Chapter 7

Lab 7.2

Immutable Infrastructure via Docker Swarm

Prerequisites

Install Docker Engine >1.13 following the instructions for your platform. The instructions are available at https://docs.docker.com/engine/installation/.

This can be done on your local machine or within a Vagrant virtual machine,assuming that there is an appropriate port forwarding.

Install Docker Compose using the instructions provided at https://docs.docker.com/compose/install/.

Introduction

In this exercise, we will be referencing the Docker documentation to demonstrate immutable infrastructure. Please take a look at the Apply rolling updates to a service tutorial. In the presented example, Redis is upgraded from 3.0.6 to 3.0.7 - one container at a time. This could apply to any frontend or backend service, as long as the containers tag is updated and pushed to the registry.

Setup Swarm Backend Redis Service

1. Initialize a single-node swarm cluster:

   ```
   $ docker swarm init
   ```

   Note: By using a combination of VirtualBox and Swarm, it is possible to create a multi-node swarm to mimic a larger environment. More details can be found at https://www.linux.com/learn/how-use-docker-machine-create-swarm-cluster.

2. Create an overlay network to enable container-to-container networking:

   ```
   $ docker network create --driver overlay --attachable redis
   ```

3. Create a 3-replica `redis:3.0.6` swarm:


```
$ docker service create --replicas 3 --name redis --network redis
--update-delay 10s -p 6379:6379 redis:3.0.6
```

4. Verify that the swarm cluster has created 3/3 replicas. This process takes approximately 30 seconds. Also, while the replicas are starting, it may show 0/3.

```
$ docker service ls
ID            NAME   MODE        REPLICAS  IMAGE
o3w5rh3lzn4n  redis  replicated  3/3       redis:3.0.6
```

Add a Frontend Application

1. Clone `liatrio/docker-compose-demo` based on `vegasbrianc/docker-compose-demo`. It is a demo of the HAProxy load balancer with the ability to scale to n number of web servers via **`docker-compose`** with a Redis backend:

```
$ git clone https://github.com/liatrio/docker-compose-demo.git
```

2. Change the directory to our newly cloned repository:

```
$ cd docker-compose-demo
```

3. Run **`docker-compose`** to spin-up the HAProxy load balancer and a single **`web`** node:

```
$ docker-compose up -d
dockercomposedemo_web_1 is up-to-date
dockercomposedemo_lb_1 is up-to-date
```

4. Scale to 3 **`web`** nodes:

```
$ docker-compose scale web=3
Creating and starting dockercomposedemo_web_2 ... done
Creating and starting dockercomposedemo_web_3 ... done
```

5. Run the following command a few times to verify connectivity between the load balanced hosts:

```
$ curl localhost

Hello World!
I have been seen b'38' times.
My Host name is fbab6f87f815
```


Upgrade the Redis Backend

1. Update the Redis container image. This will trigger the rolling update to shut down one-by-one 3.0.6 containers and bring up 3.0.7:

```
$ docker service update --image redis:3.0.7 redis
```

2. Verify that the upgrade took place:

```
$ docker service ps redis
ID            NAME          IMAGE        NODE  DESIRED STATE  CURRENT STATE
ERROR  PORTS
t8chftzlei6l  redis.1       redis:3.0.7  moby  Ready          Ready 1 second
ago
wj7sp9c8p19t   \_ redis.1   redis:3.0.6  moby  Shutdown       Running 1 second
ago
4ohbybhtnzm1  redis.2       redis:3.0.7  moby  Running        Running 26
seconds ago
47qii16qhbus   \_ redis.2   redis:3.0.6  moby  Shutdown       Shutdown 27
seconds ago
vokqv2qbnn6m  redis.3       redis:3.0.7  moby  Running        Running 11
seconds ago
p3wamg4mt1l5   \_ redis.3   redis:3.0.6  moby  Shutdown       Shutdown 13
seconds ago
```

3. Verify that the application is still running:

```
$ curl localhost

Hello World!
I have been seen b'19' times.
My Host name is 23d29ce6305e
```

Summary

By introducing a combination of Swarm with other containers, we can simply point to the Redis network, which is a load balanced endpoint. An application using this Redis endpoint does not experience downtime as the older versions of the container are being shut down one-by-one, and a newer version is brought into the pool.

