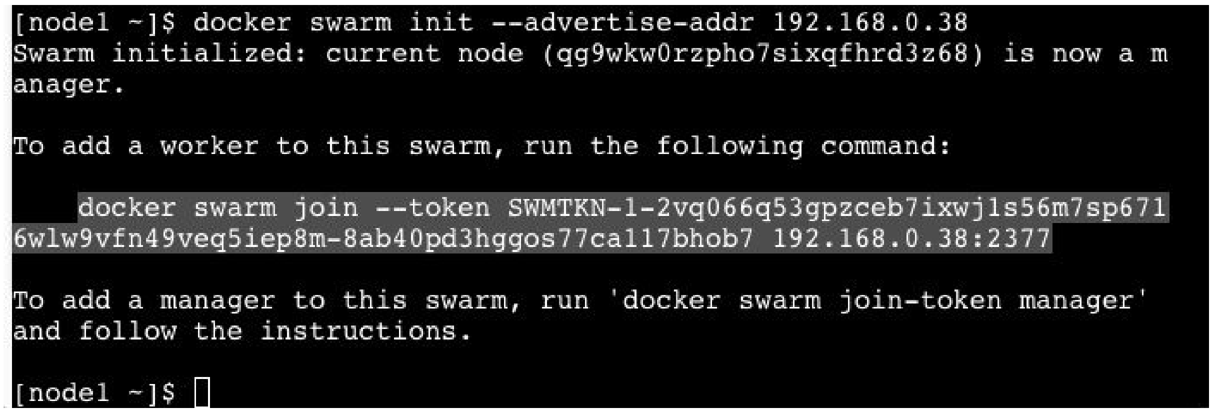
CPD Práctica 3

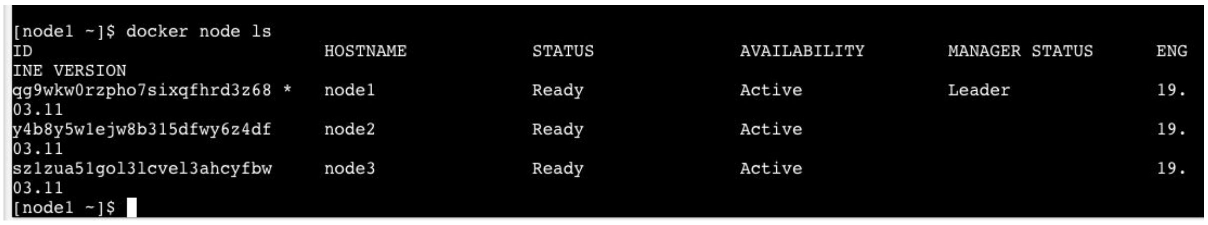
Jesús Manuel Pérez Terrón

1. Creación de las máquinas virtuales con docker-machine o bien el acceso al laboratorio remoto e inicio del manager de docker swarm.



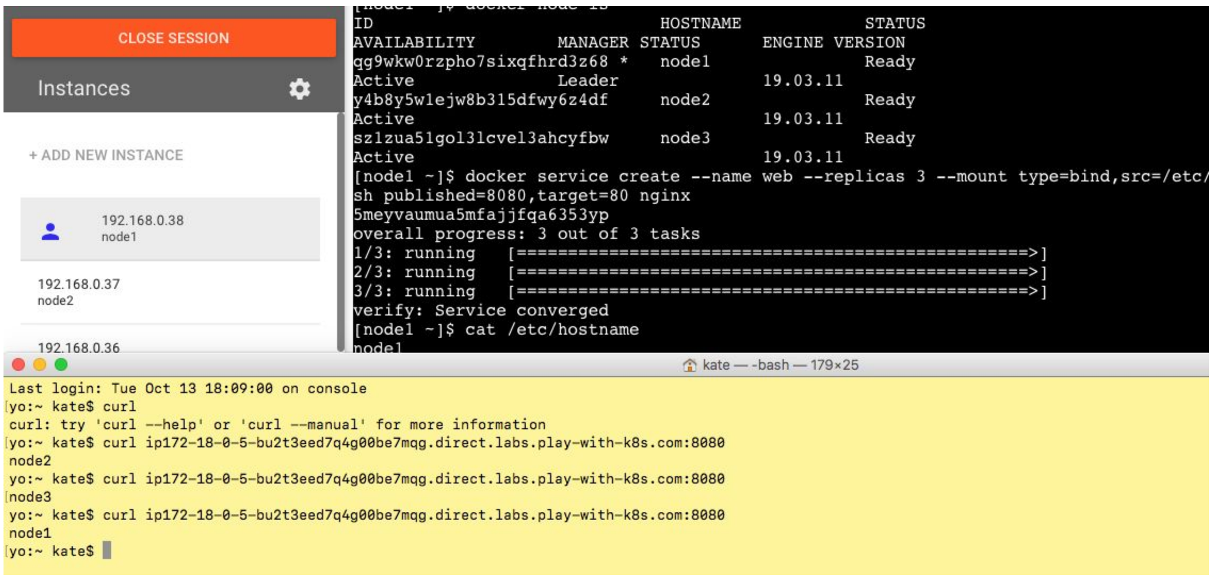
```
[node1 ~]$ docker swarm init --advertise-addr 192.168.0.38
Swarm initialized: current node (qg9wkw0rzpho7sixqfhrd3z68) is now a m
anager.

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

    docker swarm join --token SWMTKN-1-2vq066q53gpzceb7ixwj1s56m7sp671
6w1w9vfn49veq5iep8m-8ab40pd3hggos77ca117bhob7 192.168.0.38:2377

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

[node1 ~]$
```

2. Ejecución del servicio web y 3 nodos activos.



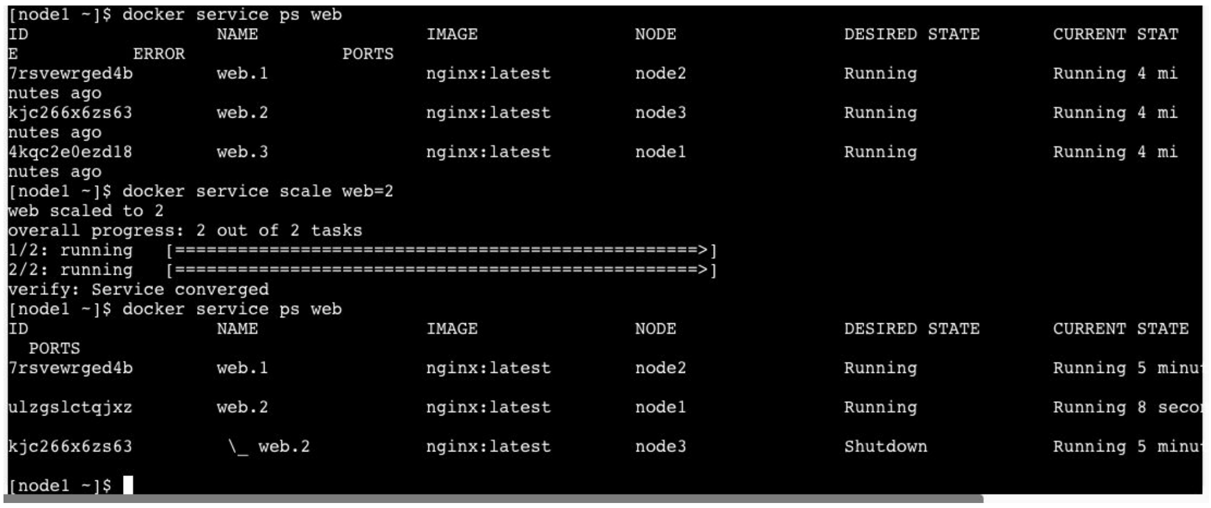
```
[node1 ~]$ docker node ls
ID                          HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS      ENG
INE VERSION
qg9wkw0rzpho7sixqfhrd3z68 *   node1               Ready               Active              Leader              19.
03.11
y4b8y5w1ejw8b315dfwy6z4df     node2               Ready               Active                                  19.
03.11
sz1zua51gol3lcvel3ahcyfbw     node3               Ready               Active                                  19.
03.11
[node1 ~]$
```

3. Cuando se cambia de escala a 2.



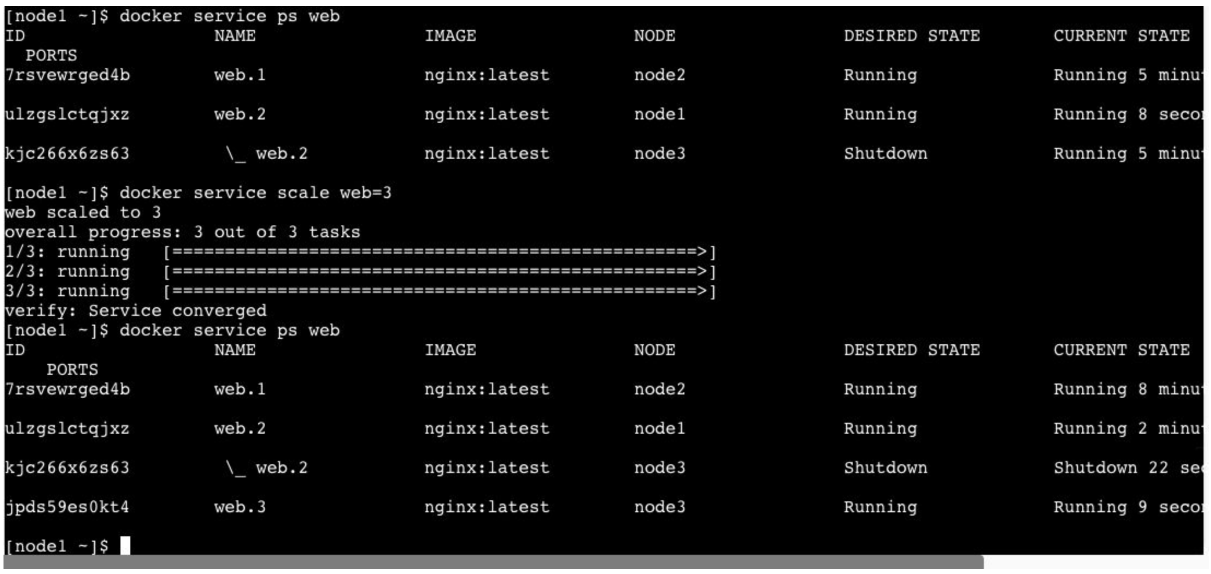
```
ID                          HOSTNAME            STATUS
AVAILABILITY        MANAGER STATUS      ENGINE VERSION
qg9wkw0rzpho7sixqfhrd3z68 *   node1               Ready
Active              Leader              19.03.11
y4b8y5w1ejw8b315dfwy6z4df     node2               Ready
Active                                  19.03.11
sz1zua51gol3lcvel3ahcyfbw     node3               Ready
Active                                  19.03.11
[node1 ~]$ docker service create --name web --replicas 3 --mount type=bind,src=/etc/
sh published=8080,target=80 nginx
5meyvaumua5mfajjfqa6353yp
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
[node1 ~]$ cat /etc/hostname
node1
```

```
Last login: Tue Oct 13 18:09:00 on console
yo:~ kate$ curl
curl: try 'curl --help' or 'curl --manual' for more information
yo:~ kate$ curl ip172-18-0-5-bu2t3eed7q4g00be7mqg.direct.labs.play-with-k8s.com:8080
node2
yo:~ kate$ curl ip172-18-0-5-bu2t3eed7q4g00be7mqg.direct.labs.play-with-k8s.com:8080
node3
yo:~ kate$ curl ip172-18-0-5-bu2t3eed7q4g00be7mqg.direct.labs.play-with-k8s.com:8080
node1
yo:~ kate$
```



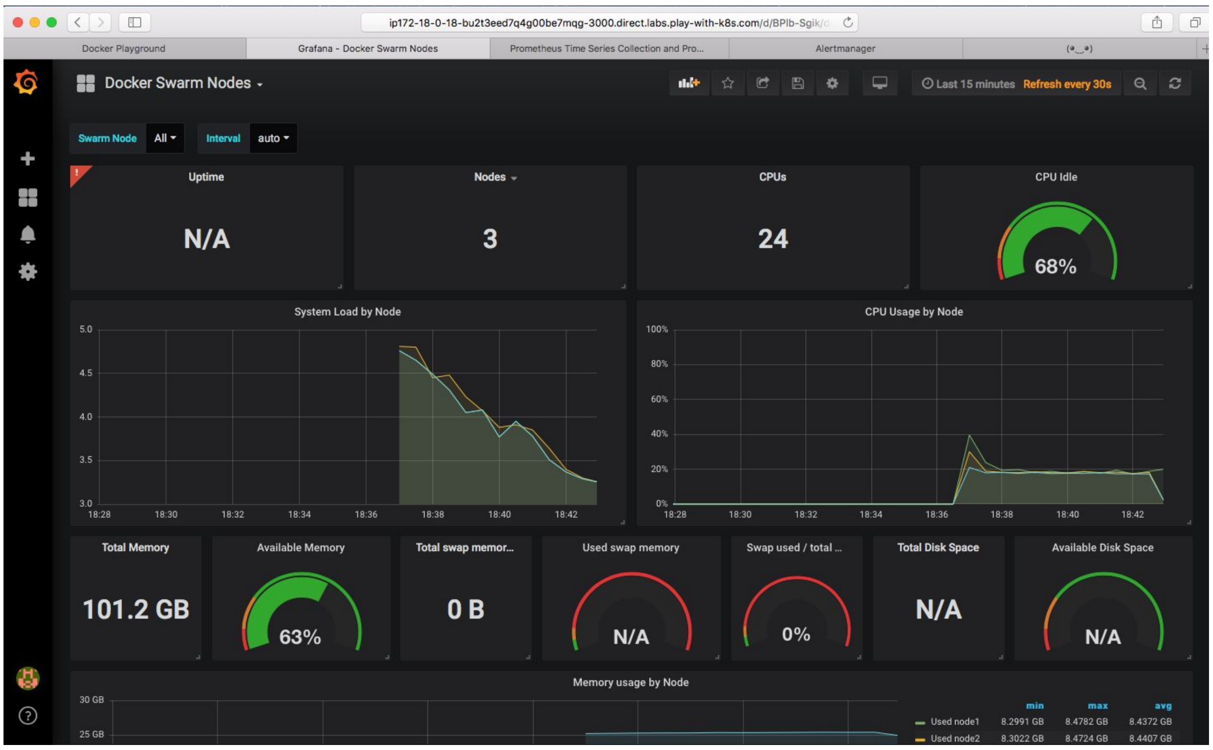
```
[node1 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE                DESIRED STATE       CURRENT STAT
E           ERROR               PORTS
7rsvewrged4b        web.1               nginx:latest        node2               Running             Running 4 mi
nutes ago
kjc266x6zs63        web.2               nginx:latest        node3               Running             Running 4 mi
nutes ago
4kqc2e0ezd18        web.3               nginx:latest        node1               Running             Running 4 mi
nutes ago
[node1 ~]$ 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
[node1 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE                DESIRED STATE       CURRENT STATE
  PORTS
7rsvewrged4b        web.1               nginx:latest        node2               Running             Running 5 minu
                                                                                                    
ulzgslctqjxz        web.2               nginx:latest        node1               Running             Running 8 seco

kjc266x6zs63         \_ web.2           nginx:latest        node3               Shutdown            Running 5 minu

[node1 ~]$ 
```

4. Reactivación automática del nodo.

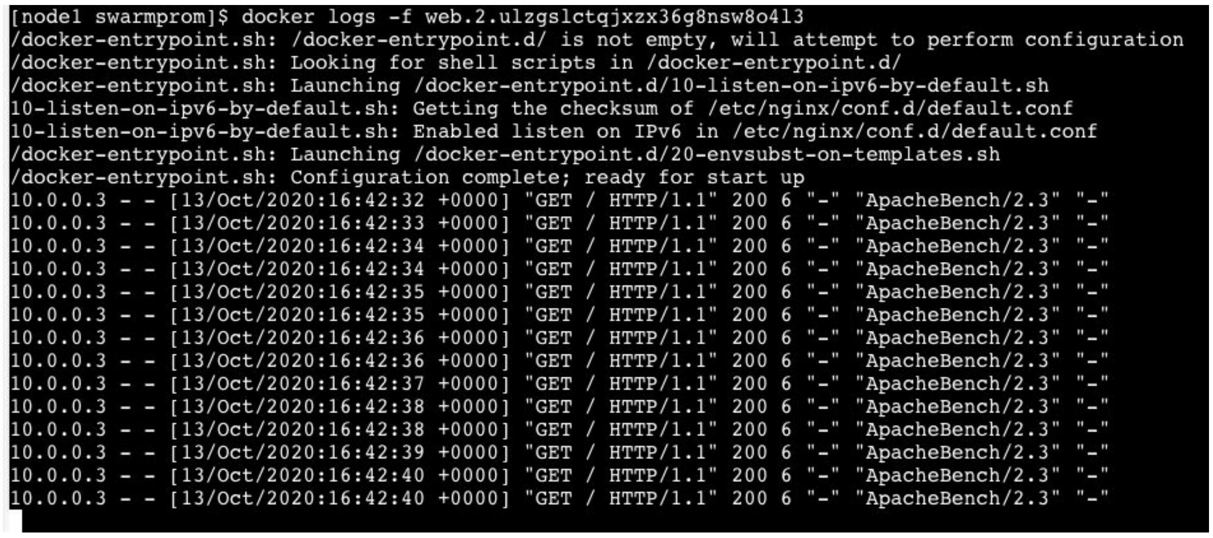


```
[node1 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE                DESIRED STATE       CURRENT STATE
  PORTS
7rsvewrged4b        web.1               nginx:latest        node2               Running             Running 5 minu

ulzgslctqjxz        web.2               nginx:latest        node1               Running             Running 8 seco

kjc266x6zs63         \_ web.2           nginx:latest        node3               Shutdown            Running 5 minu

[node1 ~]$ docker service scale web=3
web scaled to 3
overall progress: 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
[node1 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE                DESIRED STATE       CURRENT STATE
    PORTS
7rsvewrged4b        web.1               nginx:latest        node2               Running             Running 8 minu

ulzgslctqjxz        web.2               nginx:latest        node1               Running             Running 2 minu

kjc266x6zs63         \_ web.2           nginx:latest        node3               Shutdown            Shutdown 22 se

jpds59es0kt4        web.3               nginx:latest        node3               Running             Running 9 seco

[node1 ~]$ 
```

5. Grafana

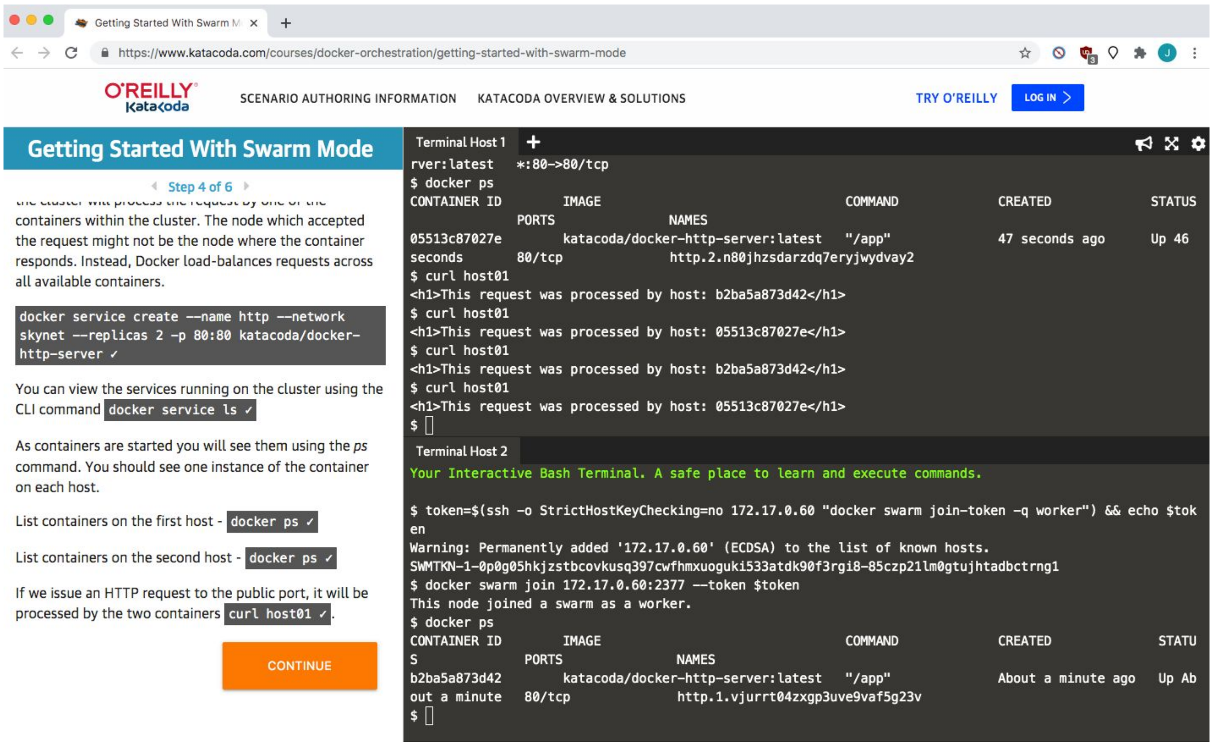


6. Docker logs



```
[node1 swarmprom]$ docker logs -f web.2.ulzgslctqjxzx36g8nsw8o413
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
10.0.0.3 - - [13/Oct/2020:16:42:32 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:33 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:34 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:34 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:35 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:35 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:36 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:36 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:37 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:38 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:38 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:39 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:40 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
10.0.0.3 - - [13/Oct/2020:16:42:40 +0000] "GET / HTTP/1.1" 200 6 "-" "ApacheBench/2.3" "-"
```

7. (Opcional) Testeo en Katakoda



Getting Started With Swarm M
https://www.katacoda.com/courses/docker-orchestration/getting-started-with-swarm-mode
O'REILLY Katacoda
SCENARIO AUTHORING INFORMATION
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Getting Started With Swarm Mode
Step 4 of 6
Terminal Host 1
Terminal Host 2
CONTINUE

containers within the cluster. The node which accepted the request might not be the node where the container responds. Instead, Docker load-balances requests across all available containers.

```
docker service create --name http --network skynet --replicas 2 -p 80:80 katacoda/docker-http-server ✓
```

You can view the services running on the cluster using the CLI command `docker service ls ✓`

As containers are started you will see them using the *ps* command. You should see one instance of the container on each host.

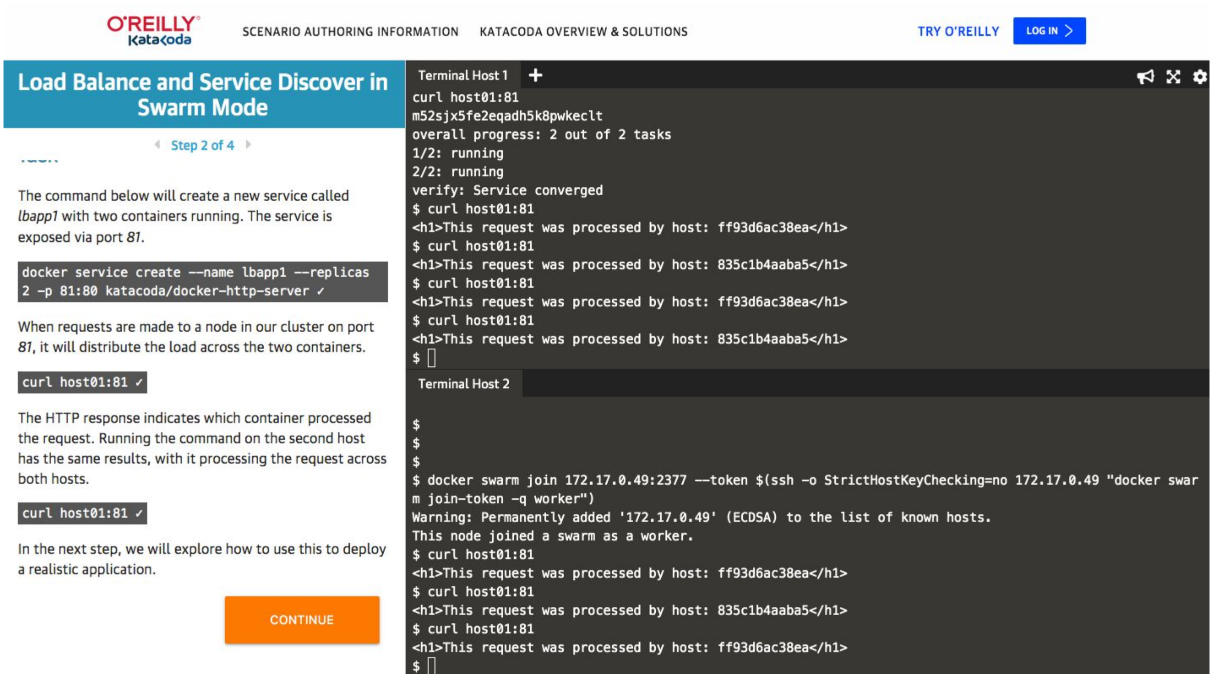
List containers on the first host - `docker ps ✓`

List containers on the second host - `docker ps ✓`

If we issue an HTTP request to the public port, it will be processed by the two containers `curl host01 ✓`.

```
rver:latest   *:80->80/tcp
$ docker ps
CONTAINER ID        IMAGE                                COMMAND             CREATED             STATUS
              PORTS                NAMES
05513c87027e        katacoda/docker-http-server:latest   "/app"              47 seconds ago      Up 46
seconds       80/tcp               http.2.n80jhzsdarzdq7eryjwydvay2
$ curl host01
<h1>This request was processed by host: b2ba5a873d42</h1>
$ curl host01
<h1>This request was processed by host: 05513c87027e</h1>
$ curl host01
<h1>This request was processed by host: b2ba5a873d42</h1>
$ curl host01
<h1>This request was processed by host: 05513c87027e</h1>
$
```

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ token=$(ssh -o StrictHostKeyChecking=no 172.17.0.60 "docker swarm join-token -q worker") && echo $tok
en
Warning: Permanently added '172.17.0.60' (ECDSA) to the list of known hosts.
SWMTKN-1-0p0g05hkjzstbcovkusq397cwfhmxuoguki533atdk90f3rgi8-85czp21lm0gtujhtadbctrng1
$ docker swarm join 172.17.0.60:2377 --token $token
This node joined a swarm as a worker.
$ docker ps
CONTAINER ID        IMAGE                                COMMAND             CREATED              STATU
S               PORTS                NAMES
b2ba5a873d42        katacoda/docker-http-server:latest   "/app"              About a minute ago   Up Ab
out a minute    80/tcp               http.1.vjurrt04zxgp3uve9vaf5g23v
$
```



O'REILLY Katacoda
SCENARIO AUTHORING INFORMATION
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Load Balance and Service Discover in Swarm Mode
Step 2 of 4
Terminal Host 1
Terminal Host 2
CONTINUE

The command below will create a new service called *lbapp1* with two containers running. The service is exposed via port *81*.

```
docker service create --name lbapp1 --replicas 2 -p 81:80 katacoda/docker-http-server ✓
```

When requests are made to a node in our cluster on port *81*, it will distribute the load across the two containers.

`curl host01:81 ✓`

The HTTP response indicates which container processed the request. Running the command on the second host has the same results, with it processing the request across both hosts.

`curl host01:81 ✓`

In the next step, we will explore how to use this to deploy a realistic application.

```
curl host01:81
m52sjx5fe2eqadh5k8pwkeclt
overall progress: 2 out of 2 tasks
1/2: running
2/2: running
verify: Service converged
$ curl host01:81
<h1>This request was processed by host: ff93d6ac38ea</h1>
$ curl host01:81
<h1>This request was processed by host: 835c1b4aaba5</h1>
$ curl host01:81
<h1>This request was processed by host: ff93d6ac38ea</h1>
$ curl host01:81
<h1>This request was processed by host: 835c1b4aaba5</h1>
$
```

```

$
$
$
$ docker swarm join 172.17.0.49:2377 --token $(ssh -o StrictHostKeyChecking=no 172.17.0.49 "docker swar
m join-token -q worker")
Warning: Permanently added '172.17.0.49' (ECDSA) to the list of known hosts.
This node joined a swarm as a worker.
$ curl host01:81
<h1>This request was processed by host: ff93d6ac38ea</h1>
$ curl host01:81
<h1>This request was processed by host: 835c1b4aaba5</h1>
$ curl host01:81
<h1>This request was processed by host: ff93d6ac38ea</h1>
$
```