How to do it

Follow these steps:

1. Log in to `dockerhost-1` and initialize the Swarm. The `--advertise-addr` argument is the IP address for your host where you will be listening for Swarm traffic:

```
$ docker swarm init --advertise-addr <your host ip>

Swarm initialized: current node (4daiatuoef7eh0ne6kawtflml) is no
To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-2nyaeu0l2rw7fv6wpgco4o1spos0el

To add a manager to this swarm, run 'docker swarm join-token mana
```

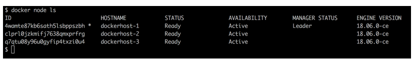
The Swarm token that is returned by the Swarm `init` command will be different for everyone. Make sure you use the one that you get when you start your Swarm.

2. Login to `dockerhost-2` and `dockerhost-3` and join the Swarm as a worker:

```
$ docker swarm join --token SWMTKN-1-2nyaeu0l2rw7fv6wpgco4o1spos0
```

3. On `host1`, list the nodes in the Swarm cluster as shown in the following screenshot:

```
$ docker node ls
```



```
$ docker node ls
ID                            HOSTNAME       STATUS   AVAILABILITY   MANAGER STATUS   ENGINE VERSION
4wamte87kb6sath5lsbppszbh *   dockerhost-1   Ready    Active         Leader           18.06.0-ce
clprl0jzkmifj7638qmxprfrg     dockerhost-2   Ready    Active                          18.06.0-ce
q7qtu08y96u0gyfip4txzi0u4     dockerhost-3   Ready    Active                          18.06.0-ce
$
```

As you can see, we have three nodes in the cluster: one manager and two workers. Now that you have a Swarm, you can schedule tasks on the Swarm.

4. Start a service on the cluster:

```
$ docker service create --name demo --publish 80:80 nginx
```

5. Look at the status of the service by referring to the follwoing screenshot:

```
$ docker service ls
$ docker service ps demo
```

```
$ docker service ls
ID             NAME     MODE         REPLICAS   IMAGE          PORTS
y4zw1uhhx1fj   demo     replicated   1/1        nginx:latest   *:80->80/tcp
$ docker service ps demo
ID             NAME     IMAGE          NODE           DESIRED STATE   CURRENT STATE            ERROR   PORTS
i7m3njcsbvfr   demo.1   nginx:latest   dockerhost-2   Running         Running 20 seconds ago
$
```

6. Scale the service to 3:

```
$ docker service scale demo=3
```

7. Check to make sure that the service has scaled to `3`. It might take a few minutes for the scaling event to complete:

```
$ docker service ls
ID             NAME     MODE         REPLICAS   IMAGE          PORTS
y4zw1uhhx1fj   demo     replicated   3/3        nginx:latest   *:80->80/tcp
$ docker service ps demo
ID             NAME     IMAGE          NODE           DESIRED STATE   CURRENT STATE            ERROR   PORTS
i7m3njcsbvfr   demo.1   nginx:latest   dockerhost-2   Running         Running 2 minutes ago
ejh41vpts1f0   demo.2   nginx:latest   dockerhost-1   Running         Running 10 seconds ago
696umwvgyvop   demo.3   nginx:latest   dockerhost-3   Running         Running 10 seconds ago
$
```

PREV
Getting ready

NEXT
How it works…