Nicola Santillo 24.12.2023

Application Deployment with Docker Swarm mode:

Configurations:

To install compose on Ubuntu:
https://docs.docker.com/compose/install/linux/?fbclid=IwAR29439VC9eNuQr2RQI0hQj1YcNwa6hCDyyzp1EwU1UbAvDornNE7r4EURk

To update Ubuntu:

sudo apt update

sudo apt upgrade

To change the name of the host:

- sudo nano /etc/hostname
- sudo nano /etc/hosts
- sudo reboot

Deploy a stack to a swarm

Create the swarm on the manager node (vm with this IP is manager)

docker swarm init --advertise-addr 192.168.56.103

Run this command on the worker nodes to join the swarm

docker swarm join --token SWMTKN-1-02u5fw38drn8eoc0mwx7rni52ewmw7eil942rvqkxq4bdydni9-6ozcc8152q3c7q3h137fs6pmn 192.168.1.109:2377

To view the list of nodes in the swarm

docker node ls

To inspect an individual node from the manager node

docker node inspect <node-id> self --pretty

To distribute images to all the nodes, create a service registry on the manager node

docker service create --name registry --publish published=5000,target=5000 registry:2

To confirm that the service is created correctly

docker service ls

Start the app with docker compose up. This builds the web app image and creates containers.

docker compose up -d

Check the app is running

docker compose ps

To distribute the web app's image across the swarm pushing it to the registry

docker compose push

To deploy the specified services, creating a set of interconnected containers across multiple nodes in a Docker Swarm cluster (llamawebapp is the name of this docker stack)

docker stack deploy --compose-file compose.yml llamawebapp

To inspect the service

docker service inspect --pretty llamawebapp_llama_webapp

If you want to remove the stack:

docker stack rm llamawebapp

docker service rm registry

docker swarm leave (from each worker node).

docker swarm leave –force (from master nodes)

LOADBALANCING:

Deploy the Nginx service on the master node

docker service create --name backend --replicas 2 --publish 8080:80 nginx

Initialize a single Swarm cluster on the load balancer node

docker swarm init --advertise-addr 192.168.1.17

To create an Nginx container and allow connections to the web services hosted by your Docker Swarm:

sudo mkdir -p /data/loadbalancer

sudo chmod 777 /data/loadbalancer

Create file default.conf in /data/loadbalancer with the IP Addresses of manager and worker nodes

```
server {
   listen 80;
   location / {
      proxy_pass http://backend;
   }
}
upstream backend {
   server 192.168.1.109:7860;
   server 192.168.1.104:7860;
   server 192.168.1.84:7860;
}
```

docker service create --name loadbalancer --mount type=bind,source=/data/loadbalancer,target=/etc/nginx/conf.d --publish 80:80 nginx

Useful instructions:

docker build -t llama-webapp-img .

docker run --name llama-webapp-cont -p 7860:7860 llama-webapp-img

docker image rm [OPTIONS] IMAGE [IMAGE...]

docker node ls

docker service ps llamawebapp_llama_webapp

docker node demote <node-id> (master to worker)

docker swarm join-token worker

References:

https://docs.docker.com/engine/swarm/

https://cloudinfrastructureservices.co.uk/how-to-setup-docker-swarm-load-balancing-using-nginx-on-ubuntu-20-04/?fbclid=IwAR2OeK6jajs5RLQaiIKTV5ZoBi63se-woIslcQ54zNjADHGIwiPMB85Sm5k

https://upcloud.com/resources/tutorials/load-balancing-docker-swarm-mode?fbclid=IwAR0uP8EDWXPPEiOprUAs1Z5hCp4AYfn_NNElLSstRxqRiSaPMbCivYzNhlU