

Docker Swarm on CentOS7 VMs.

1. Setup a swarm of three Vms
2. Create a service of 2 replicas
3. Scale the service to 3 replicas
4. Clean up

Three VMs all running the docker engine.

lin70: 192.168.4.240 (manager)

lin71: 192.168.4.241 (worker)

lin72: 192.168.4.243 (worker)

1. Setup a swarm of three VMs

```
lin70 #docker swarm init --advertise-addr 192.168.4.240
Swarm initialized: current node (dxnlzf6l6lqsbljosjja83ngz) is now a
manager.
```

This will setup the Manager that will advertise itself with address 192.168.4.240.

Join the two other Vms by copy paste of the swarm init output.

```
lin71 #docker swarm join \
  --token SWMTKN-1-
49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-
8vxv8rssmk743ojnwacrr2e7c \
  192.168.4.240:2377

lin72 #docker swarm join \
  --token SWMTKN-1-
49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-
8vxv8rssmk743ojnwacrr2e7c \
  192.168.4.240:2377
```

2. Create the service to run on 2 replica's

```
lin70 # docker service create --replicas 2 --name helloworld alpine
ping docker.com
9uk4639qpg7npwf3fn2aasksr
```

List the containers helloworld.1 and helloworld.2

```
lin70 #docker service ps helloworld
ID                NAME          IMAGE          NODE   DESIRED STATE
CURRENT STATE    ERROR        PORTS
7oehvu04hxa4    helloworld.1  alpine:latest  lin71  Running
Running 5 hours ago
3s93yfidbwq0    helloworld.2  alpine:latest  lin70  Running
Running 5 hours ago
```

3. Scale the service to 3 replica's

```
lin70 #docker service scale helloworld=3
helloworld scaled to 3
```

Run "docker ps -a" on all three Vms to check the helloworld containers.

4. Clean up.

```
lin70 #docker service rm helloworld
(service is stopped)
lin70 #docker ps -a
(no helloworld container)
lin71 #docker ps -a
(no helloworld container)
lin72 #docker ps -a
(no helloworld container)
```

```
lin72 #docker swarm leave
lin71 #docker swarm leave
lin70 #docker swarm leave --force
```