Working with Docker Swarm

Create a swarm

```
$ docker swarm init --advertise-addr 192.168.99.100
Swarm initialized: current node (dxn1zf6l61qsb1josjja83ngz) is now a manager.
    docker swarm join \
    --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \
    192.168.99.100:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

The --advertise-addr flag configures the manager node to publish its address as 192.168.99.100. The other nodes in the swarm must be able to access the manager at the IP address.

The output includes the commands to join new nodes to the swarm. Nodes will join as managers or workers depending on the value for the --token flag.

To add a worker to this swarm, run the below command on the other Docker Hosts:

```
docker swarm join \
    --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \
    192.168.99.100:2377
```

Run docker info to view the current state of the swarm:

```
$ docker info
Containers: 2
Running: 0
Paused: 0
Stopped: 2
  ...snip...
Swarm: active
  NodeID: dxn1zf6l61qsb1josjja83ngz
```

```
 Is Manager: true
 Managers: 1
 Nodes: 1
 ...snip...
```

Run the docker node ls command to view information about nodes:

```
$ docker node ls
ID                          HOSTNAME STATUS AVAILABILITY MANAGER STATUS
dxn1zf6l61qsb1josjja83ngz *   manager1   Ready   Active        Leader
```

The * next to the node ID indicates that you’re currently connected on this node.

After you create a swarm, you can deploy a service to the swarm. For this tutorial, you also added worker nodes, but that is not a requirement to deploy a service.

Run the following command:

```
$ docker service create --replicas 10 --name helloworld nginx
```

The docker service create command creates the service.

The --name flag names the service helloworld.

The --replicas flag specifies the desired state of 10 running instance.

Run docker service ls to see the list of running services:

```
$ docker service ls
ID            NAME        SCALE IMAGE COMMAND
9uk4639qpg7n helloworld 1/1 alpine ping docker.com
```

Inspect the service :

```
$ docker service inspect  servicename
```

Scale the service in the swarm

Once you have deployed a service to a swarm, you are ready to use the Docker CLI to scale the number of containers in the service. Containers running in a service are called “tasks.”

Run the following command to change the desired state of the service running in the swarm:

```
$ docker service scale <SERVICE-ID>=<NUMBER-OF-TASKS>
```

For example:

```
$ docker service scale helloworld=5
```

```
helloworld scaled to 5
```

Run docker service ps <SERVICE-ID> to see the updated task list:

```
$ docker service ps helloworld
```

Run docker ps to see the containers running on the node where you’re connected. The following example shows the tasks running on manager1:

```
$ docker ps
```