

M11 – Seguretat Informàtica

Pràctica 3 – Cluster Docker_v03

Nil Massó



ASIX 2			
M11 – Seguretat Informàtica – UF1		Tipus	Individual
Cognoms, Nom:	Massó Cabaña, Nil	Curs	2022-23
Observacions:			

Table of Contents

Pràctica 3 – Clusters amb Docker Swarm.....	1
Enllaços.....	1
Part I: Creació de les imatges amb Dockerfile (5p).....	1
Part II: Creació del Docker swarm (3p).....	3
Part III: Escalabilitat amb Docker Swarm (2p).....	4

Pràctica 3 – Clusters amb Docker Swarm

Enllaços

https://linuxhint.com/mount_vmware_shares_command_line_linux_vm/
<https://www.howtogeek.com/devops/how-to-use-docker-to-containerise-php-and-apache/>
https://dockertips.com/apache_y_php
<https://dockerwebdev.com/tutorials/docker-php-development/>
<https://dockerfile.readthedocs.io/en/latest/content/DockerImages/dockerfiles/php-nginx.html>
https://www.youtube.com/watch?v=m6WgX_LBtEk

Treballes en una empresa que ha aconseguit un contracte municipal per a fer una botiga web de venda d'entrades dels esdeveniments que organitza l'Ajuntament. La botiga tindrà pics de tràfic quan s'anunciïn els concerts de la festa major, per tant, suggereixes fer servir un sistema de contenidors per allotjar diferents servidors web que es puguin escalar. La teva cap de departament et demana que preparis una demo per a convèncer el gerent que l'ús de Docker és una opció a considerar.

Part I: Creació de les imatges amb Dockerfile (5p)

A partir de les imatges del servidor web i del servidor de base de dades que vas crear a la pràctica anterior, crea un cluster de nodes per a dotar d'alta disponibilitat a la botiga virtual que us han encarregat.

Necessitaràs dues màquines virtuals que tinguin Docker instal·lat. Pots fer una còpia de la màquina virtual de la pràctica anterior o treballar amb l'Isard/VDI.

Aprofitarem les imatges de la pràctica anterior, ja que ja contenen un servidor web amb php i una base de dades que funcionen. Però les imatges no contenen el codi de la pàgina web, ja que treballàvem sobre volums. Aquesta nova imatge ens permetrà crear un cluster 100% funcional.

1) Primer **posa en marxa el contenidor amb la base de dades**, però hi hauran alguns canvis respecte l'anterior pràctica: **(1,5 punts)**

- **No especifiquis cap interfície de xarxa.** Farem servir la per defecte
- **El volum (nom datadbJPswarm) es muntarà a /var/lib/mysql**
- Recorda **fer servir el paràmetre -dit** per a poder entrar en mode consola als contenidors
- El servidor web ha d'escoltar pel port 3306 del teu host: **-p 3306:3306**
- El servidor de base de dades s'ha d'engegar amb les variables d'entorn següents:

o **-e MYSQL_ROOT_PASSWORD=test -e MYSQL_DATABASE=blog_samples**

- El contenidor de la base de dades s'ha de dir **dbserverInicialNomInicialCognomSwarm**
- Hauràs de **tornar a importar la taula de la web d'exemple a la base de dades**

Posa captures de com has engegat el contenidor de la base de dades, llista els contenidors que estan funcionant i mostra el contingut de la taula a la base de dades:

```
root@us-nmc:~# docker run -dit -p 3306:3306 -v datadbnmswarm:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=test -e MYSQL_DATABASE=blog_samples --name dbservernmswarm nmasso1/dbservernm:prac2
85203cc3b940e8750f8d1c70055f59cf4b507384b7cfeb1000a6f5e2c79a070f
root@us-nmc:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
85203cc3b940	nmasso1/dbservernm:prac2	"docker-entrypoint.s..."	42 minutes ago	Up 42 minutes	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	dbservernmswarm

```
mysql> use blog_samples
Database changed
mysql> show tables;
Empty set (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_blog_samples |
+-----+
| tblproduct              |
+-----+
1 row in set (0.00 sec)

mysql> select * from tblproduct;
```

id	name	code	image	price
1	FinePix Pro2 3D Camera	3DcAM01	product-images/camera.jpg	1500.00
2	EXP Portable Hard Drive	USB02	product-images/external-hard-drive.jpg	800.00
3	Luxury Ultra thin Wrist Watch	wristWear03	product-images/watch.jpg	300.00
4	XP 1155 Intel Core Laptop	LPN45	product-images/laptop.jpg	800.00

2) El mètode que farem servir per crear una nova imatge serà mitjançant una definició en un fitxer **Dockerfile**, tal i com hem vist als apunts. **(2 punts)**

En un directori del host (per exemple a ~/prac3/) hi crees un fitxer Dockerfile (**tens exemples als enllaços del principi de la pràctica**) i també hi copies tot el codi necessari per a que funcioni la web. Pots partir d'una imatge de l'nginx o de l'Apache. Abans modifica el contingut de **dbcontroller.php** i hi poses **l'adreça IP del host**. Així aconseguirem que els nodes del cluster, també vegin el servidor de la base de dades. Quelcom semblant a:

Al Dockerfile li direm que copii tot el codi i fitxers i directoris necessaris de la botiga dins de `/var/www/html` o del directori que tingui el contenidor que correrà el servidor web.

Recorda que també cal instal·lar-hi el mysql. Ara és un bon moment per a inclur-ho al fitxer Dockerfile!!

Posa una captura del contingut del directori que es copiarà a `/var/www/html` o el que sigui quan es creï la imatge i del contingut de Dockerfile. Mostra el canvi dins de `dbcontroller.php` i la IP del host.

```
root@us-nmc:~/prac3# ls -la botiga/
total 48
drwxr-xr-x 3 root root 4096 Aug 31 2018 .
drwxr-xr-x 4 root root 4096 Mar 16 18:14 ..
-rw-r--r-- 1 root root 8196 Aug 30 2018 .DS_Store
-rw-r--r-- 1 root root 737 Aug 24 2018 dbcontroller.php
-rw-r--r-- 1 root root 250 Aug 30 2018 icon-delete.png
-rwxr-xr-x 1 root root 4387 Aug 31 2018 index.php
drwxr-xr-x 2 root root 4096 Aug 30 2018 product-images
-rwxr-xr-x 1 root root 1862 Aug 31 2018 style.css
-rw-r--r-- 1 root root 1043 Aug 31 2018 tblproduct.sql
```

```
root@us-nmc:~/prac3# cat botiga/dbcontroller.php | grep '$host'
    private $host = "192.168.122.227";
root@us-nmc:~/prac3# cat Dockerfile
FROM php:7.4-apache
RUN docker-php-ext-install mysqli

COPY botiga/ /var/www/html/

EXPOSE 80
root@us-nmc:~/prac3# cat botiga/dbcontroller.php | grep '$host'
    private $host = "192.168.122.227";
root@us-nmc:~/prac3# 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: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:7c:e2:66 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 192.168.122.227/24 metric 100 brd 192.168.122.255 scope global dynamic ens3
```

3) Amb el que hem vist als apunts, crea la nova imatge del servidor web a partir del fitxer Dockerfile anterior i li poses el nom `usuari/dockerhub/webbotigaInicialNomInicialCognom` i el tag **prac3**. (1,5 punts)

Engega el contenidor a partir d'aquesta imatge. A diferència de la pràctica anterior:

- **No especifiquis cap interfície de xarxa.** Farem servir la per defecte
- Recorda **fer servir el paràmetre -dit** per a poder entrar en mode consola als contenidors
- El servidor web ha d'escoltar pel port 80 del teu host: **-p 80:80**
- El contenidor del servei web s'ha de dir **webserverInicialNomInicialCognomSwarm**
- Envia la imatge al teu Dockerhub amb un push. Els nodes del swarm se la podran descarregar

Posa captures de com has engegat el servidor web, llista els contenidors que estan funcionant i demostra que **funciona com a l'anterior pràctica, però ara ho fem sense volum!:**

```
root@us-nmc:~/prac3# docker run -dit -p 80:80 --name webservernmswazm nmassol/webbotiganm:prac3
12317cd9609e42f08ac80359900558f9353e9a28e996b4c5b923ca2a08f7a6db
root@us-nmc:~/prac3# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
12317cd9609e	nmassol/webbotiganm:prac3	"docker-php-entrypoi..."	25 seconds ago	Up 24 seconds	0.0.0.0:80->80/tcp, :::80->80/tcp	webservernmswazm
be9d5ddd1cec	nmassol/dbservernm:prac2	"docker-entrypoint.s..."	14 minutes ago	Up 14 minutes	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	dbservernmswazm


→ http://192.168.122.227/index.php

Shopping Cart

Empty Cart


Your Cart is Empty

Products




FinePix Pro2 3D Camera

\$1500.00 1 Add to Cart



EXP Portable Hard Drive

\$800.00 1 Add to Cart

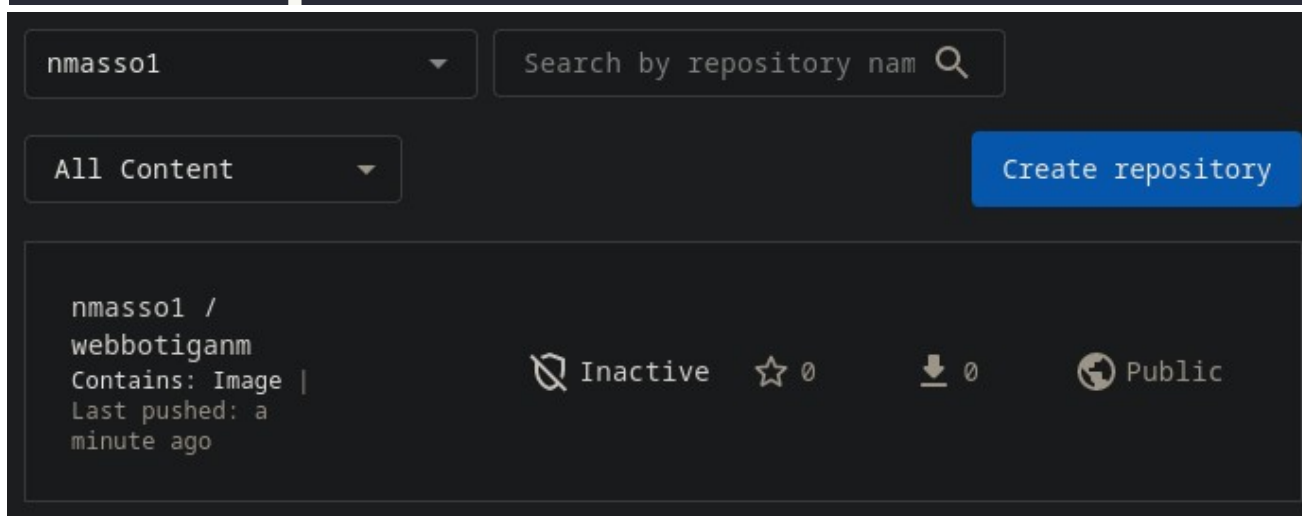


Luxury Ultra thin Wrist Watch

\$300.00 1 Add to Cart

```
root@us-nmc:~/prac3# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@us-nmc:~/prac3# docker push nmasso1/webbotiganm:prac3
The push refers to repository [docker.io/nmasso1/webbotiganm]
2a46e456ad06: Preparing
a4def1e152ec: Preparing
3d33242bf117: Preparing
529016396883: Preparing
5464bcc3f1c2: Preparing
28192e867e79: Preparing
d173e78df32e: Mounted from library/php
0be1ec4fbfdc: Mounted from library/php
30fa0c430434: Mounted from library/php
a538c5a6e4e0: Mounted from library/php
e5d40f64dcb4: Mounted from library/php
44148371c697: Mounted from library/php
797a7c0590e0: Mounted from library/php
f60117696410: Mounted from library/php
ec4a38999118: Mounted from library/php
prac3: digest: sha256:08101904d13430ea74fc033d89e1f30ab8be49ee1fa11eb31c71300c2448a8af size: 3453
```



Part II: Creació del Docker swarm (3p)

4) **Apaga el contenidor del servidor web i l'esborres. Manté en funcionament el contenidor de la base de dades.**

Ara iniciarem el mode swarm en aquesta màquina virtual. Això ens generarà un hash que servirà per afegir nodes al swarm. Posa una captura del resultat de *docker info* on es vegi la informació del swarm i l'adreça de xarxa del swarm **(0,5 punts)**

```
Swarm: active
NodeID: 7pd7wnengc36150p95oukv0vx
Is Manager: true
ClusterID: g3v6dkb57wo1gueao785srjrr
Managers: 1
Nodes: 1
Default Address Pool: 10.0.0.0/8
SubnetSize: 24
Data Path Port: 4789
Orchestration:
Task History Retention Limit: 5
```

5) Engega l'altra màquina virtual i l'afegeixes al swarm que ha creat la primera. Posa una captura del resultat de *docker info* on es vegi la informació del swarm

A la primera màquina, la que fa de **Manager**, hi mires quants nodes té el swarm. Fes-ne el llistat i posa la captura. **(0,5 punts)**

```
Swarm: active
NodeID: 7pd7wnengc36150p95oukv0vx
Is Manager: true
ClusterID: g3v6dkb57wo1gueao785srjrr
Managers: 1
Nodes: 2
Default Address Pool: 10.0.0.0/8
SubnetSize: 24
Data Path Port: 4789
Orchestration:
Task History Retention Limit: 5
```

2nodes

6) Ara ja podem engegar el servei que el swarm gestionarà. Engega el servei al manager del swarm a partir de la imatge amb el tag *prac3*:

- El servei s'ha de dir **webserver**
- Ha d'escollar pel port 80 -p 80:80

Posa una captura del resultat al Manager. Mira els contenidors que hi ha corrent amb *docker ps* i posa una captura. Hi ha algun contenidor amb el servei webserver? **(0,5 punts)**


```

root@us-nmc:~# docker service create --name webserver -p 80:80 nmassol/webbotiganm:prac3
q3h1oywz7p85zef46q061r7aw
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged

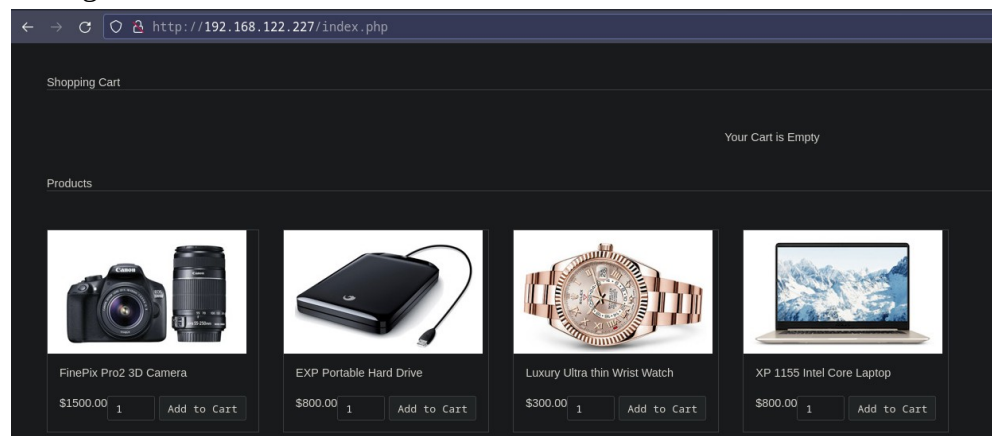
root@us-nmc:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
cc73aa34f27c   nmassol/webbotiganm:prac3          "docker-php-entrypoi..." 2 minutes ago  Up 2 minutes  80/tcp                             webserver.1.2fwirne37mof6vawphkc3gl
be9d5ddd1cec   nmassol/dbservernm:prac2          "docker-entrypoint.s..." 2 hours ago    Up 3 seconds  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp  dbservernmwarm

```

Si

7) Connecta't amb un navegador i comprova que el servei webserver funciona. On és el contenidor? Mostra una captura dels contenidors que hi ha al node que no és manager. Hi és aquí? Com es diu? (0,75 punts)

El contenidor es al manager, on l'hem iniciat, a l'altre node no hi ha res ja que no es requereix Manager:



```

root@us-nmc:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
cc73aa34f27c   nmassol/webbotiganm:prac3          "docker-php-entrypoi..." 2 minutes ago  Up 2 minutes  80/tcp                             webserver.1.2fwirne37mof6vawphkc3gl
be9d5ddd1cec   nmassol/dbservernm:prac2          "docker-entrypoint.s..." 2 hours ago    Up 3 seconds  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp  dbservernmwarm

```

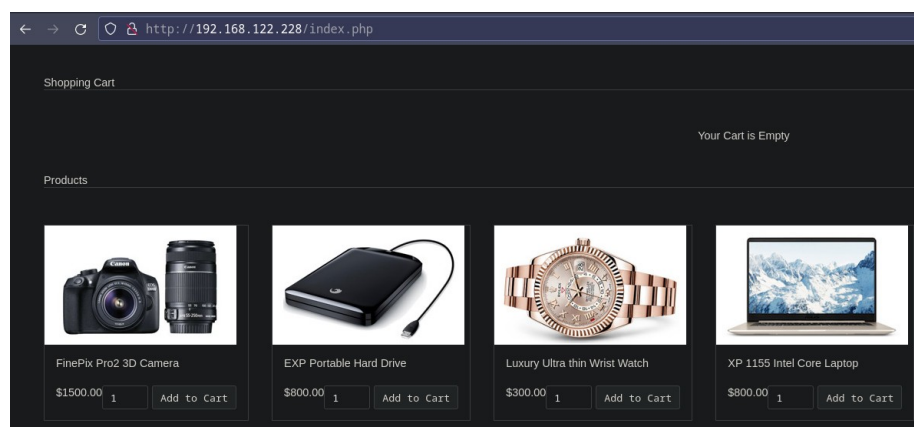
Node:

```

root@us-nmc:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES

```

Curiosament, podem veure com si accedim a la ip del node també podem accedir al serveriweb



8) Llista les interfícies de xarxa de docker i fixa't amb la que és del swarm. Inspecciona-la en els dos nodes i posa la captura on es vegi quina adreça IP té a cada un dels dos nodes. Al node no manager, hi ha una IP associada al contenidor que hi ha corrent. Posa una captura on es vegi la IP i el nom del contenidor **(0,75 punts)**

```

root@us-nmc:~# docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
ba47df81b5a7    botigaNM  bridge  local
0eb39f05a068    bridge    bridge  local
01ebb2967e34    docker_gwbridge  bridge  local
01386ab5feb0    host      host    local
sftmy2h5a6z1    ingress   overlay  swarm
6b7fe2e93266    none      null     local
root@us-nmc:~# docker network inspect sft
{
  "Name": "ingress",
  "Id": "sftmy2h5a6z1wle5n4cqj343w",
  "Created": "2023-03-17T11:46:14.373042986Z",
  "Scope": "swarm",
  "Driver": "overlay",
  "EnableIPv6": false,
  "IPAM": {
    "Driver": "default",
    "Options": null,
    "Config": [
      {
        "Subnet": "10.0.0.0/24",
        "Gateway": "10.0.0.1"
      }
    ]
  },
  "Internal": false,
  "Attachable": false,
  "Ingress": true,
  "ConfigFrom": {
    "Network": ""
  },
  "ConfigOnly": false,
  "Containers": {
    "ingress-sbox": {
      "Name": "ingress-endpoint",
      "EndpointID": "4dcc42330df72650f39eb2a065bde258dbff5bf1a7da0af8402098341f7c9129",
      "MacAddress": "02:42:0a:00:00:03",
      "IPv4Address": "10.0.0.3/24",
      "IPv6Address": ""
    }
  }
}

```

Node

```

root@us-nmc:~# docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
ba47df81b5a7    botigaNM  bridge  local
9c01d7fa2943    bridge    bridge  local
61ebb2967e34    docker_gwbridge  bridge  local
01386ab5feb0    host      host    local
sftmy2h5a6z1    ingress   overlay  swarm
6b7fe2e93266    none      null     local
root@us-nmc:~# docker network inspect sft
{
  "Name": "ingress",
  "Id": "sftmy2h5a6z1wle5n4cqj343w",
  "Created": "2023-03-17T11:35:45.336401454Z",
  "Scope": "swarm",
  "Driver": "overlay",
  "EnableIPv6": false,
  "IPAM": {
    "Driver": "default",
    "Options": null,
    "Config": [
      {
        "Subnet": "10.0.0.0/24",
        "Gateway": "10.0.0.1"
      }
    ]
  },
  "Internal": false,
  "Attachable": false,
  "Ingress": true,
  "ConfigFrom": {
    "Network": ""
  },
  "ConfigOnly": false,
  "Containers": {
    "cc73aa34f27c2eb044c8447aeb6aa9898e34d141c354f0bdcff19298d7e2b80f": {
      "Name": "webserver.1.2fwine37mof6vawphkcnt3gl",
      "EndpointID": "cda70f3386fc5cb416cc9a1c82ff15c92961ee7eff9cee058a118fbfae08d1",
      "MacAddress": "02:42:0a:00:00:05",
      "IPv4Address": "10.0.0.5/24",
      "IPv6Address": ""
    }
  }
}

```

Manager

```

"Options": {
  "com.docker.network.driver.overlay.vxlanid_list": "4096"
},
"Labels": {},
"Peers": [
  {
    "Name": "3a8395d97556",
    "IP": "192.168.122.227"
  },
  {
    "Name": "2b05ea061eb3",
    "IP": "192.168.122.228"
  }
]
}

```

```

"Options": {
  "com.docker.network.driver.overlay.vxlanid_list": "4096"
},
"Labels": {},
"Peers": [
  {
    "Name": "3a8395d97556",
    "IP": "192.168.122.227"
  },
  {
    "Name": "2b05ea061eb3",
    "IP": "192.168.122.228"
  }
]
}

```

Podem veure tant les associacions de peers amb ip com les de les ips dels dockers.

Part III: Escalabilitat amb Docker Swarm (2p)

9) Augmenta ara l'escalat del servei, posa'l a 2. I mira ara els contenidors que hi han al manager i al no manager. Posa captura de pantalla. Fes més peticions al servidor web. Pots crear més connexions fent servir una finestra d'incògnit al navegador. Mira els logs dels dos contenidors, i veuràs que les peticions s'han anat repartint als contenidors. Posa'n captures **(0,5 punts)**

```

root@us-nmc:~# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORT
cc73aa34f27c  mnasol/webbotigam:prac3  "docker-php-entrypoi..."  45 minutes ago  Up 45 minutes  80/t
cp             webserver.1.2fwine37mof6vawphkcnt3gl
be9d5dddicec  mnasol/dbserver:prac2    "docker-entrypoint.s..."  3 hours ago    Up 42 minutes  0.0.
0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp  dbserver:mnasol

```

Manager

```

root@us-nmc:~# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORT
774b0e86c8c8  mnasol/webbotigam:prac3  "docker-php-entrypoi..."  20 seconds ago  Up 18 seconds  80/t
cp             webserver.2.182q15x370tiwu3yffjcvbmax

```

Node

Manager:

```
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /index.php HTTP/1.1" 200 1040 "-" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /style.css HTTP/1.1" 200 968 "http://192.168.122.228/index.php" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /product-images/camera.jpg HTTP/1.1" 200 11447 "http://192.168.122.228/index.php" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /product-images/watch.jpg HTTP/1.1" 200 18691 "http://192.168.122.228/index.php" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /favicon.ico HTTP/1.1" 404 493 "http://192.168.122.228/index.php" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:25 +0000] "GET /index.php HTTP/1.1" 200 975 "-" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:29 +0000] "GET /index.php HTTP/1.1" 200 975 "-" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
```

Node:

```
10.0.0.3 - - [17/Mar/2023:13:07:15 +0000] "GET /index.php HTTP/1.1" 200 1040 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /product-images/laptop.jpg HTTP/1.1" 200 14829 "http://192.168.122.228/index.php" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
10.0.0.3 - - [17/Mar/2023:13:07:21 +0000] "GET /product-images/external-hard-drive.jpg HTTP/1.1" 200 9054 "http://192.168.122.228/index.php" "Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0"
```

S'han repartit les peticions

10) Augmenta ara l'escalat del servei, **posa'l a 4**. I mira ara els contenidors que hi han al manager i al no manager. Posa captura de pantalla. Es van repartint? Torna a inspeccionar la interfície de xarxa **ingress** i mira si hi han noves assignacions als nous contenidors. Posa'n captures. **(0,5 punts)**

Man:

```
root@us-nmc:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
0197cf50797b   nmasso1/webbotiganm:prac3          "docker-php-entrypoi..." About a minute ago Up About a minute 80/tcp              webserver.3.vtcmapxn7d2j4qbloexugjaa
cc73aa34f27c   nmasso1/webbotiganm:prac3          "docker-php-entrypoi..." About an hour ago Up About an hour 80/tcp              webserver.1.2fwrine37mof6vawphkcnc3g1
be9d50d01cec   nmasso1/dbserverim:prac2           "docker-entrypoint.s..." 3 hours ago    Up 59 minutes  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp    dbserverimmswazm
```

Node:

```
root@us-nmc:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
2b8101247f38   nmasso1/webbotiganm:prac3          "docker-php-entrypoi..." 2 minutes ago Up 2 minutes  80/tcp              webserver.4.mf4ye8mffl7ufzxw1qzghkd5w
774b9e06c8c8   nmasso1/webbotiganm:prac3          "docker-php-entrypoi..." 25 minutes ago Up 25 minutes  80/tcp              webserver.2.i82ql5x376t1wu9yffjcvbmax
```

Clarament es van repartint els contenidors

```
"Containers": {
  "2b8101247f380bb6b48f05fb24405d2ffa20548c765d65f849fe2f56cdce9320": {
    "Name": "webserver.4.mf4ye8mffl7ufzxw1qzghkd5w",
    "EndpointID": "7314a7e75f9a55b64b4e38e7c8b3355c3e6d3e3e6a5a299d5279dfba0ed7839f",
    "MacAddress": "02:42:0a:00:00:08",
    "IPv4Address": "10.0.0.8/24",
    "IPv6Address": ""
  },
  "774b9e06c8c87b0b296e75d0489f032ea07aa9d2dee9524f2b2d8fda4212ed4a": {
    "Name": "webserver.2.i82ql5x376t1wu9yffjcvbmax",
    "EndpointID": "a29ba6f2a1bfedff7f00b79abbd653d930adc7f83fc4d78c25cfdb88ab1a8940",
    "MacAddress": "02:42:0a:00:00:06",
    "IPv4Address": "10.0.0.6/24",
    "IPv6Address": ""
  },
  "ingress-sbox": {
    "Name": "ingress-endpoint",
    "EndpointID": "4dcc42330df72650f39eb2a065bde258dbff5bf1a7da0af8402098341f7c9129",
    "MacAddress": "02:42:0a:00:00:03",
    "IPv4Address": "10.0.0.3/24",
    "IPv6Address": ""
  }
},
"Configmap": {
  "Containers": {
    "0197cf50797b16b19db02d84f5500c05717264d89344c7174dedd2da51e196a3": {
      "Name": "webserver.3.vtcmapxn7d2j4qbloexugjaa",
      "EndpointID": "0b0721a67fbf50f70dc92188dacd370b22554b938dace2186648bbe861d950d3",
      "MacAddress": "02:42:0a:00:00:07",
      "IPv4Address": "10.0.0.7/24",
      "IPv6Address": ""
    },
    "cc73aa34f27c2eb044c8447aeb6aa9898e34d141c354f0bdcff19298d7e2b80f": {
      "Name": "webserver.1.2fwrine37mof6vawphkcnc3g1",
      "EndpointID": "cda70f3386fc5cb4166cc9a1c82ff15c92961ee7eff9cee050a118bfbae08d1",
      "MacAddress": "02:42:0a:00:00:05",
      "IPv4Address": "10.0.0.5/24",
      "IPv6Address": ""
    },
    "ingress-sbox": {
      "Name": "ingress-endpoint",
      "EndpointID": "aa83b30e9bf39c13932ede390180a03b3a55d153c9372207dd1399748e9ffbb",
      "MacAddress": "02:42:0a:00:00:02",
      "IPv4Address": "10.0.0.2/24",
      "IPv6Address": ""
    }
  }
}
```

Els hi dona ip sino no podria redirigirlis les peticions

11) Posa al node no manager en mode **drain** i mira com afecta això als contenidors del **swarm**. Han desaparegut? Posa captures. Mira també l'estat dels nodes segons els màner. Seguidament torna a activar el node. Apareixen els contenidors? Posa'n captures **(0,5 punts)**

```
root@us-nmc:~# docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
7pd7wnengc36150p95oukv0vx *	us-nmc	Ready	Active	Leader	23.0.1
pxmchgi2ubckx6jcsvn5cegd1	us-nmc	Ready	Active		23.0.1

```
root@us-nmc:~# docker node update --availability drain pxmchgi2ubckx6jcsvn5cegd1
pxmchgi2ubckx6jcsvn5cegd1
root@us-nmc:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2469b46ca9db	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	7 seconds ago	Up 1 second	80/tcp	webserver.4.rdhbnjzy59dxu420ahlsv2g4d
fc5907bbc51c	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	7 seconds ago	Up 1 second	80/tcp	webserver.2.cb7nvebkvh89vp98r272fadtw
0197cf50797b	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	28 minutes ago	Up 28 minutes	80/tcp	webserver.3.vtcmopxnf7d2j4qbloexugjaa
cc73aa34f27c	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	About an hour ago	Up About an hour	80/tcp	webserver.1.2fwrine37mof6vawphkcnt3g1
be9d5dd1cec	nmassol/dbservernm:prc2	"docker-entrypoint.s..."	3 hours ago	Up About an hour	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	dbservernmswarm

```
root@us-nmc:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
root@us-nmc:~#
```

Podem veure com tots els contenidors shan anat a l'altre node

En canvi al enjagarlo veiem com no es reparteixen de nou

```
root@us-nmc:~# docker node update --availability active pxmchgi2ubckx6jcsvn5cegd1
pxmchgi2ubckx6jcsvn5cegd1
root@us-nmc:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2469b46ca9db	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	12 minutes ago	Up 12 minutes	80/tcp	webserver.4.rdhbnjzy59dxu420ahlsv2g4d
fc5907bbc51c	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	12 minutes ago	Up 12 minutes	80/tcp	webserver.2.cb7nvebkvh89vp98r272fadtw
0197cf50797b	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	40 minutes ago	Up 40 minutes	80/tcp	webserver.3.vtcmopxnf7d2j4qbloexugjaa
cc73aa34f27c	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	2 hours ago	Up 2 hours	80/tcp	webserver.1.2fwrine37mof6vawphkcnt3g1
be9d5dd1cec	nmassol/dbservernm:prc2	"docker-entrypoint.s..."	4 hours ago	Up 2 hours	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	dbservernmswarm

```
root@us-nmc:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
root@us-nmc:~#
```

12) Finalment, elimina el servei **webserver**. Que ha passat amb els contenidors a tots els nodes? Amb **docker ps -a** apareixen? Posa'n captures. Funciona la web? Ja pots aturar també el contenidor de la base de dades. **(0,5 punts)**

```
root@us-nmc:~# docker service rm webserver
webserver
root@us-nmc:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2469b46ca9db	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	25 minutes ago	Up 25 minutes	80/tcp	webserver.4.rdhbnjzy59dxu420ahlsv2g4d
fc5907bbc51c	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	25 minutes ago	Up 25 minutes	80/tcp	webserver.2.cb7nvebkvh89vp98r272fadtw
0197cf50797b	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	53 minutes ago	Up 53 minutes	80/tcp	webserver.3.vtcmopxnf7d2j4qbloexugjaa
cc73aa34f27c	nmassol/webbotiganm:prc3	"docker-php-entrypoi..."	2 hours ago	Up 2 hours	80/tcp	webserver.1.2fwrine37mof6vawphkcnt3g1
be9d5dd1cec	nmassol/dbservernm:prc2	"docker-entrypoint.s..."	4 hours ago	Up 2 hours	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	dbservernmswarm

S'ha parat el servei però no els contenidors, tot i això no es pot accedir a la web

