

# Práctica 6

## CPD

Por: Rubén Calvo Villazán

## Creación de las máquinas:

```

root@valkyrie:P6# docker-machine create m1
Creating CA: /root/.docker/machine/certs/ca.pem
Creating client certificate: /root/.docker/machine/certs/cert.pem
Running pre-create checks...
(m1) Image cache directory does not exist, creating it at /root/.docker/machine/cache...
(m1) No default Boot2Docker ISO found locally, downloading the latest release...
(m1) Latest release for github.com/boot2docker/boot2docker is v18.06.1-ce
(m1) Downloading /root/.docker/machine/cache/boot2docker.iso from https://github.com/boot2docker/boot2docker/releases/download/v18.06.1-ce/boot2docker.iso...
(m1) 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Creating machine...
(m1) Copying /root/.docker/machine/cache/boot2docker.iso to /root/.docker/machine/machines/m1/boot2docker.iso...
(m1) Creating VirtualBox VM...
(m1) Creating SSH key...
(m1) Starting the VM...
(m1) Check network to re-create if needed...
(m1) Found a new host-only adapter: "vboxnet2"
(m1) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env m1
root@valkyrie:P6#

```

```
root@valkyrie:P6# docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
m1	-	virtualbox	Running	tcp://192.168.99.100:2376		v18.06.1-ce	

```
root@valkyrie:P6#
```

IP de la máquina m1.

```
root@valkyrie:~# docker-machine ip m1
192.168.99.100
root@valkyrie:~#
```

Nos conectamos a m1.

[illegible]

## Inicio manager Docker Swarm

En m2 y m3 debemos introducir el comando **docker swarm join** para que se unan a la red.

```
docker@m1:~$ docker swarm init --advertise-addr 192.168.99.100
Swarm initialized: current node (zxavqc25lzw3b0aqoc9l2kpi9) is now a manager.

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

    docker swarm join --token SWMTKN-1-5javvbg14wcwiwj7fj3x67v9mfwasygo4089o0iqbudkvyavmv-4y2lqf3a6h15e6aytc0lotb6i 192.168.99.100:2377

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

Nodo m1 activo.

```
docker@m1:~$ docker node ls
ID                                HOSTNAME          STATUS          AVAILABILITY          MANAGER STATUS          ENGINE VERSION
zxavqc25lzw3b0aqoc9l2kpi9 *      m1                Ready           Active                 Leader                   18.06.1-ce
```

Nos conectamos a m3 y m2.

```
root@valkyrie:~# docker-machine env m3
root@valkyrie:~# docker-machine ssh m3
##
## ## ##
## ## ## ##
#####
o
#####
Docker version 18.06.1-ce, build HEAD : c7e5c3e - Wed Aug 22 16:27:42 UTC 2018
Docker version 18.06.1-ce, build e68fc7a
docker@m3:~$ docker swarm join --token SWMTKN-1-5javvbg14wcwiwj7fj3x67v9mfwasygo4089o0iqbudkvyavmv-4y2lqf3a6h15e6aytc0lotb6i 192.168.99.100:2377
This node joined a swarm as a worker.
docker@m3:~$

root@valkyrie:~# docker-machine ssh m2
##
## ## ##
## ## ## ##
#####
o
#####
Docker version 18.06.1-ce, build HEAD : c7e5c3e - Wed Aug 22 16:27:42 UTC 2018
Docker version 18.06.1-ce, build e68fc7a
docker@m2:~$ docker swarm join --token SWMTKN-1-5javvbg14wcwiwj7fj3x67v9mfwasygo4089o0iqbudkvyavmv-4y2lqf3a6h15e6aytc0lotb6i 192.168.99.100:2377
This node joined a swarm as a worker.
docker@m2:~$
```

Creamos el servicio web.

```
docker@m1:~$ docker service create --name web --replicas 3 --mount type=bind,src=/etc/hostname,dst=/usr/share/nginx/html/index.html,readonly --publish published=8080,target=80 nginx
nginx
4fd4y3ljey3sh7x3y8gvzcvdw
overall progress: 3 out of 3 tasks
1/3: running
2/3: running
3/3: running
verify: Service converged
```

## Prueba de funcionamiento:

Reparte el trabajo entre m1, m2 y m3.

```
root@valkyrie:P6# curl http://192.168.99.102:8080
m2
root@valkyrie:P6# curl http://192.168.99.102:8080
m3
root@valkyrie:P6# curl http://192.168.99.102:8080
m1
root@valkyrie:P6#
```

Las 3 están activas.

```
docker@m1:~$ docker service ps web
ID                NAME          IMAGE          NODE          DESIRED STATE   CURRENT STATE           ERROR          PORTS
8xo8btknrkcx     web.1         nginx:latest   m1            Running         Running about a minute ago
cpi726clvm3v     web.2         nginx:latest   m2            Running         Running about a minute ago
vfllpiobjreo     web.3         nginx:latest   m3            Running         Running about a minute ago
docker@m1:~$
```

## Cambiamos la escala a 2.

```
docker@m1:~$ docker service scale web=2
web scaled to 2
overall progress: 2 out of 2 tasks
1/2: running
2/2: running
verify: Service converged
docker@m1:~$
```

Paramos m3.

```
root@valkyrie:P6# docker-machine stop m3
Stopping "m3"...
Machine "m3" was stopped.
root@valkyrie:P6#
```

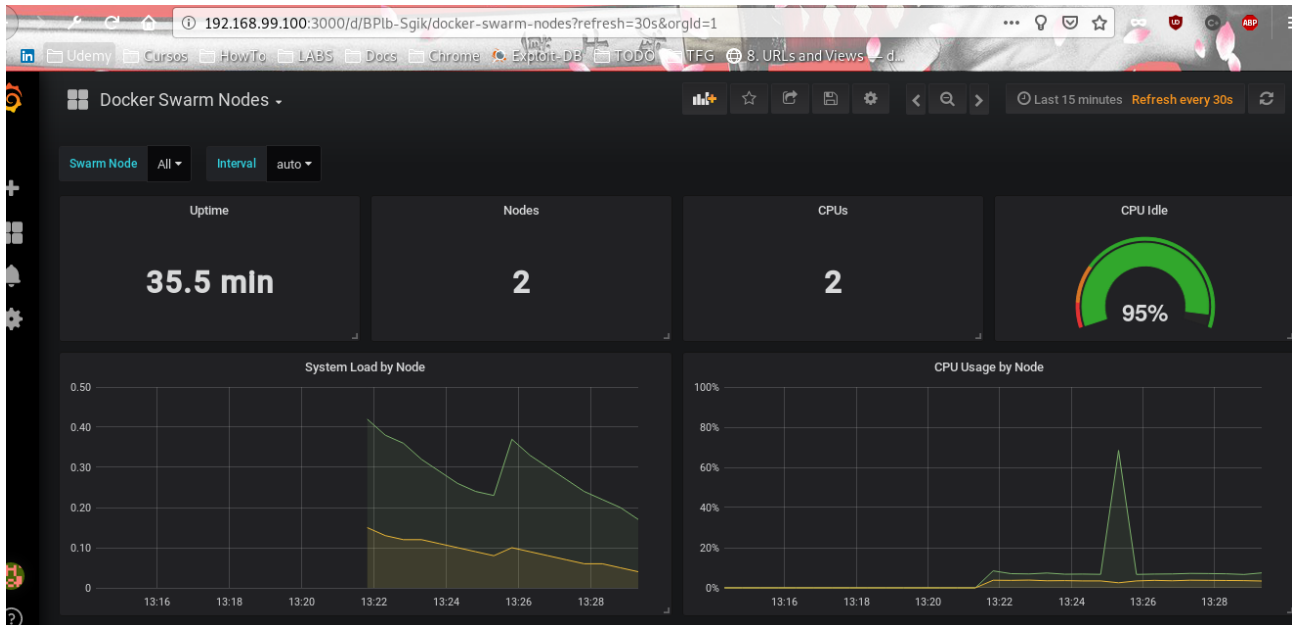
Al hacer curl, solo reparte el trabajo entre m1 y m2.

```
root@valkyrie:P6# curl http://192.168.99.100:8080
m2
root@valkyrie:P6# curl http://192.168.99.100:8080
m1
root@valkyrie:P6# curl http://192.168.99.100:8080
m2
root@valkyrie:P6# curl http://192.168.99.100:8080
m1
root@valkyrie:P6# curl http://192.168.99.100:8080
m2
root@valkyrie:P6#
```

Si volvemos a levantar m3, se volvería a repartir el trabajo entre las tres máquinas.

## Estadísticas modificando la cpu.

Entrar a 192.168.99.100:3000



Comando para cambiar la cpu asignada.

```
docker service create --name web2 --replicas 3 --mount type=bind,src=/etc/hostname,dst=/usr/share/nginx/html/index.html,readonly  
--publish published=8081,target=80 nginx --cpu-limit=0.5
```

## Capturas de la plataforma Katacoda. (<https://www.katacoda.com/>)

Docker & Containers Scenario.

The screenshot shows the 'Get Started!' section of the Katacoda platform. It includes a progress bar and two scenario cards.

**Get Started!**

- Scenarios Completed: 1 of 21
- Progress: 5%
- Points: 10

[Create Your Free Account](#)

**Deploying Your First Docker Container**

Learn how to launch containers using Docker

[Repeat Scenario](#)

**Deploy Static HTML Website as Container**

Learn how to run a static HTML website using Nginx

[Start Scenario](#)

## Docker & Containers Scenario.

The screenshot shows the Katacoda interface for a scenario titled "Deploy Static HTML Website as Container". The interface is divided into several sections:

- Header:** Katacoda logo and navigation links (Katacoda Overview & Solutions, Search, Log In, Sign Up).
- Scenario Title:** "Deploy Static HTML Website as Container".
- Step Indicator:** "Step 1 of 3".
- Description:** "Nginx. This provides the configured web server on the Linux Alpine distribution."
- Task:** "Create your *Dockerfile* for building your image by copying the contents below into the editor."
- Code Snippet:**

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

A "Copy to Editor" button is next to the code.
- Explanation:** "The first line defines our base image. The second line copies the content of the current directory into a particular location inside the container."
- Buttons:** A "CONTINUE" button is at the bottom right.
- Editor:** A code editor showing the `Dockerfile` with the same content as the snippet. The file explorer on the left shows `Dockerfile` and `index.html`.
- Terminal:** A terminal window at the bottom right, labeled "docker:80", is currently empty.

## Docker Security Scenario.

The screenshot shows the "Get Started!" section of the Docker Security Scenario. It features a progress bar and a list of scenarios:

- Get Started!**
- Progress:** 1 of 12 Scenarios Completed, 9% Progress, 10 Points.
- Scenarios:**
  - Hack Elasticsearch container:** Learn how to use application exploits to gain access to a container. (Start Scenario)
  - CGroups and Namespaces:** Learn how cgroups and namespaces limit what a container can do. (Start Scenario)
  - Scan Images for:** (Create Your Free Account)
  - Introduction to:** (Create Your Free Account)
  - Use No New:** (Create Your Free Account)
  - User:** (Create Your Free Account)