4.4 Container Scaling with Docker Swarm

This section will guide you to**:**

* Scale Docker containers on Docker swarm.

This guide has two subsections, namely**:**

4.4.1 Creating service for scaling

4.4.2 Pushing the code to GitHub repositories

* *Docker is already installed in your lab. (Refer MEAN: Lab Guide - Phase 4)*

**Step 4.4.1:** Creating service for scaling

**Please Note:** Docker containers deployed on Docker swarm cluster can be scaled up and down to implement high availability of Docker containers. If in case any Docker container gets crashed, we can get a new one created and other containers can easily handle the load.

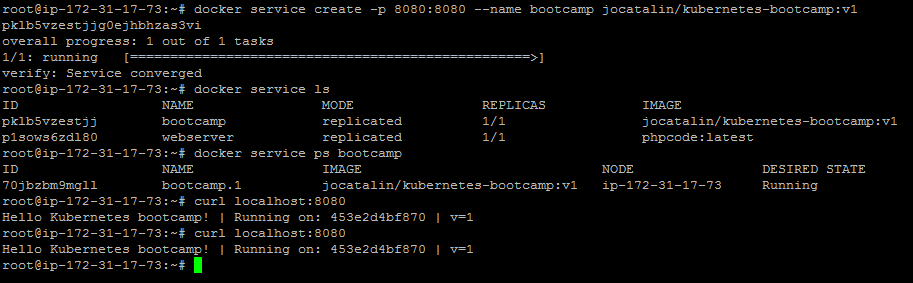
* Open the terminal
* Use the commands given below to create a service and scale the service up and down to increase or decrease Docker containers:

**docker service create -p 8080:8080 --name bootcamp jocatalin/kubernetes-bootcamp:v1**

**docker service ls**

**docker service ps bootcamp**

**curl localhost:8080**

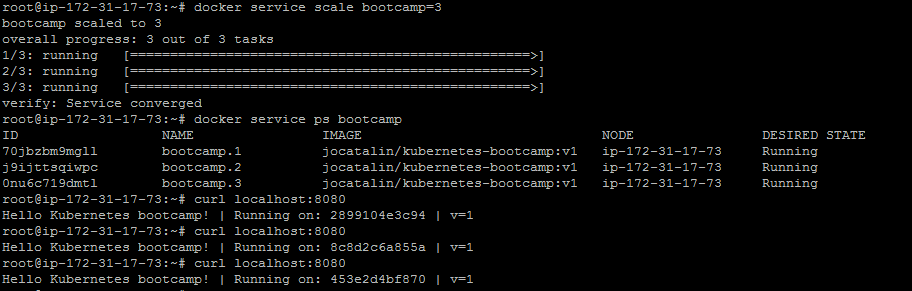


* Once the container deployed, we can scale up and down the Docker swarm service following the process shown below:

**docker service scale bootcamp=3**

**docker service ps bootcamp**

**curl localhost:8080**



**Please Note:** In the screenshot above, we can see that when we are trying to access swarm service on 8080 port, we are getting different IDs in the response. This means that our request is going to different containers in round-robin manner.

**Step 4.4.2:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master