.

Lesson 04 Demo 08

Troubleshooting an Undeployable Docker Service

Objective: To troubleshoot an undeployable in a Docker swarm by identifying configuration issues, checking task status, inspecting containers, and analyzing logs to resolve deployment failures

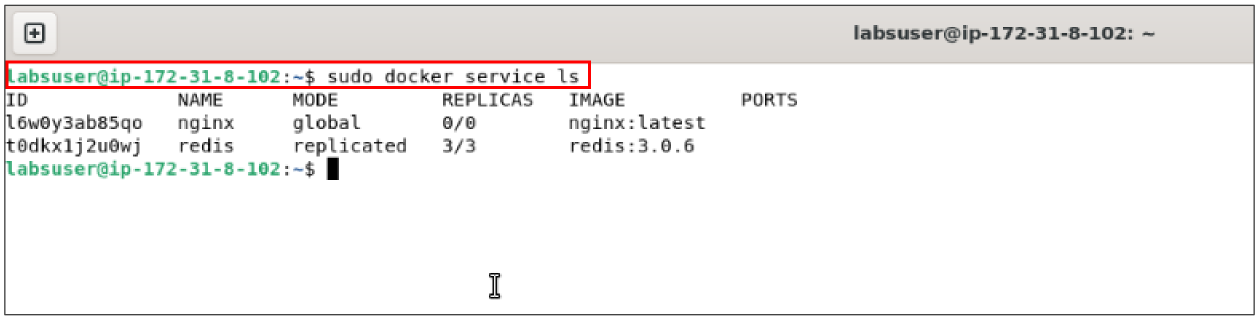
Tools required: Docker and Ubuntu OS

Prerequisites: None

Steps to be followed:
1. List and inspect the Docker services
2. Inspect and diagnose service tasks

Step 1: List and inspect the Docker services

1.1 Run the following command to list all the services running on the swarm:
sudo docker service ls

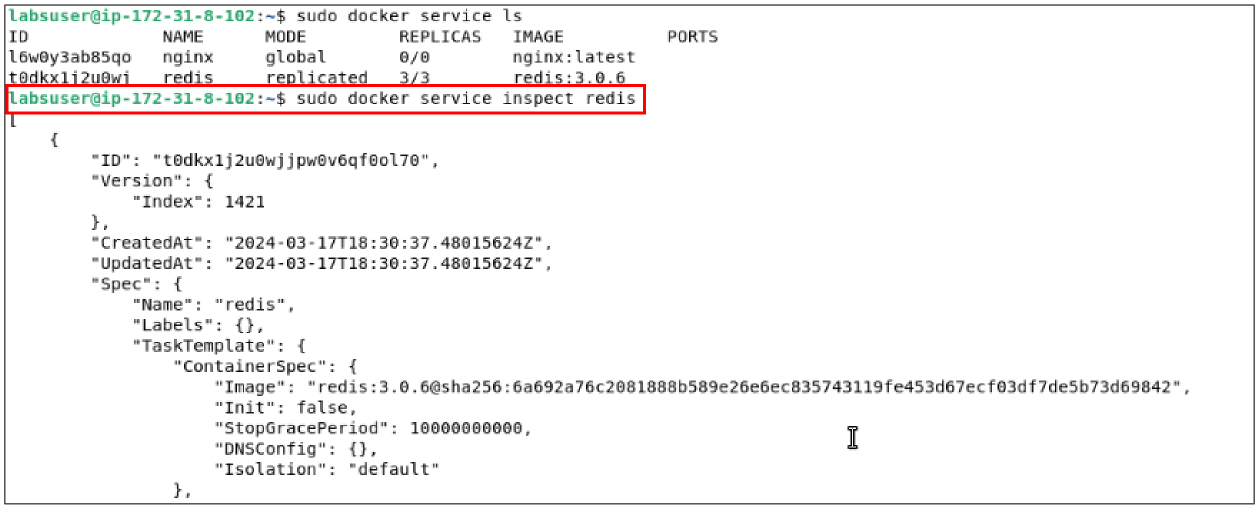


```
labsuser@ip-172-31-8-102: ~
labsuser@ip-172-31-8-102:~$ sudo docker service ls
ID             NAME      MODE         REPLICAS   IMAGE          PORTS
l6w0y3ab85qo   nginx     global       0/0        nginx:latest
t0dkx1j2u0wj   redis     replicated   3/3        redis:3.0.6
labsuser@ip-172-31-8-102:~$
```

Note: Check whether all the replicas for a service are running

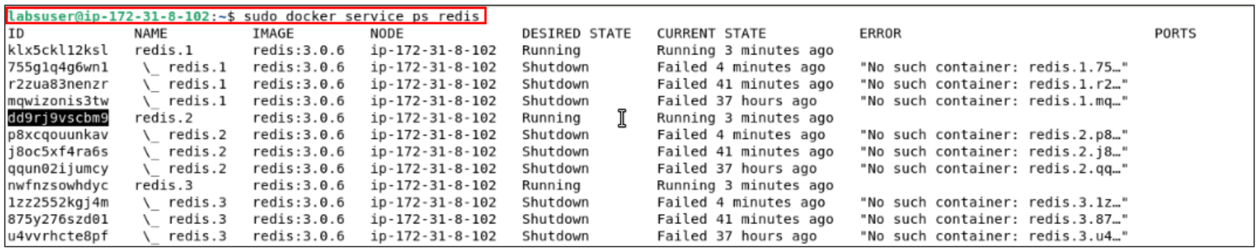
.

1.2 Run the following command to inspect the Redis service for configurations:
sudo docker service inspect redis



```
labsuser@ip-172-31-8-102:~$ sudo docker service ls
ID             NAME      MODE         REPLICAS   IMAGE          PORTS
l6w0y3ab85qo   nginx     global       0/0        nginx:latest
t0dkx1j2u0wj   redis     replicated   3/3        redis:3.0.6
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",
        "Spec": {
            "Name": "redis",
            "Labels": {},
            "TaskTemplate": {
                "ContainerSpec": {
                    "Image": "redis:3.0.6@sha256:6a692a76c2081888b589e26e6ec835743119fe453d67ecf03df7de5b73d69842",
                    "Init": false,
                    "StopGracePeriod": 10000000000,
                    "DNSConfig": {},
                    "Isolation": "default"
                },
```

1.3 List the tasks assigned to a specific service to determine their current operational status:
sudo docker service ps redis

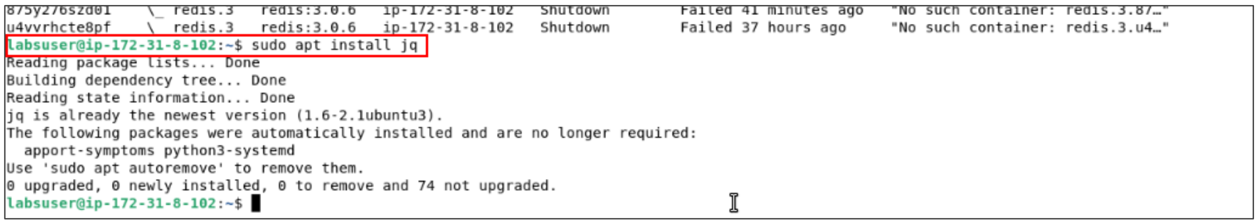


```
labsuser@ip-172-31-8-102:~$ sudo docker service ps redis
ID             NAME          IMAGE         NODE              DESIRED STATE   CURRENT STATE            ERROR                              PORTS
klx5ckl12ksl   redis.1       redis:3.0.6   ip-172-31-8-102   Running         Running 3 minutes ago
755g1q4g6wn1    \_ redis.1   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 4 minutes ago     "No such container: redis.1.75…"
r2zua83nenzr    \_ redis.1   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 41 minutes ago    "No such container: redis.1.r2…"
mqwizonis3tw    \_ redis.1   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 37 hours ago      "No such container: redis.1.mq…"
dd9rj9vscbm9   redis.2       redis:3.0.6   ip-172-31-8-102   Running         Running 3 minutes ago
p8xcqouunkav    \_ redis.2   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 4 minutes ago     "No such container: redis.2.p8…"
j8oc5xf4ra6s    \_ redis.2   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 41 minutes ago    "No such container: redis.2.j8…"
qqun02ijumcy    \_ redis.2   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 37 hours ago      "No such container: redis.2.qq…"
nwfnzsowhdyc   redis.3       redis:3.0.6   ip-172-31-8-102   Running         Running 3 minutes ago
1zz2552kgj4m    \_ redis.3   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 4 minutes ago     "No such container: redis.3.1z…"
875y276szd01    \_ redis.3   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 41 minutes ago    "No such container: redis.3.87…"
u4vvrhcte8pf    \_ redis.3   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 37 hours ago      "No such container: redis.3.u4…"
```

Note: Copy the ID of the Redis task running on the node; you will use it in the next steps.

Step 2: Inspect and diagnose service tasks

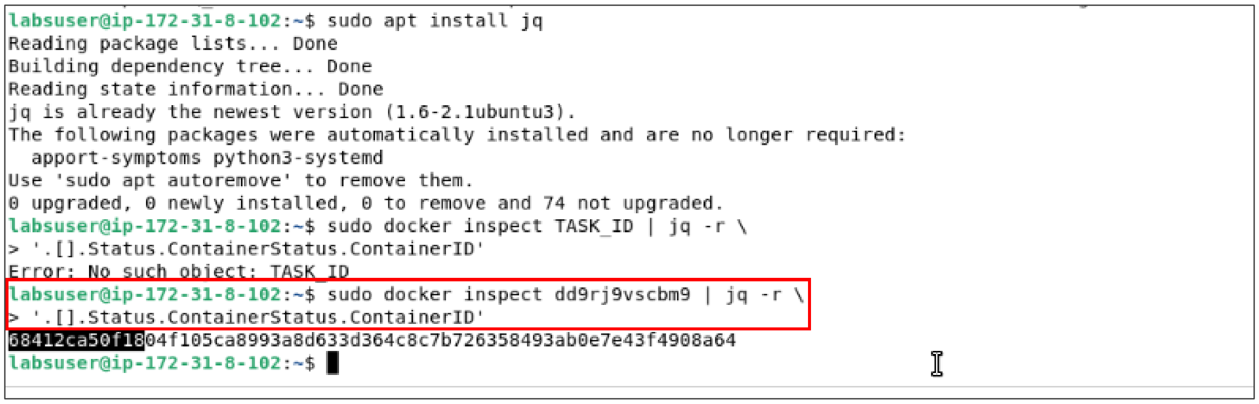
2.1 Run the following command to install the jq package:
sudo apt install jq



```
875y276szd01    \_ redis.3   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 41 minutes ago    "No such container: redis.3.87…"
u4vvrhcte8pf    \_ redis.3   redis:3.0.6   ip-172-31-8-102   Shutdown        Failed 37 hours ago      "No such container: redis.3.u4…"
labsuser@ip-172-31-8-102:~$ sudo apt install jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.6-2.1ubuntu3).
The following packages were automatically installed and are no longer required:
  apport-symptoms python3-systemd
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
labsuser@ip-172-31-8-102:~$
```

.

2.2 Execute the following command to inspect a specific task and identify the container associated with the provided service:
**sudo docker inspect TASK_ID | jq -r \
'.[].Status.ContainerStatus.ContainerID'**

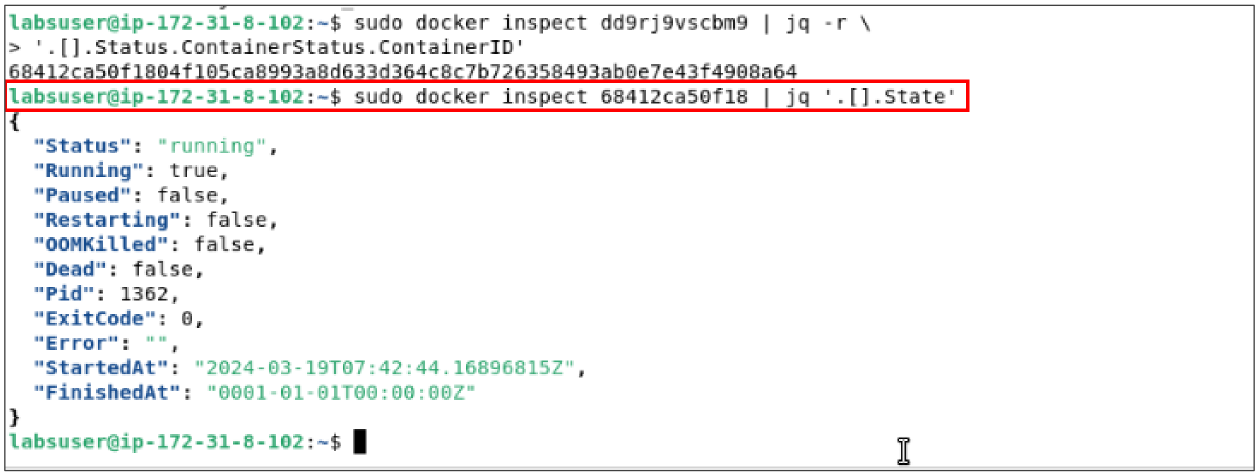


```
labsuser@ip-172-31-8-102:~$ sudo apt install jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.6-2.1ubuntu3).
The following packages were automatically installed and are no longer required:
  apport-symptoms python3-systemd
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
labsuser@ip-172-31-8-102:~$ sudo docker inspect TASK_ID | jq -r \
> '.[].Status.ContainerStatus.ContainerID'
Error: No such object: TASK_ID
labsuser@ip-172-31-8-102:~$ sudo docker inspect dd9rj9vscbm9 | jq -r \
> '.[].Status.ContainerStatus.ContainerID'
68412ca50f1804f105ca8993a8d633d364c8c7b726358493ab0e7e43f4908a64
labsuser@ip-172-31-8-102:~$
```

Note: Replace **TASK_ID** with the ID of a task selected from step 1.3. In this case, the task ID is **dd9rj9vscbm9**.

Note: Copy the first 12 characters of the container ID, as shown in the screenshot above. You will use them in the next steps.

2.3 Execute the following command to get the container state:
sudo docker inspect CONTAINER_ID | jq '.[].State'



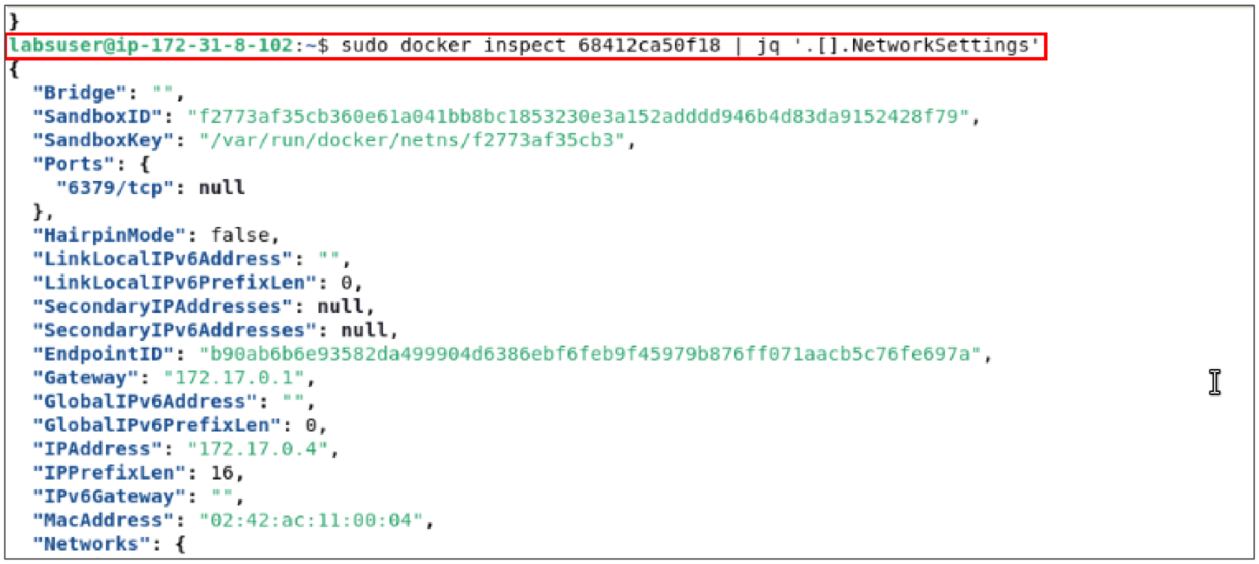
```
labsuser@ip-172-31-8-102:~$ sudo docker inspect dd9rj9vscbm9 | jq -r \
> '.[].Status.ContainerStatus.ContainerID'
68412ca50f1804f105ca8993a8d633d364c8c7b726358493ab0e7e43f4908a64
labsuser@ip-172-31-8-102:~$ sudo docker inspect 68412ca50f18 | jq '.[].State'
{
  "Status": "running",
  "Running": true,
  "Paused": false,
  "Restarting": false,
  "OOMKilled": false,
  "Dead": false,
  "Pid": 1362,
  "ExitCode": 0,
  "Error": "",
  "StartedAt": "2024-03-19T07:42:44.168968152",
  "FinishedAt": "0001-01-01T00:00:00Z"
}
labsuser@ip-172-31-8-102:~$
```

Note: Replace **CONTAINER_ID** with the first 12 characters of the container ID received in the previous step. In this case, the Container ID is **68412ca50f18.**

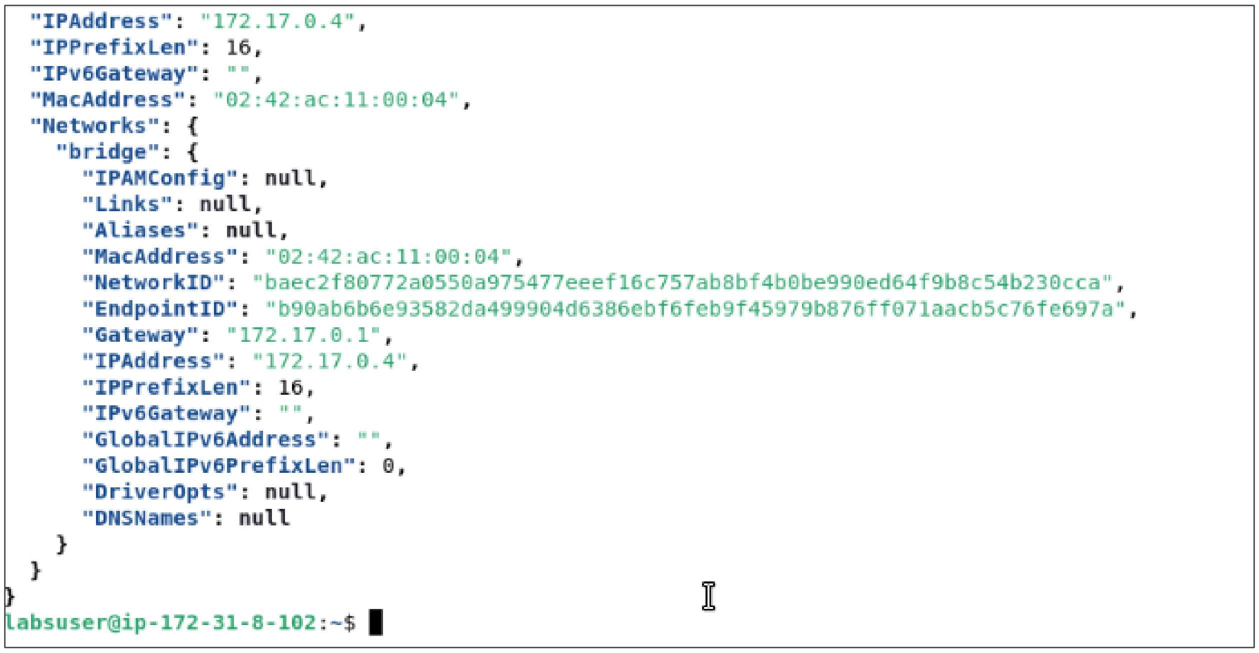
.

Note: The **ExitCode** and **Error** tags show why a container exited.

2.4 Run the following command to check the network settings of the container:
sudo docker inspect CONTAINER_ID | jq '.[].NetworkSettings'



```
}
labsuser@ip-172-31-8-102:~$ sudo docker inspect 68412ca50f18 | jq '.[].NetworkSettings'
{
  "Bridge": "",
  "SandboxID": "f2773af35cb360e61a041bb8bc1853230e3a152adddd946b4d83da9152428f79",
  "SandboxKey": "/var/run/docker/netns/f2773af35cb3",
  "Ports": {
    "6379/tcp": null
  },
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "b90ab6b6e93582da499904d6386ebf6feb9f45979b876ff071aacb5c76fe697a",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.4",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:04",
  "Networks": {
```

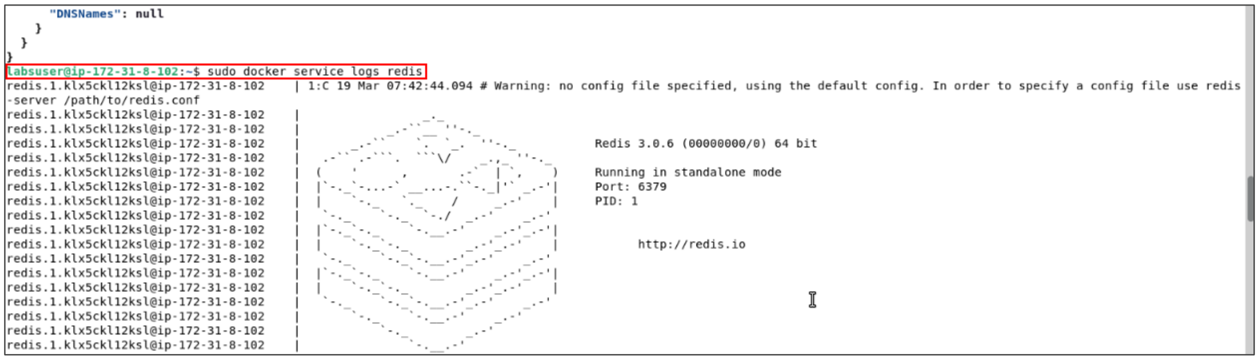


```
  "IPAddress": "172.17.0.4",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:04",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "MacAddress": "02:42:ac:11:00:04",
      "NetworkID": "baec2f80772a0550a975477eeef16c757ab8bf4b0be990ed64f9b8c54b230cca",
      "EndpointID": "b90ab6b6e93582da499904d6386ebf6feb9f45979b876ff071aacb5c76fe697a",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.4",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "DriverOpts": null,
      "DNSNames": null
    }
  }
}
labsuser@ip-172-31-8-102:~$ 
```

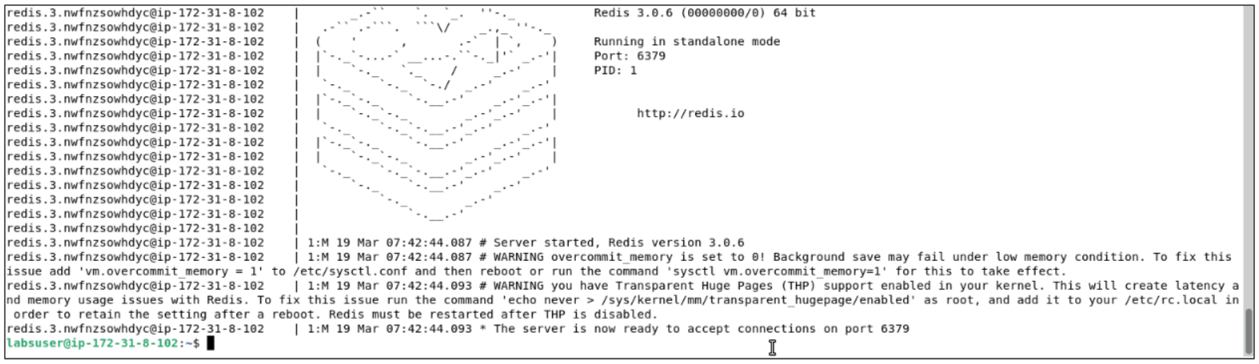
Note: Replace **CONTAINER_ID** with the first 12 characters of the container ID received in the previous step. In this case, the container ID is **68412ca50f18.**

.

2.5 Execute the following command to get detailed logs of the Redis service:
sudo docker service logs redis



```
      "DNSNames": null
    }
  }
}
labsuser@ip-172-31-8-102:~$ sudo docker service logs redis
redis.1.klx5ckl12ksl@ip-172-31-8-102    | 1:C 19 Mar 07:42:44.094 # Warning: no config file specified, using the default config. In order to specify a config file use redis
-server /path/to/redis.conf
redis.1.klx5ckl12ksl@ip-172-31-8-102    |                 _._
redis.1.klx5ckl12ksl@ip-172-31-8-102    |            _.-``__ ''-._
redis.1.klx5ckl12ksl@ip-172-31-8-102    |       _.-``    `.  `_.  ''-._           Redis 3.0.6 (00000000/0) 64 bit
redis.1.klx5ckl12ksl@ip-172-31-8-102    |   .-`` .-```.  ```\/    _.,_ ''-._
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  (    '      ,       .-`  | `,    )     Running in standalone mode
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  |    `-._   `._    /     _.-'    |     PID: 1
redis.1.klx5ckl12ksl@ip-172-31-8-102    |   `-._    `-._  `-./  _.-'    _.-'
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  |`-._`-._    `-.__.-'    _.-'_.-'|
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  |    `-._`-._        _.-'_.-'    |           http://redis.io
redis.1.klx5ckl12ksl@ip-172-31-8-102    |   `-._    `-._`-.__.-'_.-'    _.-'
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  |`-._`-._    `-.__.-'    _.-'_.-'|
redis.1.klx5ckl12ksl@ip-172-31-8-102    |  |    `-._`-._        _.-'_.-'    |
redis.1.klx5ckl12ksl@ip-172-31-8-102    |   `-._    `-._`-.__.-'_.-'    _.-'
redis.1.klx5ckl12ksl@ip-172-31-8-102    |       `-._    `-.__.-'    _.-'
redis.1.klx5ckl12ksl@ip-172-31-8-102    |           `-._        _.-'
redis.1.klx5ckl12ksl@ip-172-31-8-102    |               `-.__.-'
```



```
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |       _.-``    `.  `_.  ''-._           Redis 3.0.6 (00000000/0) 64 bit
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |   .-`` .-```.  ```\/    _.,_ ''-._
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  (    '      ,       .-`  | `,    )     Running in standalone mode
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  |    `-._   `._    /     _.-'    |     PID: 1
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |   `-._    `-._  `-./  _.-'    _.-'
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  |`-._`-._    `-.__.-'    _.-'_.-'|
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  |    `-._`-._        _.-'_.-'    |           http://redis.io
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |   `-._    `-._`-.__.-'_.-'    _.-'
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  |`-._`-._    `-.__.-'    _.-'_.-'|
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |  |    `-._`-._        _.-'_.-'    |
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |   `-._    `-._`-.__.-'_.-'    _.-'
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |       `-._    `-.__.-'    _.-'
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |           `-._        _.-'
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |               `-.__.-'
redis.3.nwfnzsowhdyc@ip-172-31-8-102    |
redis.3.nwfnzsowhdyc@ip-172-31-8-102    | 1:M 19 Mar 07:42:44.087 # Server started, Redis version 3.0.6
redis.3.nwfnzsowhdyc@ip-172-31-8-102    | 1:M 19 Mar 07:42:44.087 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this
issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis.3.nwfnzsowhdyc@ip-172-31-8-102    | 1:M 19 Mar 07:42:44.093 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency a
nd memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in
 order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
redis.3.nwfnzsowhdyc@ip-172-31-8-102    | 1:M 19 Mar 07:42:44.093 * The server is now ready to accept connections on port 6379
labsuser@ip-172-31-8-102:~$
```

By following these steps, you have successfully diagnosed and resolved issues that were preventing the deployment of a service within a Docker swarm to ensure smooth operation and optimal service availability.