

#creacion del cluster

```
$ docker swarm init --advertise-addr 192.168.3.80
```

Añadiremos ahora un nodo Worker al clúster, Para ello y desde la consola del Worker ejecutamos:

```
$ docker swarm join --token SWMTKN-1-50edlk935u9qgvrs8alhpzf1awgdil2dmfs4zgpd8ue2ltkmww-3y9tb0pnieqao9ahkutpvppe 192.168.3.80:2377
```

This node joined a swarm as a worker.

Para dar de alta un nuevo servicio nos vamos a la consola de un Manager y ejecutamos lo siguiente:

```
$ docker service create --replicas 1 --name helloworld alpine ping enmilocalfunciona.io
```

Ahora podemos ver un listado de todos los servicios en el clúster:

```
$ docker service ls
```

Podemos ver información sobre el servicio:

```
$ docker service inspect --pretty helloworld
```

Podemos ver todas las tareas que se están ejecutando para este servicio:

```
$ docker service ps helloworld
```

Si queremos aumentar o disminuir el número de réplicas ejecutamos:

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

Y después comprobamos en qué nodos están corriendo las nuevas réplicas:

```
$ docker service ps helloworld
```

EJEMPLO CON VAGRANT

```
$ vagrant up
```

#Con este comando establecemos una conexión SSH con el nodo Manager

```
$ vagrant ssh manager
```

#Podemos ver todos los nodos:

```
$ docker node ls
```

#Crear un servicio nuevo:

```
$ docker service create --replicas 10 --name helloworld -p 8080:8080 drhelius/helloworld-node-microservice
```

```
ewa6g6c3vg2kbqzl5uqvi3pli
```

#Y ver sus tareas:

```
$ docker service ps helloworld
```

#Podemos probarlo apuntando al Manager y utilizando el puerto que expusimos en su creación, 8080:

```
ubuntu@manager:~$ curl localhost:8080
```

#Para probarlo, terminamos la sesión SSH con el Manager, entramos en el worker03 y lo apagamos:

```
$ exit
```

```
$ vagrant ssh worker03
```

```
$ sudo shutdown -h now
```

#Entramos de nuevo en el Manager y vemos qué ha pasado:

```
$ docker service ps helloworld
```