QA Consulting

Docker Swarm Introduction

TRAINING MATERIALS - MODULE HANDOUT

Contacts

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview

Swarm mode for Docker is a container orchestration tool exists natively for Docker, meaning that there are no other installation requirements other than Docker. As an orchestration tool, Swarm allows us to cluster Docker containers for redundancy and high availability.

Swarm Clusters

A swarm is a group of machines working together, a cluster. One of these machines are referred to as the Swarm manager node, when other machines join the cluster, they are known as worker nodes. Connections between the nodes are secured by default, using a generated token.

Manager Nodes

This node is where we can execute commands for managing the Swarm, including the nodes inside it. You may have more than one Manager node.

Worker Nodes

The worker's job is to just run containers, complying with the manager nodes requests. Containers for deployed services will be automatically provisioned evenly across worker nodes. When new worker nodes are added, deployed services must be updated or recreated to take advantage of the new nodes.

Ingress

The routing mesh in Docker Swarm allows for services to be load balanced by default. When you access a published service on any node, your request first goes through a load balancer. Which means if you make a request to the manager node, while there are 2 worker nodes, your request may end up on a container on any of the nodes if there is at least one replica of that service on each node.

Tasks

In this exercise we will be initialising Swarm, with one manager and one worker node. For this exercise you will need two machines, ideally on the same network, which can connect to each other. Both the machines must have Docker installed on them.

Initialising the Swarm

We need to create our first node in the Swarm which will serve as a Manager node. On the first machine you have, run the follow command.

```
docker swarm init
```

Adding a Worker Node

At this point the cluster is now setup and you may start creating services, however we will add a worker node first. Upon initilising the Swarm, Docker would have likely prompted you with a command that allows you to join a worker node to the swarm. If not, or if you can't obtain that information anymore, you can use the **join-token** command to get it back.

```
docker swarm join-token worker
```

Once you have obtained the join command, you can simply run it on the second machine that you have. The command will look similar to the one below, however the token and IP address will be different for yours. Also take a moment here to notice which port the workers are connecting from (**2377**), this will be important to allow on locked down networks.

```
docker swarm join --token SWMTKN-1-4y5lnvp7h7l7pt2qusvykiyk8p0hn0yywq3gk47ogau8pl64f9-15
    hstsq0lahr72cz6dppex0i6 10.154.15.224:2377
```

Create Your First Swarm Service

When managing Swarm services, ensure that you are first on the manager node. Here we will create a simple NGINX service and use the curl CLI tool to access it.

```
docker service create --name nginx --publish 80:80 nginx:latest
```

Once you have created the service, it will be accessible on the private IP address of the machine, not the loopback interface (localhost). To access the NGINX service for this example, the curl command below can be used. Keep in mind that your private IP will be different.

```
curl http://10.154.15.224
```

Delete Your First Swarm Service

Let’s remove the service that we created.

```
docker service rm nginx
```