4.5 Distribute Your App Across a Swarm Cluster

This section will guide you to**:**

* Deploy a Docker container on Docker swarm and have multiple nodes inside a cluster.

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

4.5.1 Setting up a Docker instance

4.5.2 Setting up a Docker swarm with multiple nodes

4.5.3 Deploying a custom Docker image to a Docker swarm cluster

4.5.4 Pushing the code to GitHub repositories

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

**Step 4.5.1:** Setting up a Docker instance

* Open the terminal
* Before proceeding with the Docker swarm cluster, install Docker on both **master** **server** and **worker node**. Please follow the set of commands below to install Docker on Ubuntu server.

**apt install docker.io**

**docker version**



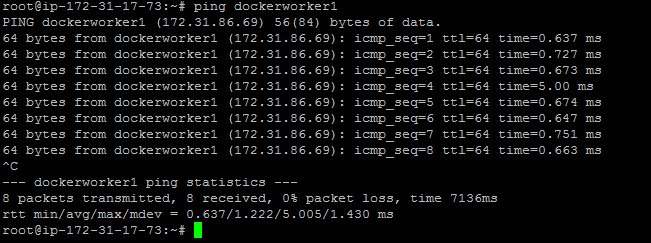
**Step 4.5.2:** Setting up Docker swarm with multiple nodes

* Edit the **/etc/hosts** file across the two nodes via **gedit** or **vim** and make the following changes:
* Type **sudo vim /etc/hosts**

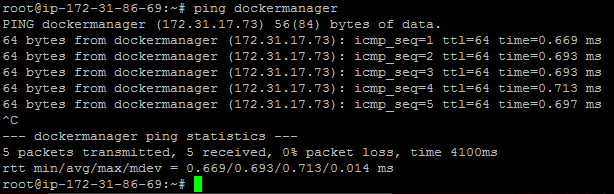
172.31.17.73 dockermanager

172.31.86.69 dockerworker1

* After modifying the host file with the details mentioned above, check the connectivity with **ping** between all the nodes using **ping dockerworker1** command



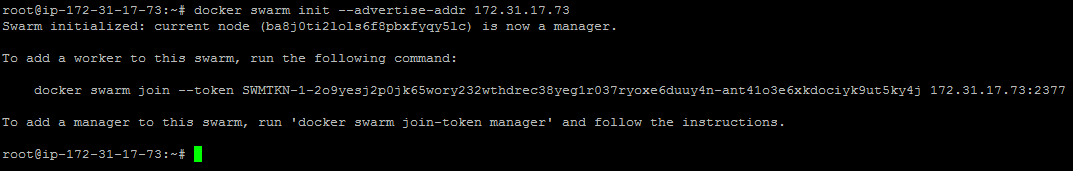
* From Docker Worker Node instance: Type **ping dockermanager**



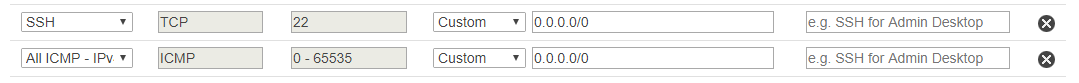
* Initialize the Docker swarm mode by running the following docker command on the **dockermanager** node:

docker swarm init --advertise-addr<manager node IP address>

docker swarm init --advertise-addr172.31.17.73



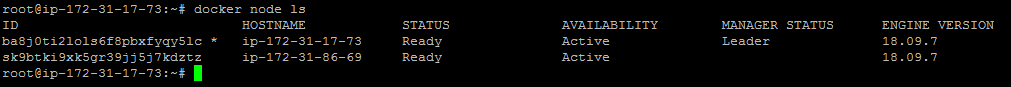
* Once the swarm cluster is initialized, allow the ports mentioned below in security groups:





* While initializing the Docker swarm cluster, you will get docker swarm join command which can be executed on node manager to add node to swarm cluster
* Run the command below to see the node status:

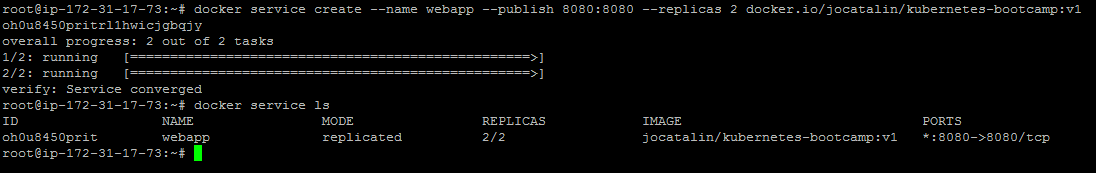
**docker node ls**



**Step 4.5.3:** Deploying a custom Docker image to a Docker swarm cluster

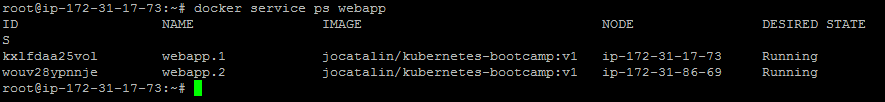
* Create service in Docker swarm cluster

**docker service create --name webapp --publish 8080:8080 --replicas 2 docker.io/jocatalin/kubernetes-bootcamp:v1**

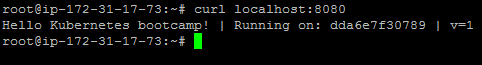


* You can now validate if Docker containers got deployed on both nodes or not using the command below:

**docker service ps webapp**



**Please Note:** We can validate the application using the **curl localhost:8080** command to see if the application is up and running.



**Step 4.5.4:** 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