Inspect a service on the swarm

Estimated reading time: 2 minutes

When you have deployed a service (https://docs.docker.com/engine/swarm/swarm-tutorial/deploy-service/) to your swarm, you can use the Docker CLI to see details about the service running in the swarm.

1. If you haven't already, open a terminal and ssh into the machine where you run your manager node. For example, the tutorial uses a machine named `manager1` .

2. Run `docker service inspect --pretty <SERVICE-ID>` to display the details about a service in an easily readable format.

   To see the details on the `helloworld` service:

   ```
   [manager1]$ docker service inspect --pretty helloworld

   ID:             9uk4639qpg7npwf3fn2aasksr
   Name:           helloworld
   Service Mode:   REPLICATED
    Replicas:              1
   Placement:
   UpdateConfig:
    Parallelism:   1
   ContainerSpec:
    Image:         alpine
    Args:   ping docker.com
   Resources:
   Endpoint Mode:  vip
   ```

   > **Tip**: To return the service details in json format, run the same command without the `--pretty` flag.

```
[manager1]$ docker service inspect helloworld
[
{
    "ID": "9uk4639qpg7npwf3fn2aasksr",
    "Version": {
        "Index": 418
    },
    "CreatedAt": "2016-06-16T21:57:11.622222327Z",
    "UpdatedAt": "2016-06-16T21:57:11.622222327Z",
    "Spec": {
        "Name": "helloworld",
        "TaskTemplate": {
            "ContainerSpec": {
                "Image": "alpine",
                "Args": [
                    "ping",
                    "docker.com"
                ]
            },
            "Resources": {
                "Limits": {},
                "Reservations": {}
            },
            "RestartPolicy": {
                "Condition": "any",
                "MaxAttempts": 0
            },
            "Placement": {}
        },
        "Mode": {
            "Replicated": {
                "Replicas": 1
            }
        },
        "UpdateConfig": {
            "Parallelism": 1
        },
        "EndpointSpec": {
            "Mode": "vip"
        }
    },
    "Endpoint": {
        "Spec": {}
    }
}
]
```

3. Run `docker service ps <SERVICE-ID>` to see which nodes are running the service:

```
[manager1]$ docker service ps helloworld

NAME                                    IMAGE    NODE      DES
IRED STATE  LAST STATE
helloworld.1.8p1vev3fq5zm0mi8g0as41w35  alpine   worker2   Run
ning        Running 3 minutes
```

In this case, the one instance of the `helloworld` service is running on
the `worker2` node. You may see the service running on your manager
node. By default, manager nodes in a swarm can execute tasks just like
worker nodes.

Swarm also shows you the `DESIRED STATE` and `LAST STATE` of the
service task so you can see if tasks are running according to the service
definition.

4. Run `docker ps` on the node where the task is running to see details
   about the container for the task.

> **Tip**: If `helloworld` is running on a node other than your
> manager node, you must ssh to that node.

```
[worker2]$docker ps

CONTAINER ID      IMAGE              COMMAND
CREATED           STATUS             PORTS
NAMES
e609dde94e47      alpine:latest      "ping docker.com"
3 minutes ago     Up 3 minutes
helloworld.1.8p1vev3fq5zm0mi8g0as41w35
```

What's next?

Next, you can change the scale
(https://docs.docker.com/engine/swarm/swarm-tutorial/scale-service/) for the
service running in the swarm.

tutorial (https://docs.docker.com/glossary/?term=tutorial), cluster
management (https://docs.docker.com/glossary/?term=cluster management),
swarm mode (https://docs.docker.com/glossary/?term=swarm mode)