Práctica 3: Docker Swarm

Autor: Manuel Díaz-Meco Terrés

Fecha: 12 de octubre de 2024

Introducción

El objetivo de esta práctica es trabajar con Docker para la creación de imágenes personalizadas, su publicación en Docker Hub, y el despliegue de múltiples contenedores utilizando Docker-compose. Además, se realizará una prueba opcional de limitación de CPU y la subida del contenedor con los resultados obtenidos a Docker Hub.

Ejecución del Servicio Web

En este apartado se crea el *swarm* en uno de los nodos y en los otros 2 se ejecuta el comando necesario para unirse al nodo creador o *manager*. A continuación se muestran las imágenes y los comandos utilizados:

Comandos utilizados:

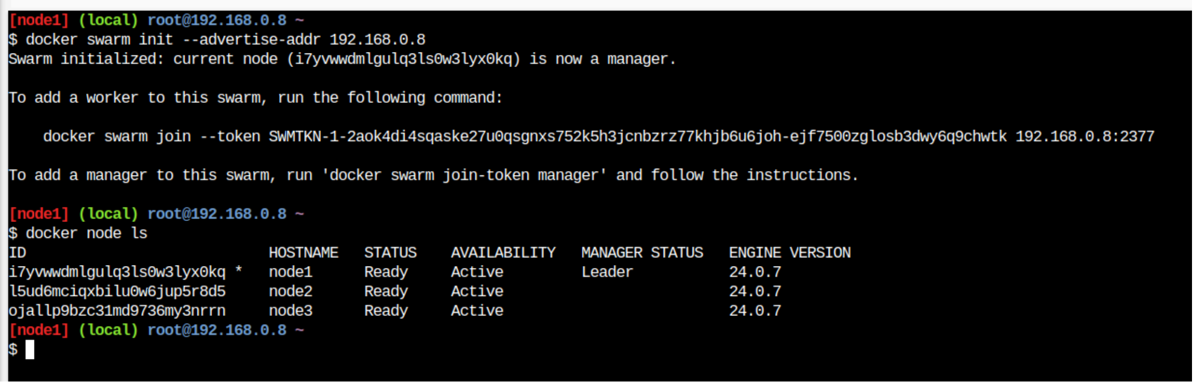
Nodo manager:

```
docker swarm init --advertise-addr 192.168.0.8
docker node ls
```

Nodos worker:

```
docker swarm join --token SWMTKN-1-
2aok4di4sqaske27u0qsgnxs752k5h3jcnbzrz77khjb6u6joh-ejf7500zglosb3dwy6q9chwtk
192.168.0.8:2377
```

Capturas de pantalla:



```
[node1] (local) root@192.168.0.8 ~
$ docker swarm init --advertise-addr 192.168.0.8
Swarm initialized: current node (i7yvwwdmlgulq3ls0w3lyx0kq) is now a manager.

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

    docker swarm join --token SWMTKN-1-2aok4di4sqaske27u0qsgnxs752k5h3jcnbzrz77khjb6u6joh-ejf7500zglosb3dwy6q9chwtk 192.168.0.8:2377

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

[node1] (local) root@192.168.0.8 ~
$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
i7yvwwdmlgulq3ls0w3lyx0kq *   node1      Ready     Active         Leader           24.0.7
l5ud6mciqxbilu0w6jup5r8d5     node2      Ready     Active                          24.0.7
ojallp9bzc31md9736my3nrrn     node3      Ready     Active                          24.0.7
[node1] (local) root@192.168.0.8 ~
$ 
```

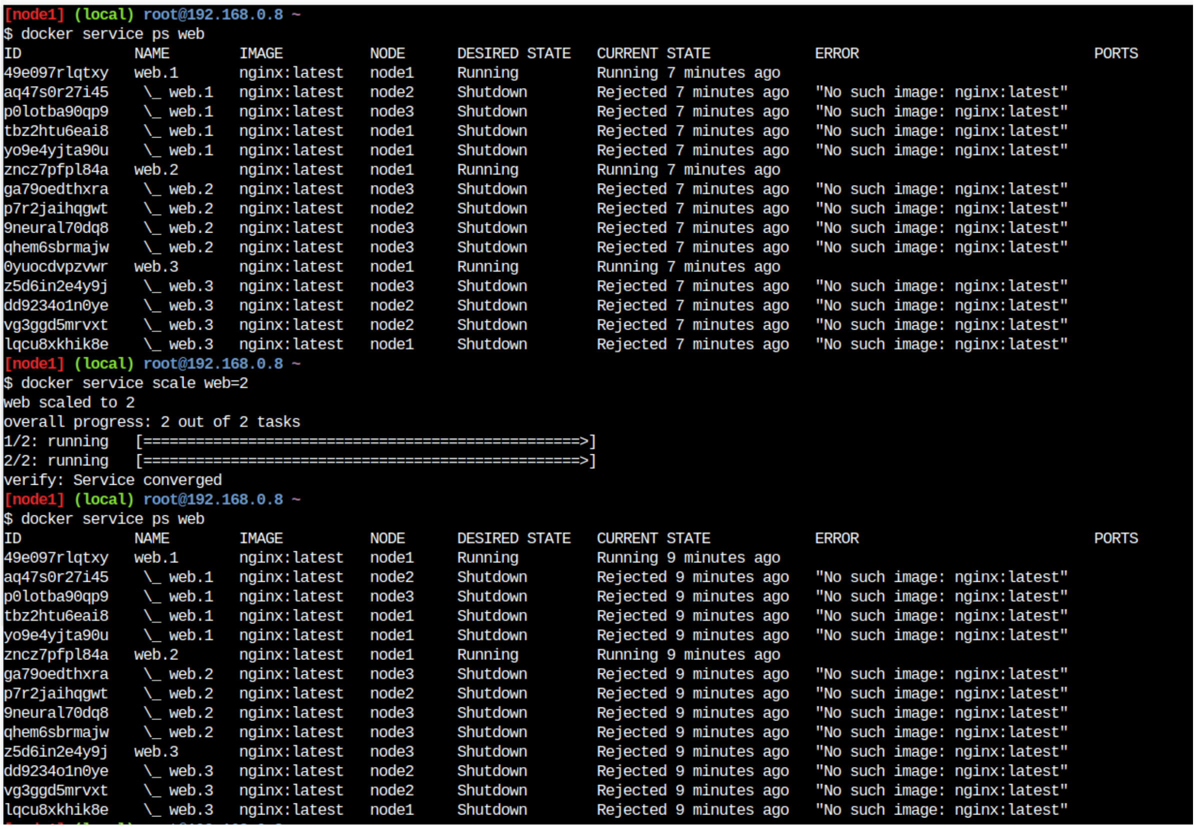
Cambio de Escala

En este apartado se muestra como afecta el cambio de escala a la hora de la ejecución del sistema "web". Notar que el aviso `No such image: nginx latest` es porque intenté ejecutar los comandos para lanzar el servicio web sin tener instalado ngnix en los nodos.

Comandos Utilizados:

```
docker service ps web
docker service scale web=2
docker service ps web
```

Capturas de Pantalla:



```
[node1] (local) root@192.168.0.8 ~
$ docker service ps web
ID             NAME         IMAGE          NODE      DESIRED STATE   CURRENT STATE           ERROR                              PORTS
49e097rlqtxy   web.1        nginx:latest   node1     Running         Running 7 minutes ago
aq47s0r27i45    \_ web.1    nginx:latest   node2     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
p0lotba90qp9    \_ web.1    nginx:latest   node3     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
tbz2htu6eai8    \_ web.1    nginx:latest   node1     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
yo9e4yjta90u    \_ web.1    nginx:latest   node1     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
zncz7pfpl84a   web.2        nginx:latest   node1     Running         Running 7 minutes ago
ga79oedthxra    \_ web.2    nginx:latest   node3     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
p7r2jaihqgwt    \_ web.2    nginx:latest   node2     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
9neural70dq8    \_ web.2    nginx:latest   node3     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
qhem6sbrmajw    \_ web.2    nginx:latest   node3     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
0yuocdvpzvwr   web.3        nginx:latest   node1     Running         Running 7 minutes ago
z5d6in2e4y9j    \_ web.3    nginx:latest   node3     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
dd9234o1n0ye    \_ web.3    nginx:latest   node2     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
vg3ggd5mrvxt    \_ web.3    nginx:latest   node2     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
lqcu8xkhik8e    \_ web.3    nginx:latest   node1     Shutdown        Rejected 7 minutes ago   "No such image: nginx:latest"
[node1] (local) root@192.168.0.8 ~
$ 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] (local) root@192.168.0.8 ~
$ docker service ps web
ID             NAME         IMAGE          NODE      DESIRED STATE   CURRENT STATE           ERROR                              PORTS
49e097rlqtxy   web.1        nginx:latest   node1     Running         Running 9 minutes ago
aq47s0r27i45    \_ web.1    nginx:latest   node2     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
p0lotba90qp9    \_ web.1    nginx:latest   node3     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
tbz2htu6eai8    \_ web.1    nginx:latest   node1     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
yo9e4yjta90u    \_ web.1    nginx:latest   node1     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
zncz7pfpl84a   web.2        nginx:latest   node1     Running         Running 9 minutes ago
ga79oedthxra    \_ web.2    nginx:latest   node3     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
p7r2jaihqgwt    \_ web.2    nginx:latest   node2     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
9neural70dq8    \_ web.2    nginx:latest   node3     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
qhem6sbrmajw    \_ web.2    nginx:latest   node3     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
z5d6in2e4y9j   web.3        nginx:latest   node3     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
dd9234o1n0ye    \_ web.3    nginx:latest   node2     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
vg3ggd5mrvxt    \_ web.3    nginx:latest   node2     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
lqcu8xkhik8e    \_ web.3    nginx:latest   node1     Shutdown        Rejected 9 minutes ago   "No such image: nginx:latest"
```

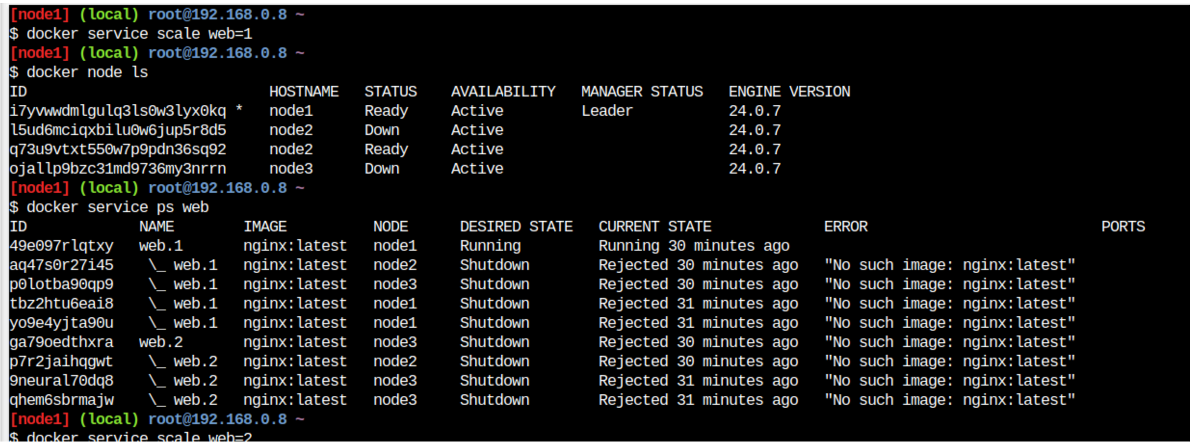
Ejecución de un único Nodo activación automática del segundo Nodo

En este apartado lo que se hace es borrar los nodos 2 y 3 y cambiar de escala para que el servicio web se adapte al uso de un único nodo, el *manager*. Tras crear un nuevo nodo y configurarlo como antes rehacemos el cambio de escala y hacemos que use 2 nodos nuevamente. Podemos ver en las capturas que hay 2 node2, uno apagado y otro listo para ser usado.

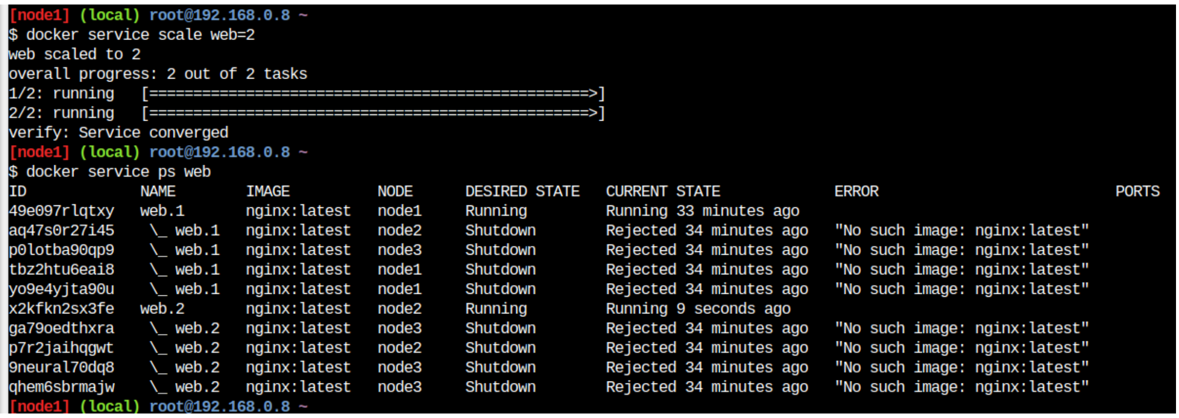
Comandos Utilizados:

```
docker service scale web=1
docker node ls
docker service ps web
docker service scale web=2
docker service ps web
```

Capturas de Pantalla:



```
[node1] (local) root@192.168.0.8 ~
$ docker service scale web=1
[node1] (local) root@192.168.0.8 ~
$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
i7yvwwdmlgulq3ls0w3lyx0kq *   node1      Ready     Active         Leader           24.0.7
l5ud6mciqxbilu0w6jup5r8d5     node2      Down      Active                          24.0.7
q73u9vtxt550w7p9pdn36sq92     node2      Ready     Active                          24.0.7
ojallp9bzc31md9736my3nrrn     node3      Down      Active                          24.0.7
[node1] (local) root@192.168.0.8 ~
$ docker service ps web
ID             NAME          IMAGE          NODE      DESIRED STATE   CURRENT STATE             ERROR                              PORTS
49e097rlqtxy   web.1         nginx:latest   node1     Running         Running 30 minutes ago
aq47s0r27i45    \_ web.1     nginx:latest   node2     Shutdown        Rejected 30 minutes ago   "No such image: nginx:latest"
p0lotba90qp9    \_ web.1     nginx:latest   node3     Shutdown        Rejected 30 minutes ago   "No such image: nginx:latest"
tbz2htu6eai8    \_ web.1     nginx:latest   node1     Shutdown        Rejected 31 minutes ago   "No such image: nginx:latest"
yo9e4yjta90u    \_ web.1     nginx:latest   node1     Shutdown        Rejected 31 minutes ago   "No such image: nginx:latest"
ga79oedthxra   web.2         nginx:latest   node3     Shutdown        Rejected 30 minutes ago   "No such image: nginx:latest"
p7r2jaihqgwt    \_ web.2     nginx:latest   node2     Shutdown        Rejected 30 minutes ago   "No such image: nginx:latest"
9neural70dq8    \_ web.2     nginx:latest   node3     Shutdown        Rejected 31 minutes ago   "No such image: nginx:latest"
qhem6sbrmajw    \_ web.2     nginx:latest   node3     Shutdown        Rejected 31 minutes ago   "No such image: nginx:latest"
[node1] (local) root@192.168.0.8 ~
$ docker service scale web=2
```



```
[node1] (local) root@192.168.0.8 ~
$ 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] (local) root@192.168.0.8 ~
$ docker service ps web
ID             NAME          IMAGE          NODE      DESIRED STATE   CURRENT STATE             ERROR                              PORTS
49e097rlqtxy   web.1         nginx:latest   node1     Running         Running 33 minutes ago
aq47s0r27i45    \_ web.1     nginx:latest   node2     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
p0lotba90qp9    \_ web.1     nginx:latest   node3     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
tbz2htu6eai8    \_ web.1     nginx:latest   node1     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
yo9e4yjta90u    \_ web.1     nginx:latest   node1     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
x2kfkn2sx3fe   web.2         nginx:latest   node2     Running         Running 9 seconds ago
ga79oedthxra    \_ web.2     nginx:latest   node3     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
p7r2jaihqgwt    \_ web.2     nginx:latest   node2     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
9neural70dq8    \_ web.2     nginx:latest   node3     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
qhem6sbrmajw    \_ web.2     nginx:latest   node3     Shutdown        Rejected 34 minutes ago   "No such image: nginx:latest"
[node1] (local) root@192.168.0.8 ~
```

Apartado Opcional con Vagrant

Para este último apartado vamos a generar los nodos y el manager a partir del uso de vagrant y virtualbox, en vez de usar la página web que se muestra en las prácticas y que he utilizado anteriormente. Tras instalar Vagrant y VirtualBox se crea un archivo Vagrantfile en el directorio donde estabamos desarrollando la práctica.

```
Vagrant.configure("2") do |config|
# Configuración del nodo manager
config.vm.define "manager" do |manager|
    manager.vm.box = "ubuntu/focal64"
    manager.vm.network "private_network", type: "dhcp"
    manager.vm.hostname = "manager"
    manager.vm.provider "virtualbox" do |vb|
    vb.memory = "1024"
    end

    manager.vm.synced_folder ".", "/vagrant", disabled: true

    manager.vm.provision "shell", inline: <<-SHELL
    sudo apt-get update
    sudo apt-get install -y docker.io
    sudo docker swarm init --advertise-addr $(hostname -I | cut -d' ' -f1)
    SHELL
end

# Configuración de los nodos
(1..2).each do |i|
    config.vm.define "nodo#{i}" do |worker|
    worker.vm.box = "ubuntu/focal64"
    worker.vm.network "private_network", type: "dhcp"
    worker.vm.hostname = "nodo#{i}"
    worker.vm.provider "virtualbox" do |vb|
        vb.memory = "1024"
    end

    worker.vm.synced_folder ".", "/vagrant", disabled: true

    worker.vm.provision "shell", inline: <<-SHELL
        sudo apt-get update
        sudo apt-get install -y docker.io
    SHELL
    end
end
end
```

Comandos Utilizados:

En mi máquina principal:

```
vagrant up
vagrant ssh manager
vagrant ssh nodo1
vagrant ssh nodo2
```

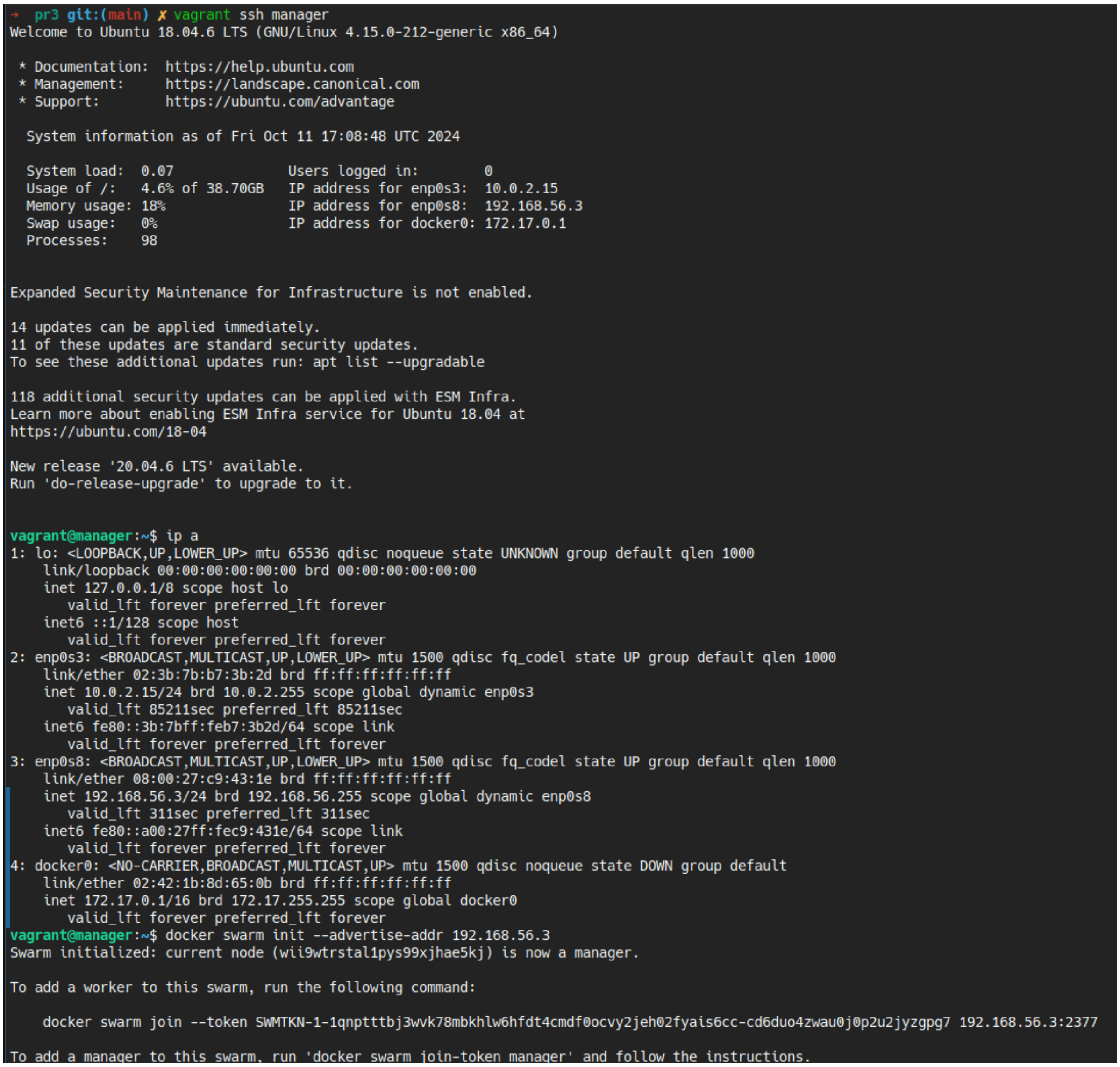
En el nodo manager:

```
docker swarm init --advertise-addr 192.168.56.3
docker node ls
```

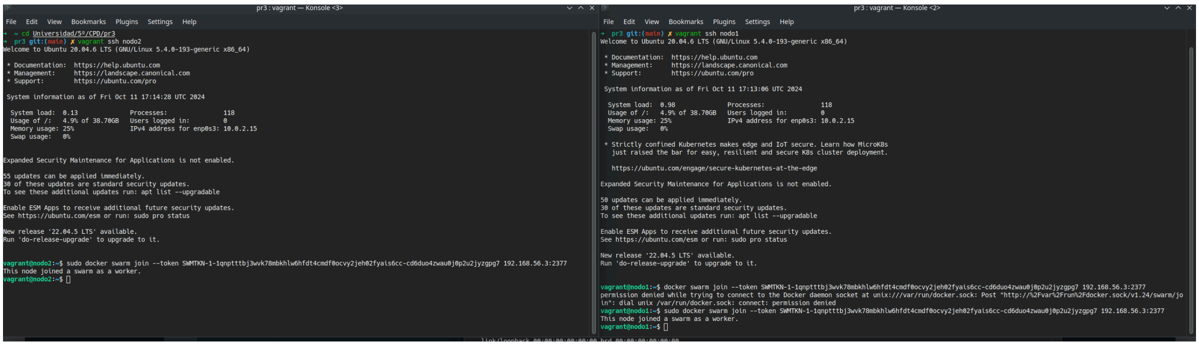
En los nodos:

```
sudo docker swarm join --token SWMTKN-1-1qnptttbj3wvk78mbkhlw6hfdt4cmdf0ocvy2jeh02fyais6cc-cd6duo4zwau0j0p2u2jyzgpg7 192.168.56.3:2377
```

Capturas de Pantalla:



```
➜  pr3 git:(main) ✗ vagrant ssh manager
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-212-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Oct 11 17:08:48 UTC 2024

  System load:  0.07               Users logged in:        0
  Usage of /:   4.6% of 38.70GB    IP address for enp0s3:  10.0.2.15
  Memory usage: 18%                IP address for enp0s8:  192.168.56.3
  Swap usage:   0%                 IP address for docker0: 172.17.0.1
  Processes:    98


Expanded Security Maintenance for Infrastructure is not enabled.

14 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

118 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@manager:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:3b:7b:b7:3b:2d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 85211sec preferred_lft 85211sec
    inet6 fe80::3b:7bff:feb7:3b2d/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c9:43:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.3/24 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 311sec preferred_lft 311sec
    inet6 fe80::a00:27ff:fec9:431e/64 scope link
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:1b:8d:65:0b brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
vagrant@manager:~$ docker swarm init --advertise-addr 192.168.56.3
Swarm initialized: current node (wii9wtrstal1pys99xjhae5kj) is now a manager.

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

    docker swarm join --token SWMTKN-1-1qnptttbj3wvk78mbkhlw6hfdt4cmdf0ocvy2jeh02fyais6cc-cd6duo4zwau0j0p2u2jyzgpg7 192.168.56.3:2377

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

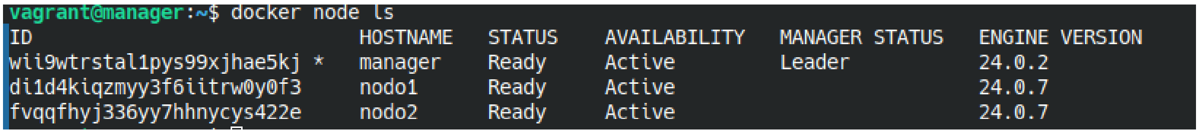


pr3 : vagrant — Konsole <3>
File Edit View Bookmarks Plugins Settings Help

```
➜  ~ cd Universidad/5º/CPD/pr3
➜  pr3 git:(main) ✗ vagrant ssh nodo2
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-193-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Oct 11 17:14:28 UTC 2024

  System load:  0.13               Processes:               118
  Usage of /:   4.9% of 38.70GB    Users logged in:         0
  Memory usage: 25%                IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

55 updates can be applied immediately.
30 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@nodo2:~$ sudo docker swarm join --token SWMTKN-1-1qnptttbj3wvk78mbkhlw6hfdt4cmdf0ocvy2jeh02fyais6cc-cd6duo4zwau0j0p2u2jyzgpg7 192.168.56.3:2377
This node joined a swarm as a worker.
vagrant@nodo2:~$ 
```

pr3 : vagrant — Konsole <2>
File Edit View Bookmarks Plugins Settings Help

```
➜  pr3 git:(main) ✗ vagrant ssh nodo1
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-193-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Oct 11 17:13:06 UTC 2024

  System load:  0.98               Processes:               118
  Usage of /:   4.9% of 38.70GB    Users logged in:         0
  Memory usage: 25%                IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

50 updates can be applied immediately.
30 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@nodo1:~$ docker swarm join --token SWMTKN-1-1qnptttbj3wvk78mbkhlw6hfdt4cmdf0ocvy2jeh02fyais6cc-cd6duo4zwau0j0p2u2jyzgpg7 192.168.56.3:2377
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/swarm/jo
in": dial unix /var/run/docker.sock: connect: permission denied
vagrant@nodo1:~$ sudo docker swarm join --token SWMTKN-1-1qnptttbj3wvk78mbkhlw6hfdt4cmdf0ocvy2jeh02fyais6cc-cd6duo4zwau0j0p2u2jyzgpg7 192.168.56.3:2377
This node joined a swarm as a worker.
vagrant@nodo1:~$ 
```



```
vagrant@manager:~$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
wii9wtrstal1pys99xjhae5kj *   manager    Ready     Active         Leader           24.0.2
di1d4kiqzmyy3f6iitrw0y0f3     nodo1      Ready     Active                          24.0.7
fvqqfhyj336yy7hhnycys422e     nodo2      Ready     Active                          24.0.7
```