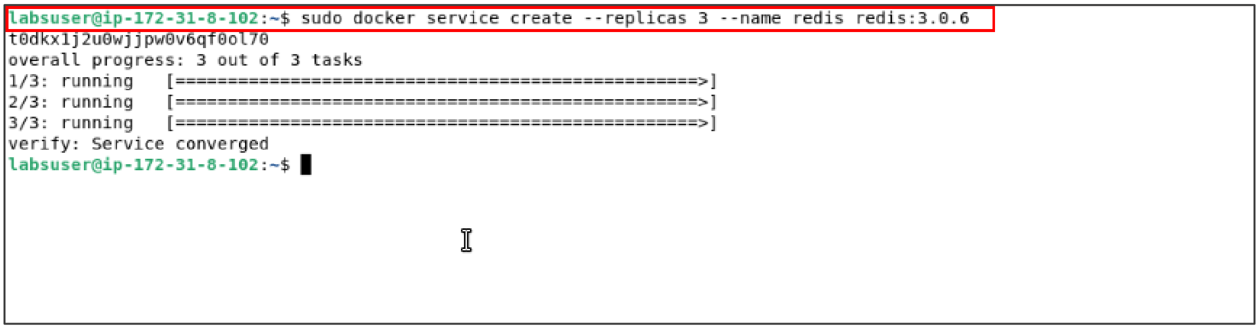
Prerequisites: None

Steps to be followed:

1. Deploy a replicated Redis service
2. Inspect the service on the Docker swarm

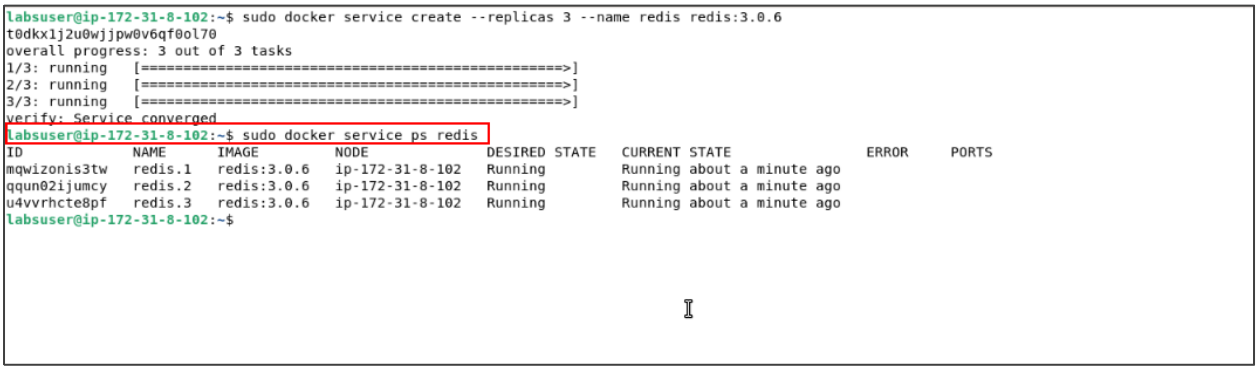
Step 1: Deploy a replicated Redis service

1.1 Run the following command to start a replicated service using the Redis image:
sudo docker service create --replicas 3 --name redis redis:3.0.6



```
labsuser@ip-172-31-8-102:~$ sudo docker service create --replicas 3 --name redis redis:3.0.6
t0dkx1j2u0wjjpw0v6qf0ol70
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
labsuser@ip-172-31-8-102:~$ 
```

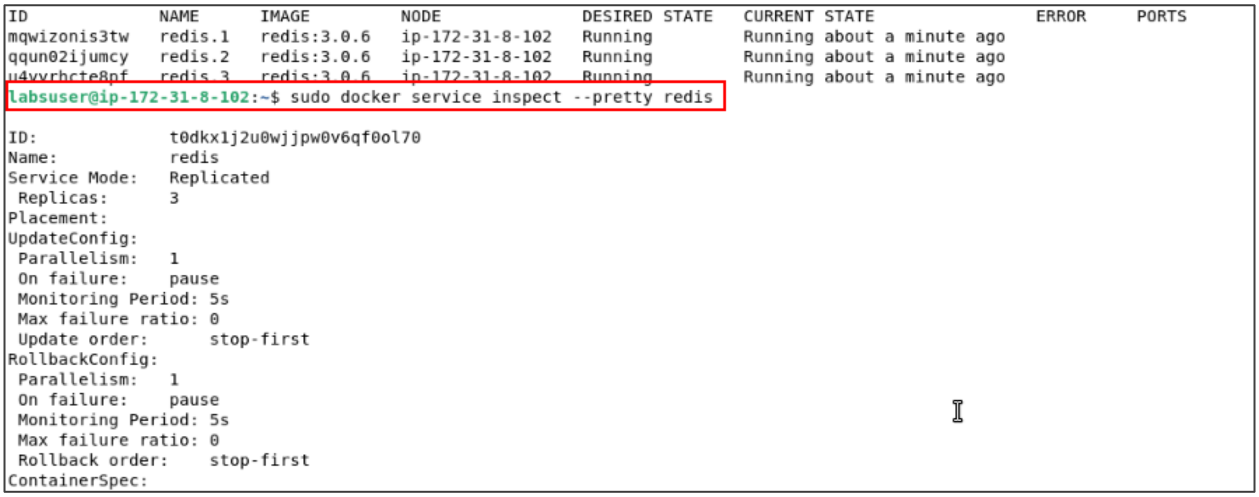
1.2 Run the following command to check the service tasks assigned to different nodes by the swarm manager:
sudo docker service ps redis



```
labsuser@ip-172-31-8-102:~$ sudo docker service create --replicas 3 --name redis redis:3.0.6
t0dkx1j2u0wjjpw0v6qf0ol70
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
labsuser@ip-172-31-8-102:~$ sudo docker service ps redis
ID             NAME      IMAGE         NODE              DESIRED STATE   CURRENT STATE              ERROR     PORTS
mqwizonis3tw   redis.1   redis:3.0.6   ip-172-31-8-102   Running         Running about a minute ago
qqun02ijumcy   redis.2   redis:3.0.6   ip-172-31-8-102   Running         Running about a minute ago
u4vvrhcte8pf   redis.3   redis:3.0.6   ip-172-31-8-102   Running         Running about a minute ago
labsuser@ip-172-31-8-102:~$
```

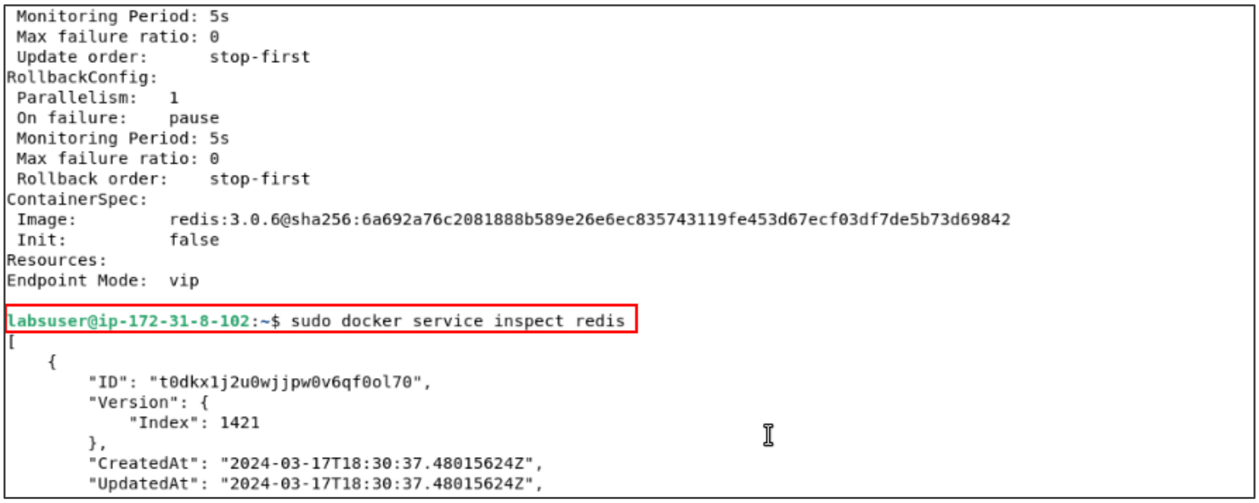
Step 2: Inspect the service on the Docker swarm

2.1 Use the following command to inspect the Redis service:
sudo docker service inspect --pretty redis

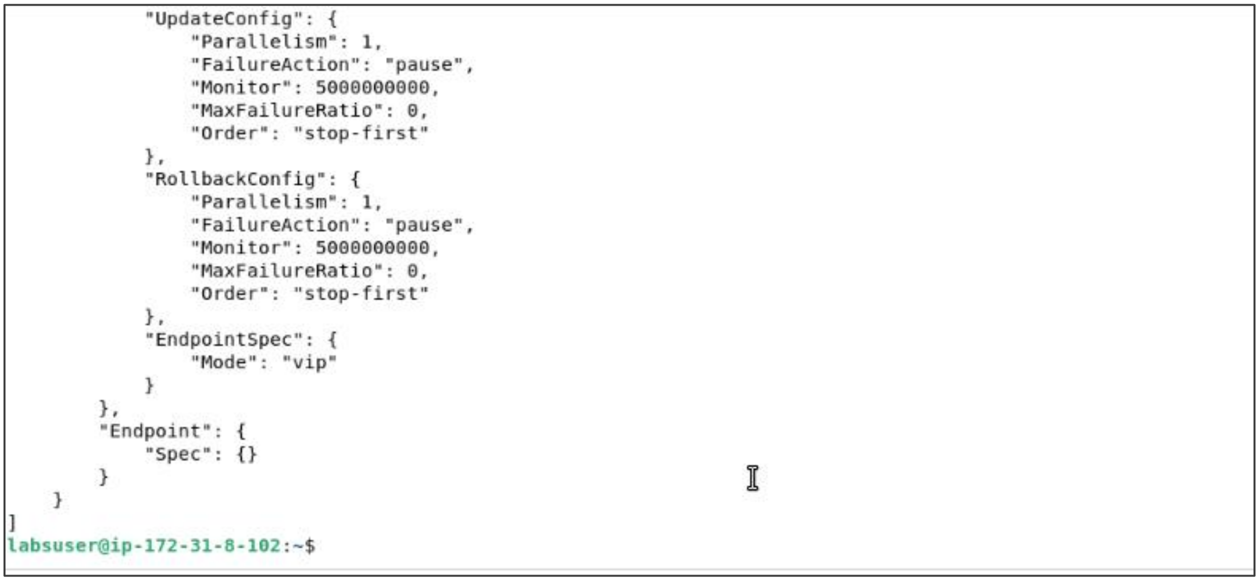


```
ID             NAME      IMAGE         NODE              DESIRED STATE   CURRENT STATE              ERROR     PORTS
mqwizonis3tw   redis.1   redis:3.0.6   ip-172-31-8-102   Running         Running about a minute ago
qqun02ijumcy   redis.2   redis:3.0.6   ip-172-31-8-102   Running         Running about a minute ago
u4vvrhcte8pf   redis.3   redis:3.0.6   ip-172-31-8-102   Running         Running about a minute ago
labsuser@ip-172-31-8-102:~$ sudo docker service inspect --pretty redis

ID:             t0dkx1j2u0wjjpw0v6qf0ol70
Name:           redis
Service Mode:   Replicated
 Replicas:      3
Placement:
UpdateConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
```

2.2 Use the following command to return the service details in JSON format:
sudo docker service inspect redis



```
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
 Image:         redis:3.0.6@sha256:6a692a76c2081888b589e26e6ec835743119fe453d67ecf03df7de5b73d69842
 Init:          false
Resources:
Endpoint Mode:  vip

labsuser@ip-172-31-8-102:~$ sudo docker service inspect redis
[
    {
        "ID": "t0dkx1j2u0wjjpw0v6qf0ol70",
        "Version": {
            "Index": 1421
        },
        "CreatedAt": "2024-03-17T18:30:37.48015624Z",
        "UpdatedAt": "2024-03-17T18:30:37.48015624Z",
```



```
            "UpdateConfig": {
                "Parallelism": 1,
                "FailureAction": "pause",
                "Monitor": 5000000000,
                "MaxFailureRatio": 0,
                "Order": "stop-first"
            },
            "RollbackConfig": {
                "Parallelism": 1,
                "FailureAction": "pause",
                "Monitor": 5000000000,
                "MaxFailureRatio": 0,
                "Order": "stop-first"
            },
            "EndpointSpec": {
                "Mode": "vip"
            }
        },
        "Endpoint": {
            "Spec": {}
        }
    }
]
labsuser@ip-172-31-8-102:~$
```

By following these steps, you have successfully inspected a service on a Docker swarm to ensure efficient monitoring of task distribution and obtaining detailed service information in various formats.