Lab Swarm Mode Introduction

Time Spent: 45 min

For real applications IT users and app teams need more sophisticated tools. Docker supplies two such tools: ***Docker Compose*** and ***Docker Swarm Mode***. The two tools have some similarities but some important differences:

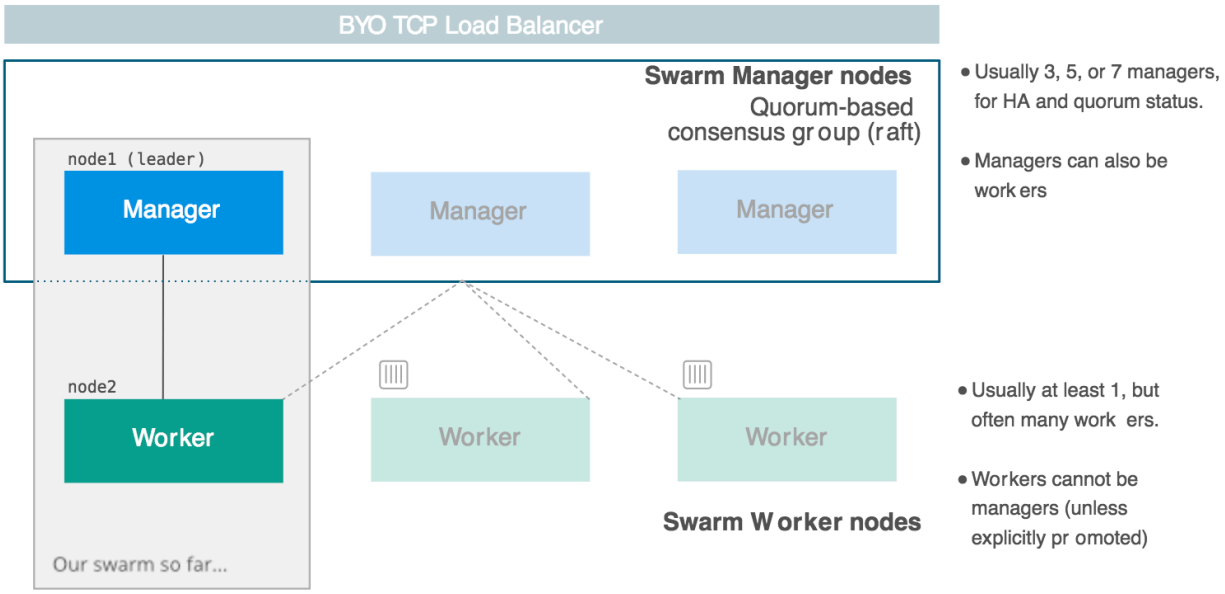
- **Compose** is used to control multiple containers on a single system. Much like the *Dockerfile* we looked at to build an image, there is a text file that describes the application: which images to use, how many instances, the network connections, etc. But *Compose* only runs on a single system so while it is useful, we are going to skip *Compose*[1] and go straight to *Docker Swarm Mode*.
- **Swarm Mode** tells Docker that you will be running many Docker engines and you want to coordinate operations across all of them. Swarm mode combines the ability to not only define the application architecture, like Compose, but to define and maintain high availability levels, scaling, load balancing, and more. With all this functionality, *Swarm mode* is used more often in production environments than it's more simplistic cousin, Compose.

1. Initialize Your Swarm

```
$ docker swarm init --advertise-addr $(hostname -i)
Swarm initialized: current node (tklw4u4ev55fsemaxhepuo5d3) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-29ioghok2f3scf0eubgw7z2x984mz67gv3h7vx7ll2omk06u
8-bswq1f5lxm41oqptgqezo1ohq 192.168.0.22:2377

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

2. Show Swarm Members

Docker Enterprise Edition: Swarm Architectur e



3. Deploy a Stack

```
[node1] (local) root@192.168.0.22 ~/example-voting-app
$ git clone https://github.com/docker/example-voting-app
Cloning into 'example-voting-app'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 777 (delta 4), reused 0 (delta 0), pack-reused 763
Receiving objects: 100% (777/777), 856.39 KiB | 16.16 MiB/s, done.
Resolving deltas: 100% (272/272), done.
[node1] (local) root@192.168.0.22 ~/example-voting-app
$ cd example-voting-app
[node1] (local) root@192.168.0.22 ~/example-voting-app/example-voting-app
$ cat docker-stack.yml
version: "3"
services:
```

```
[node1] (local) root@192.168.0.22 ~/example-voting-app/example-voting-app
$ docker stack deploy --compose-file=docker-stack.yml voting_stack
Creating network voting_stack_frontend
Creating network voting_stack_backend
Creating network voting_stack_default
Creating service voting_stack_db
Creating service voting_stack_vote
Creating service voting_stack_result
Creating service voting_stack_worker
```